UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

182115 7590 10/10/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER	
BOUTAH, ALINA A	

ART UNIT	PAPER NUMBER
2442	

NOTIFICATION DATE	DELIVERY MODE
10/10/2023	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@appcoll.com
kate@klspatents.com
office@klspatents.com

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00001

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.
17/671,199	10-Oct-2023	11785104

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EGRANT NOTIFICATION

Your electronic patent grant (eGrant) is now available, which can be accessed via Patent Center at https://patentcenter.uspto.gov

The electronic patent grant is the official patent grant under 35 U.S.C. 153. For more information, please visit https://www.uspto.gov/electronicgrants

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00002

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	10/10/2023	11785104	L0041B1CIP2	1070

182115 7590 09/20/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

ISSUE NOTIFICATION

The projected patent number and issue date are specified above. The patent will issue electronically. The electronically issued patent is the official patent grant pursuant to 35 U.S.C. § 153. The patent may be accessed on or after the issue date through Patent Center at https://patentcenter.uspto.gov/. The patent will be available in both the public and the private sides of Patent Center. Further assistance in electronically accessing the patent, or about Patent Center, is available by calling the Patent Electronic Business Center at 1-888-217-9197.

The USPTO is implementing electronic patent issuance with a transition period, during which period the USPTO will mail a ceremonial paper copy of the electronic patent grant to the correspondence address of record. Additional copies of the patent (i.e., certified and presentation copies) may be ordered for a fee from the USPTO's Certified Copy Center at https://certifiedcopycenter.uspto.gov/index.html. The Certified Copy Center may be reached at (800)972-6382.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Center (https://patentcenter.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Patents Stakeholder Experience (OPSE), Stakeholder Support Division (SSD) at (571)-272-4200.

INVENTOR(s) (Please see PATENT CENTER site https://patentcenter.uspto.gov for additional inventors):

ÚLFAR ERLINGSSON, PALO ALTO, CA;
YIJOU CHEN, CUPERTINO, CA;

APPLICANT(s) (Please see PATENT CENTER site https://patentcenter.uspto.gov for additional applicants):

LACEWORK, INC., San Jose, CA;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00004

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

182115 7590 07/27/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER	
BOUTAH, ALINA A	

ART UNIT	PAPER NUMBER
2442	

NOTIFICATION DATE	DELIVERY MODE
07/27/2023	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@appcoll.com
kate@klspatents.com
office@klspatents.com

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00005

Response to Rule 312 Communication	Application No. 17/671,199	Applicant(s)
	Examiner	Art Unit

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address –

1. ☒ The amendment filed on *21 July 2023* under 37 CFR 1.312 has been considered, and has been:

   a) ☒ entered.

   b) ☐ entered as directed to matters of form not affecting the scope of the invention.

   c) ☐ disapproved because the amendment was filed after the payment of the issue fee.
   Any amendment filed after the date the issue fee is paid must be accompanied by a petition under 37 CFR 1.313(c)(1) and the required fee to withdraw the application from issue.

   d) ☐ disapproved. See explanation below.

   e) ☐ entered in part. See explanation below.

/C. Garner/

Publishing Division

Office of Data Management

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00006

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/671,199	**07/24/2023 05:54:30 PM ET**	**L0041B1CIP2**

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1070	FILED BY	Edward Lenart
PATENT CENTER #	62494429	FILING DATE	02/14/2022
CUSTOMER #	182115	FIRST NAMED INVENTOR	ÚLFAR ERLINGSSON
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	-

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
web85b.pdf	3	Issue Fee Payment (PTO-85B)	61 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
web85b.pdf	19D06937FC44219CAE8A2D5AE64FD4ADC12A3DD540B14CB5 B6C0ABAC8DC9C2F09208674FC6D60E6BD819147DE4431E81 853E8B6886202180A876D1C64D7AA0A3

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as

described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/671,199	07/24/2023 05:54:30 PM ET	L0041B1CIP2

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1070	FILED BY	Edward Lenart
PATENT CENTER #	62494429	AUTHORIZED BY	-
CUSTOMER #	182115	FILING DATE	02/14/2022
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	ÚLFAR ERLINGSSON

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1003	E20237NH54538363	Edward Lenart

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1501	UTILITY ISSUE FEE	1200.00	1	1200.00
			TOTAL AMOUNT:	$1,200.00

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage

submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ISSUE FEE TRANSMITTAL FORM

APPLICATION #	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET #
17671199	02/14/2022	ULFAR ERLINGSSON	L0041B1CIP2

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Nonprovisional Application under 35 USC 111(a)	DATE DUE	08/28/2023
CONFIRMATION #	1070	ISSUE FEE DUE	$ 1200
EXAMINER	ALINA BOUTAH	PUBLICATION DUE	$ 0
GROUP ART UNIT	2442	TOTAL FEES DUE	$1200
CLASS - SUBCLASS	709/224000	PREV. PAID FEE	$ 0
ENTITY STATUS	Regular Undiscounted		

1. CHANGE OF CORRESPONDENCE ADDRESS AND/OR INDICATION OF FEE ADDRESS (37 CFR 1.33 & 1.363)

Correspondence Address

CURRENT ADDRESS

Maintenance Fee Address

CURRENT ADDRESS

☐ Change of correspondence address requested,system generated AIA/122-PC form attached

☐ Fee address indication requested, system generatedSB/47-PC form attached

2. ENTITY STATUS

Change in Entity Status

NEW STATUS

3. THE FOLLOWING FEES ARE SUBMITTED

☑ Issue Fee

☑ Publication Fee

4. FEE AUTHORIZATION

☐ I authorize USPTO to apply my previously paid issue fee to the current fees due

☐ The Director is herby authorized to apply my previously paid issue fee to the current fee due and to charge deficient fees to Deposit Account Number:

☑ If in addition to the payment of the issue fee amount submitted with this form, there are any discrepencies in any amount(s) due, the Director is authorized to charge any deficiency, or credit any overpayment, to Deposit Account Number: 503082

5. FIRM AND/OR ATTORNEY NAMES TO BE PRINTED

NOTE: If no name is listed, no name will be printed
For printing on the patent front page, list to be displayed as entered

6. ASSIGNEE NAME(S) AND RESIDENCE DATA TO BE PRINTED

NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a susbstitute for filing an assignment.

LACEWORK, INC.	MOUNTAIN VIEW, CA UNITED STATES	Corporation

Signature

I certify, in accordance with 37 CFR 1.4(d)(4) that I am an attorney or agent registered to practice before the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00012

Patent and Trademark Office who has filed and has been granted power of attorney in this application. I also certify that this Fee(s) Transmittal form is being transmitted to the USPTO via Patent Center on the date indicated below.

Signature	Name	Registration #	Date
/Edward Lenart/	Edward Lenart	63665	07/24/2023

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00013

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This is a continuation in-part application for patent entitled to a filing date and claiming the benefit of earlier-filed U.S. Patent Application No. 17/504,311, filed October 18, 2021, herein incorporated by reference in its entirety, and claims priority from U.S. Provisional Patent Application No. 63/243,013, filed September 10, 2021.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The accompanying drawings illustrate various embodiments and are a part of the specification. The illustrated embodiments are merely examples and do not limit the scope of the disclosure. Throughout the drawings, identical or similar reference numbers designate identical or similar elements.

[0003] Fig. 1A shows an illustrative configuration in which a data platform is configured to perform various operations with respect to a cloud environment that includes a plurality of compute assets.

[0004] Fig. 1B shows an illustrative implementation of the configuration of Fig. 1A.

[0005] Fig. 1C illustrates an example computing device.

[0006] Fig. 1D illustrates an example of an environment in which activities that occur within datacenters are modeled.

[0007] Fig. 2A illustrates an example of a process, used by an agent, to collect and report information about a client.

[0008] Fig. 2B illustrates a 5-tuple of data collected by an agent, physically and logically.

[0009] Fig. 2C illustrates a portion of a polygraph.

[0010] Fig. 2D illustrates a portion of a polygraph.

[0011] Fig. 2E illustrates an example of a communication polygraph.

[0012] Fig. 2F illustrates an example of a polygraph.

[0013] Fig. 2G illustrates an example of a polygraph as rendered in an interface.

[0014] Fig. 2H illustrates an example of a portion of a polygraph as rendered in an interface.

[0015] Fig. 2I illustrates an example of a portion of a polygraph as rendered in an interface.

[0016] Fig. 2J illustrates an example of a portion of a polygraph as rendered in an interface.

[0017] Fig. 2K illustrates an example of a portion of a polygraph as rendered in an interface.

[0018] Fig. 2L illustrates an example of an insider behavior graph as rendered in an interface.

[0019] Fig. 2M illustrates an example of a privilege change graph as rendered in an interface.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00014

[0020] Fig. 2N illustrates an example of a user login graph as rendered in an interface.

[0021] Fig. 2O illustrates an example of a machine server graph as rendered in an interface.

[0022] Fig. 3A illustrates an example of a process for detecting anomalies in a network environment.

[0023] Fig. 3B depicts a set of example processes communicating with other processes.

[0024] Fig. 3C depicts a set of example processes communicating with other processes.

[0025] Fig. 3D depicts a set of example processes communicating with other processes.

[0026] Fig. 3E depicts two pairs of clusters.

[0027] Fig. 3F is a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection.

[0028] Fig. 3G is an alternate representation of actions occurring in Fig. 3F.

[0029] Fig. 3H illustrates an example of a process for performing extended user tracking.

[0030] Fig. 3I is a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection.

[0031] Fig. 3J illustrates an example of a process for performing extended user tracking.

[0032] Fig. 3K illustrates example records.

[0033] Fig. 3L illustrates example output from performing an ssh connection match.

[0034] Fig. 3M illustrates example records.

[0035] Fig. 3N illustrates example records.

[0036] Fig. 3O illustrates example records.

[0037] Fig. 3P illustrates example records.

[0038] Fig. 3Q illustrates an adjacency relationship between two login sessions.

[0039] Fig. 3R illustrates example records.

[0040] Fig. 3S illustrates an example of a process for detecting anomalies.

[0041] Fig. 4A illustrates a representation of an embodiment of an insider behavior graph.

[0042] Fig. 4B illustrates an embodiment of a portion of an insider behavior graph.

[0043] Fig. 4C illustrates an embodiment of a portion of an insider behavior graph.

[0044] Fig. 4D illustrates an embodiment of a portion of an insider behavior graph.

[0045] Fig. 4E illustrates a representation of an embodiment of a user login graph.

[0046] Fig. 4F illustrates an example of a privilege change graph.

[0047] Fig. 4G illustrates an example of a privilege change graph.

[0048] Fig. 4H illustrates an example of a user interacting with a portion of an interface.

[0049] Fig. 4I illustrates an example of a dossier for an event.

[0050] Fig. 4J illustrates an example of a dossier for a domain.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00015

[0051] Fig. 4K depicts an example of an Entity Join graph by FilterKey and FilterKey Group (implicit join).

[0052] Fig. 4L illustrates an example process for dynamically generating and executing a query.

[0053] Fig. 5 sets forth a flowchart illustrating an example method of improving developer efficiency and application quality in accordance with some embodiments.

[0054] Fig. 6 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0055] Fig. 7 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0056] Fig. 8 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0057] Fig. 9 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0058] Fig. 10 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0059] Fig. 11 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0060] Fig. 12 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

~~**[0059]**~~

DETAILED DESCRIPTION

~~**[0060]**~~**[0061]** Various illustrative embodiments are described herein with reference to the accompanying drawings. It will, however, be evident that various modifications and changes may be made thereto, and additional embodiments may be implemented, without departing from the scope of the invention as set forth in the claims. For example, certain features of one embodiment described herein may be combined with or substituted for features of another embodiment described herein. The description and drawings are accordingly to be regarded in an illustrative rather than a restrictive sense.

~~**[0061]**~~**[0062]** Fig. 1A shows an illustrative configuration 10 in which a data platform 12 is configured to perform various operations with respect to a cloud environment 14 that includes a plurality of compute assets 16-1 through 16-N (collectively "compute assets 16). For example, data platform 12 may include data ingestion resources 18 configured to ingest data from cloud environment 14 into data platform 12, data processing resources 20 configured to perform data

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00016

processing operations with respect to the data, user interface resources 22 configured to provide one or more external users and/or compute resources (e.g., computing device 24) with access to an output of data processing resources 20. Each of these resources are described in detail herein.

~~[0062]~~[0063] Cloud environment 14 may include any suitable network-based computing environment as may serve a particular application. For example, cloud environment 14 may be implemented by one or more compute resources provided and/or otherwise managed by one or more cloud service providers, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, and/or any other cloud service provider configured to provide public and/or private access to network-based compute resources.

~~[0063]~~[0064] Compute assets 16 may include, but are not limited to, containers (e.g., container images, deployed and executing container instances, etc.), virtual machines, workloads, applications, processes, physical machines, compute nodes, clusters of compute nodes, software runtime environments (e.g., container runtime environments), and/or any other virtual and/or physical compute resource that may reside in and/or be executed by one or more computer resources in cloud environment 14. In some examples, one or more compute assets 16 may reside in one or more datacenters.

~~[0064]~~[0065] A compute asset 16 may be associated with (e.g., owned, deployed, or managed by) a particular entity, such as a customer or client of cloud environment 14 and/or data platform 12. Accordingly, for purposes of the discussion herein, cloud environment 14 may be used by one or more entities.

~~[0065]~~[0066] Data platform 12 may be configured to perform one or more data security monitoring and/or remediation services, compliance monitoring services, anomaly detection services, DevOps services, compute asset management services, and/or any other type of data analytics service as may serve a particular implementation. Data platform 12 may be managed or otherwise associated with any suitable data platform provider, such as a provider of any of the data analytics services described herein. The various resources included in data platform 12 may reside in the cloud and/or be located on-premises and be implemented by any suitable combination of physical and/or virtual compute resources, such as one or more computing devices, microservices, applications, etc.

~~[0066]~~[0067] Data ingestion resources 18 may be configured to ingest data from cloud environment 14 into data platform 12. This may be performed in various ways, some of which are described in detail herein. For example, as illustrated by arrow 26, data ingestion resources 18 may be configured to receive the data from one or more agents deployed within cloud environment 14, utilize an event streaming platform (e.g., Kafka) to obtain the data, and/or pull

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00017

data (e.g., configuration data) from cloud environment 14. In some examples, data ingestion resources 18 may obtain the data using one or more agentless configurations.

~~[0067]~~[0068] The data ingested by data ingestion resources 18 from cloud environment 14 may include any type of data as may serve a particular implementation. For example, the data may include data representative of configuration information associated with compute assets 16, information about one or more processes running on compute assets 16, network activity information, information about events (creation events, modification events, communication events, user-initiated events, etc.) that occur with respect to compute assets 16, etc. In some examples, the data may or may not include actual customer data processed or otherwise generated by compute assets 16.

~~[0068]~~[0069] As illustrated by arrow 28, data ingestion resources 18 may be configured to load the data ingested from cloud environment 14 into a data store 30. Data store 30 is illustrated in Fig. 1A as being separate from and communicatively coupled to data platform 12. However, in some alternative embodiments, data store 30 is included within data platform 12.

~~[0069]~~[0070] Data store 30 may be implemented by any suitable data warehouse, data lake, data mart, and/or other type of database structure as may serve a particular implementation. Such data stores may be proprietary or may be embodied as vendor provided products or services such as, for example, Snowflake, Google BigQuery, Druid, Amazon Redshift, IBM Db2, Dremio, Databricks Lakehouse Platform, Cloudera, Azure Synapse Analytics, and others.

~~[0070]~~[0071] Although the examples described herein largely relate to embodiments where data is collected from agents and ultimately stored in a data store such as those provided by Snowflake, in other embodiments data that is collected from agents and other sources may be stored in different ways. For example, data that is collected from agents and other sources may be stored in a data warehouse, data lake, data mart, and/or any other data store.

~~[0071]~~[0072] A data warehouse may be embodied as an analytic database (e.g., a relational database) that is created from two or more data sources. Such a data warehouse may be leveraged to store historical data, often on the scale of petabytes. Data warehouses may have compute and memory resources for running complicated queries and generating reports. Data warehouses may be the data sources for business intelligence ('BI') systems, machine learning applications, and/or other applications. By leveraging a data warehouse, data that has been copied into the data warehouse may be indexed for good analytic query performance, without affecting the write performance of a database (e.g., an Online Transaction Processing ('OLTP') database). Data warehouses also enable the joining data from multiple sources for analysis. For example, a sales OLTP application probably has no need to know about the weather at various sales locations, but

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00018

sales predictions could take advantage of that data. By adding historical weather data to a data warehouse, it would be possible to factor it into models of historical sales data.

~~[0072]~~[0073] Data lakes, which store files of data in their native format, may be considered as "schema on read" resources. As such, any application that reads data from the lake may impose its own types and relationships on the data. Data warehouses, on the other hand, are "schema on write," meaning that data types, indexes, and relationships are imposed on the data as it is stored in an enterprise data warehouse (EDW). "Schema on read" resources may be beneficial for data that may be used in several contexts and poses little risk of losing data. "Schema on write" resources may be beneficial for data that has a specific purpose, and good for data that must relate properly to data from other sources. Such data stores may include data that is encrypted using homomorphic encryption, data encrypted using privacy-preserving encryption, smart contracts, non-fungible tokens, decentralized finance, and other techniques.

~~[0073]~~[0074] Data marts may contain data oriented towards a specific business line whereas data warehouses contain enterprise-wide data. Data marts may be dependent on a data warehouse, independent of the data warehouse (e.g., drawn from an operational database or external source), or a hybrid of the two. In embodiments described herein, different types of data stores (including combinations thereof) may be leveraged.

~~[0074]~~[0075] Data processing resources 20 may be configured to perform various data processing operations with respect to data ingested by data ingestion resources 18, including data ingested and stored in data store 30. For example, data processing resources 20 may be configured to perform one or more data security monitoring and/or remediation operations, compliance monitoring operations, anomaly detection operations, DevOps operations, compute asset management operations, and/or any other type of data analytics operation as may serve a particular implementation. Various examples of operations performed by data processing resources 20 are described herein.

~~[0075]~~[0076] As illustrated by arrow 32, data processing resources 20 may be configured to access data in data store 30 to perform the various operations described herein. In some examples, this may include performing one or more queries with respect to the data stored in data store 30. Such queries may be generated using any suitable query language.

~~[0076]~~[0077] In some examples, the queries provided by data processing resources 20 may be configured to direct data store 30 to perform one or more data analytics operations with respect to the data stored within data store 30. These data analytics operations may be with respect to data specific to a particular entity (e.g., data residing in one or more silos within data store 30 that are associated with a particular customer) and/or data associated with multiple entities. For example,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00019

data processing resources 20 may be configured to analyze data associated with a first entity and use the results of the analysis to perform one or more operations with respect to a second entity.

~~[0077]~~[0078] One or more operations performed by data processing resources 20 may be performed periodically according to a predetermined schedule. For example, one or more operations may be performed by processing resources 20 every hour or any other suitable time interval. Additionally or alternatively, one or more operations performed by data processing resources 20 may be performed in substantially real-time (or near real-time) as data is ingested into data platform 12. In this manner, the results of such operations (e.g., one or more detected anomalies in the data) may be provided to one or more external entities (e.g., computing device 24 and/or one or more users) in substantially real-time and/or in near real-time.

~~[0078]~~[0079] User interface resources 22 may be configured to perform one or more user interface operations, examples of which are described herein. For example, user interface resources 22 may be configured to present one or more results of the data processing performed by data processing resources 20 to one or more external entities (e.g., computing device 24 and/or one or more users), as illustrated by arrow 34. As illustrated by arrow 36, user interface resources 22 may access data in data store 30 to perform the one or more user interface operations

~~[0079]~~[0080] Fig. 1B illustrates an implementation of configuration 10 in which an agent 38 (e.g., agent 38-1 through agent 38-N) is installed on each of compute assets 16. As used herein, an agent may include a self-contained binary and/or other type of code or application that can be run on any appropriate platforms, including within containers and/or other virtual compute assets. Agents 38 may monitor the nodes on which they execute for a variety of different activities, including but not limited to, connection, process, user, machine, and file activities. In some examples, agents 38 can be executed in user space, and can use a variety of kernel modules (e.g., auditd, iptables, netfilter, pcap, etc.) to collect data. Agents can be implemented in any appropriate programming language, such as C or Golang, using applicable kernel APIs.

~~[0080]~~[0081] Agents 38 may be deployed in any suitable manner. For example, an agent 38 may be deployed as a containerized application or as part of a containerized application. As described herein, agents 38 may selectively report information to data platform 12 in varying amounts of detail and/or with variable frequency.

~~[0081]~~[0082] Also shown in Fig. 1B is a load balancer 40 configured to perform one or more load balancing operations with respect to data ingestion operations performed by data ingestion resources 18 and/or user interface operations performed by user interface resources 22. Load balancer 40 is shown to be included in data platform 12. However, load balancer 40 may alternatively be located external to data platform 12. Load balancer 40 may be implemented by

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00020

any suitable microservice, application, and/or other computing resources. In some alternative examples, data platform 12 may not utilize a load balancer such as load balancer 40.

~~[0082]~~[0083] Also shown in Fig. 1B is long term storage 42 with which data ingestion resources may interface, as illustrated by arrow 44. Long term storage 42 may be implemented by any suitable type of storage resources, such as cloud-based storage (e.g., AWS S3, etc.) and/or on-premises storage and may be used by data ingestion resources 18 as part of the data ingestion process. Examples of this are described herein. In some examples, data platform 12 may not utilize long term storage 42.

~~[0083]~~[0084] The embodiments described herein can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the principles described herein. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

~~[0084]~~[0085] In some examples, a non-transitory computer-readable medium storing computer-readable instructions may be provided in accordance with the principles described herein. The instructions, when executed by a processor of a computing device, may direct the processor and/or computing device to perform one or more operations, including one or more of the operations described herein. Such instructions may be stored and/or transmitted using any of a variety of known computer-readable media.

~~[0085]~~[0086] A non-transitory computer-readable medium as referred to herein may include any non-transitory storage medium that participates in providing data (e.g., instructions) that may be read and/or executed by a computing device (e.g., by a processor of a computing device). For example, a non-transitory computer-readable medium may include, but is not limited to, any combination of non-volatile storage media and/or volatile storage media. Exemplary non-volatile storage media include, but are not limited to, read-only memory, flash memory, a solid-state drive, a magnetic storage device (e.g. a hard disk, a floppy disk, magnetic tape, etc.), ferroelectric random-access memory ("RAM"), and an optical disc (e.g., a compact disc, a digital video disc, a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00021

Blu-ray disc, etc.). Exemplary volatile storage media include, but are not limited to, RAM (e.g., dynamic RAM).

~~[0086]~~[0087] Fig. 1C illustrates an example computing device 50 that may be specifically configured to perform one or more of the processes described herein. Any of the systems, microservices, computing devices, and/or other components described herein may be implemented by computing device 50.

~~[0087]~~[0088] As shown in Fig. 1C, computing device 50 may include a communication interface 52, a processor 54, a storage device 56, and an input/output ("I/O") module 58 communicatively connected one to another via a communication infrastructure 60. While an exemplary computing device 50 is shown in Fig. 1C, the components illustrated in Fig. 1C are not intended to be limiting. Additional or alternative components may be used in other embodiments. Components of computing device 50 shown in Fig. 1C will now be described in additional detail.

~~[0088]~~[0089] Communication interface 52 may be configured to communicate with one or more computing devices. Examples of communication interface 52 include, without limitation, a wired network interface (such as a network interface card), a wireless network interface (such as a wireless network interface card), a modem, an audio/video connection, and any other suitable interface.

~~[0089]~~[0090] Processor 54 generally represents any type or form of processing unit capable of processing data and/or interpreting, executing, and/or directing execution of one or more of the instructions, processes, and/or operations described herein. Processor 54 may perform operations by executing computer-executable instructions 62 (e.g., an application, software, code, and/or other executable data instance) stored in storage device 56.

~~[0090]~~[0091] Storage device 56 may include one or more data storage media, devices, or configurations and may employ any type, form, and combination of data storage media and/or device. For example, storage device 56 may include, but is not limited to, any combination of the non-volatile media and/or volatile media described herein. Electronic data, including data described herein, may be temporarily and/or permanently stored in storage device 56. For example, data representative of computer-executable instructions 62 configured to direct processor 54 to perform any of the operations described herein may be stored within storage device 56. In some examples, data may be arranged in one or more databases residing within storage device 56.

~~[0091]~~[0092] I/O module 58 may include one or more I/O modules configured to receive user input and provide user output. I/O module 58 may include any hardware, firmware, software, or combination thereof supportive of input and output capabilities. For example, I/O module 58 may include hardware and/or software for capturing user input, including, but not limited to, a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00022

keyboard or keypad, a touchscreen component (e.g., touchscreen display), a receiver (e.g., an RF or infrared receiver), motion sensors, and/or one or more input buttons.

~~[0092]~~[0093] I/O module 58 may include one or more devices for presenting output to a user, including, but not limited to, a graphics engine, a display (e.g., a display screen), one or more output drivers (e.g., display drivers), one or more audio speakers, and one or more audio drivers. In certain embodiments, I/O module 58 is configured to provide graphical data to a display for presentation to a user. The graphical data may be representative of one or more graphical user interfaces and/or any other graphical content as may serve a particular implementation.

~~[0093]~~[0094] Fig. 1D illustrates an example implementation 100 of configuration 10. As such, one more components shown in Fig. 1D may implement one or more components shown in Fig. 1A and/or Fig. 1B. In particular, implementation 100 illustrates an environment in which activities that occur within datacenters are modeled using data platform 12. Using techniques described herein, a baseline of datacenter activity can be modeled, and deviations from that baseline can be identified as anomalous. Anomaly detection can be beneficial in a security context, a compliance context, an asset management context, a DevOps context, and/or any other data analytics context as may serve a particular implementation.

~~[0094]~~[0095] Two example datacenters (104 and 106) are shown in Fig. 1D, and are associated with (e.g., belong to) entities named entity A and entity B, respectively. A datacenter may include dedicated equipment (e.g., owned and operated by entity A, or owned/leased by entity A and operated exclusively on entity A's behalf by a third party). A datacenter can also include cloud-based resources, such as infrastructure as a service (IaaS), platform as a service (PaaS), and/or software as a service (SaaS) elements. The techniques described herein can be used in conjunction with multiple types of datacenters, including ones wholly using dedicated equipment, ones that are entirely cloud-based, and ones that use a mixture of both dedicated equipment and cloud-based resources.

~~[0095]~~[0096] Both datacenter 104 and datacenter 106 include a plurality of nodes, depicted collectively as set of nodes 108 and set of nodes 110, respectively, in Fig. 1D. These nodes may implement compute assets 16. Installed on each of the nodes are in-server / in-virtual machine (VM) / embedded in IoT device agents (e.g., agent 112), which are configured to collect data and report it to data platform 12 for analysis. As described herein, agents may be small, self-contained binaries that can be run on any appropriate platforms, including virtualized ones (and, as applicable, within containers). Agents may monitor the nodes on which they execute for a variety of different activities, including: connection, process, user, machine, and file activities. Agents can be executed in user space, and can use a variety of kernel modules (e.g., auditd,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00023

iptables, netfilter, pcap, etc.) to collect data. Agents can be implemented in any appropriate programming language, such as C or Golang, using applicable kernel APIs.

~~[0096]~~[0097] As described herein, agents can selectively report information to data platform 12 in varying amounts of detail and/or with variable frequency. As is also described herein, the data collected by agents may be used by data platform 12 to create polygraphs, which are graphs of logical entities, connected by behaviors. In some embodiments, agents report information directly to data platform 12. In other embodiments, at least some agents provide information to a data aggregator, such as data aggregator 114, which in turn provides information to data platform 12. The functionality of a data aggregator can be implemented as a separate binary or other application (distinct from an agent binary), and can also be implemented by having an agent execute in an "aggregator mode" in which the designated aggregator node acts as a Layer 7 proxy for other agents that do not have access to data platform 12. Further, a chain of multiple aggregators can be used, if applicable (e.g., with agent 112 providing data to data aggregator 114, which in turn provides data to another aggregator (not pictured) which provides data to data platform 12). An example way to implement an aggregator is through a program written in an appropriate language, such as C or Golang.

~~[0097]~~[0098] Use of an aggregator can be beneficial in sensitive environments (e.g., involving financial or medical transactions) where various nodes are subject to regulatory or other architectural requirements (e.g., prohibiting a given node from communicating with systems outside of datacenter 104). Use of an aggregator can also help to minimize security exposure more generally. As one example, by limiting communications with data platform 12 to data aggregator 114, individual nodes in nodes 108 need not make external network connections (e.g., via Internet 124), which can potentially expose them to compromise (e.g., by other external devices, such as device 118, operated by a criminal). Similarly, data platform 12 can provide updates, configuration information, etc., to data aggregator 114 (which in turn distributes them to nodes 108), rather than requiring nodes 108 to allow incoming connections from data platform 12 directly.

~~[0098]~~[0099] Another benefit of an aggregator model is that network congestion can be reduced (e.g., with a single connection being made at any given time between data aggregator 114 and data platform 12, rather than potentially many different connections being open between various of nodes 108 and data platform 12). Similarly, network consumption can also be reduced (e.g., with the aggregator applying compression techniques/bundling data received from multiple agents).

~~[0099]~~[00100] One example way that an agent (e.g., agent 112, installed on node 116) can provide information to data aggregator 114 is via a REST API, formatted using data serialization

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00024

protocols such as Apache Avro. One example type of information sent by agent 112 to data aggregator 114 is status information. Status information may be sent by an agent periodically (e.g., once an hour or once any other predetermined amount of time). Alternatively, status information may be sent continuously or in response to occurrence of one or more events. The status information may include, but is not limited to, a. an amount of event backlog (in bytes) that has not yet been transmitted, b. configuration information, c. any data loss period for which data was dropped, d. a cumulative count of errors encountered since the agent started, e. version information for the agent binary, and/or f. cumulative statistics on data collection (e.g., number of network packets processed, new processes seen, etc.).

~~[00100]~~[00101] A second example type of information that may be sent by agent 112 to data aggregator 114 is event data (described in more detail herein), which may include a UTC timestamp for each event. As applicable, the agent can control the amount of data that it sends to the data aggregator in each call (e.g., a maximum of 10MB) by adjusting the amount of data sent to manage the conflicting goals of transmitting data as soon as possible, and maximizing throughput. Data can also be compressed or uncompressed by the agent (as applicable) prior to sending the data.

~~[00101]~~[00102] Each data aggregator may run within a particular customer environment. A data aggregator (e.g., data aggregator 114) may facilitate data routing from many different agents (e.g., agents executing on nodes 108) to data platform 12. In various embodiments, data aggregator 114 may implement a SOCKS 5 caching proxy through which agents can connect to data platform 12. As applicable, data aggregator 114 can encrypt (or otherwise obfuscate) sensitive information prior to transmitting it to data platform 12, and can also distribute key material to agents which can encrypt the information (as applicable). Data aggregator 114 may include a local storage, to which agents can upload data (e.g., pcap packets). The storage may have a key-value interface. The local storage can also be omitted, and agents configured to upload data to a cloud storage or other storage area, as applicable. Data aggregator 114 can, in some embodiments, also cache locally and distribute software upgrades, patches, or configuration information (e.g., as received from data platform 12).

~~[00102]~~[00103] Various examples associated with agent data collection and reporting will now be described.

~~[00103]~~[00104] In the following example, suppose that a user (e.g., a network administrator) at entity A (hereinafter "user A") has decided to begin using the services of data platform 12. In some embodiments, user A may access a web frontend (e.g., web app 120) using a computer 126 and enrolls (on behalf of entity A) an account with data platform 12. After enrollment is complete, user A may be presented with a set of installers, pre-built and customized for the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00025

environment of entity A, that user A can download from data platform 12 and deploy on nodes 108. Examples of such installers include, but are not limited to, a Windows executable file, an iOS app, a Linux package (e.g., .deb or .rpm), a binary, or a container (e.g., a Docker container). When a user (e.g., a network administrator) at entity B (hereinafter "user B") also signs up for the services of data platform 12, user B may be similarly presented with a set of installers that are pre-built and customized for the environment of entity B.

~~[00104]~~[00105] User A deploys an appropriate installer on each of nodes 108 (e.g., with a Windows executable file deployed on a Windows-based platform or a Linux package deployed on a Linux platform, as applicable). As applicable, the agent can be deployed in a container. Agent deployment can also be performed using one or more appropriate automation tools, such as Chef, Puppet, Salt, and Ansible. Deployment can also be performed using managed/hosted container management/orchestration frameworks such as Kubernetes, Mesos, and/or Docker Swarm.

~~[00105]~~[00106] In various embodiments, the agent may be installed in the user space (i.e., is not a kernel module), and the same binary is executed on each node of the same type (e.g., all Windows-based platforms have the same Windows-based binary installed on them). An illustrative function of an agent, such as agent 112, is to collect data (e.g., associated with node 116) and report it (e.g., to data aggregator 114). Other tasks that can be performed by agents include data configuration and upgrading.

~~[00106]~~[00107] One approach to collecting data as described herein is to collect virtually all information available about a node (and, e.g., the processes running on it). Alternatively, the agent may monitor for network connections, and then begin collecting information about processes associated with the network connections, using the presence of a network packet associated with a process as a trigger for collecting additional information about the process. As an example, if a user of node 116 executes an application, such as a calculator application, which does not typically interact with the network, no information about use of that application may be collected by agent 112 and/or sent to data aggregator 114. If, however, the user of node 116 executes an ssh command (e.g., to ssh from node 116 to node 122), agent 112 may collect information about the process and provide associated information to data aggregator 114. In various embodiments, the agent may always collect/report information about certain events, such as privilege escalation, irrespective of whether the event is associated with network activity.

~~[00107]~~[00108] An approach to collecting information (e.g., by an agent) is as follows, and described in conjunction with process 200 depicted in Fig. 2A. An agent (e.g., agent 112) monitors its node (e.g., node 116) for network activity. One example way that agent 112 can monitor node 116 for network activity is by using a network packet capture tool (e.g., listening

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00026

using libpcap). As packets are received (201), the agent obtains and maintains (e.g., in an in-memory cache) connection information associated with the network activity (202). Examples of such information include DNS query/response, TCP, UDP, and IP information.

~~[00108]~~[00109] The agent may also determine a process associated with the network connection (203). One example approach is for the agent to use a kernel network diagnostic API (e.g., netlink_diag) to obtain inode/process information from the kernel. Another example approach is for the agent to scan using netstat (e.g., on /proc/net/tcp, /proc/net/tcp6, /proc/net/udp, and /proc/net/udp6) to obtain sockets and relate them to processes. Information such as socket state (e.g., whether a socket is connected, listening, etc.) can also be collected by the agent.

~~[00109]~~[00110] One way an agent can obtain a mapping between a given inode and a process identifier is to scan within the /proc/pid directory. For each of the processes currently running, the agent examines each of their file descriptors. If a file descriptor is a match for the inode, the agent can determine that the process associated with the file descriptor owns the inode. Once a mapping is determined between an inode and a process identifier, the mapping is cached. As additional packets are received for the connection, the cached process information is used (rather than a new search being performed).

~~[00110]~~[00111] In some cases, exhaustively scanning for an inode match across every file descriptor may not be feasible (e.g., due to CPU limitations). In various embodiments, searching through file descriptors is accordingly optimized. User filtering is one example of such an optimization. A given socket is owned by a user. Any processes associated with the socket will be owned by the same user as the socket. When matching an inode (identified as relating to a given socket) against processes, the agent can filter through the processes and only examine the file descriptors of processes sharing the same user owner as the socket. In various embodiments, processes owned by root are always searched against (e.g., even when user filtering is employed).

~~[00111]~~[00112] Another example of an optimization is to prioritize searching the file descriptors of certain processes over others. One such prioritization is to search through the subdirectories of /proc/ starting with the youngest process. One approximation of such a sort order is to search through /proc/ in reverse order (e.g., examining highest numbered processes first). Higher numbered processes are more likely to be newer (i.e., not long-standing processes), and thus more likely to be associated with new connections (i.e., ones for which inode-process mappings are not already cached). In some cases, the most recently created process may not have the highest process identifier (e.g., due to the kernel wrapping through process identifiers).

~~[00112]~~[00113] Another example prioritization is to query the kernel for an identification of the most recently created process and to search in a backward order through the directories in /proc/ (e.g., starting at the most recently created process and working backwards, then wrapping to the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00027

highest value (e.g., 32768) and continuing to work backward from there). An alternate approach is for the agent to keep track of the newest process that it has reported information on (e.g., to data aggregator 114), and begin its search of /proc/ in a forward order starting from the PID of that process.

~~[00113]~~[00114] Another example prioritization is to maintain, for each user actively using node 116, a list of the five (or any other number) most recently active processes. Those processes are more likely than other processes (less active, or passive) on node 116 to be involved with new connections, and can thus be searched first. For many processes, lower valued file descriptors tend to correspond to non-sockets (e.g., stdin, stdout, stderr). Yet another optimization is to preferentially search higher valued file descriptors (e.g., across processes) over lower valued file descriptors (that are less likely to yield matches).

~~[00114]~~[00115] In some cases, while attempting to locate a process identifier for a given inode, an agent may encounter a socket that does not correspond to the inode being matched against and is not already cached. The identity of that socket (and its corresponding inode) can be cached, once discovered, thus removing a future need to search for that pair.

~~[00115]~~[00116] In some cases, a connection may terminate before the agent is able to determine its associated process (e.g., due to a very short-lived connection, due to a backlog in agent processing, etc.). One approach to addressing such a situation is to asynchronously collect information about the connection using the audit kernel API, which streams information to user space. The information collected from the audit API (which can include PID/inode information) can be matched by the agent against pcap/inode information. In some embodiments, the audit API is always used, for all connections. However, due to CPU utilization considerations, use of the audit API can also be reserved for short/otherwise problematic connections (and/or omitted, as applicable).

~~[00116]~~[00117] Once the agent has determined which process is associated with the network connection (203), the agent can then collect additional information associated with the process (204). As will be described in more detail below, some of the collected information may include attributes of the process (e.g., a process parent hierarchy, and an identification of a binary associated with the process). As will also be described in more detail below, other of the collected information is derived (e.g., session summarization data and hash values).

~~[00117]~~[00118] The collected information is then transmitted (205), e.g., by an agent (e.g., agent 112) to a data aggregator (e.g., data aggregator 114), which in turn provides the information to data platform 12. In some embodiments, all information collected by an agent may be transmitted (e.g., to a data aggregator and/or to data platform 12). In other embodiments, the amount of data transmitted may be minimized (e.g., for efficiency reasons), using various techniques.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00028

~~[00118]~~[00119] One approach to minimizing the amount of data flowing from agents (such as agents installed on nodes 108) to data platform 12 is to use a technique of implicit references with unique keys. The keys can be explicitly used by data platform 12 to extract/derive relationships, as necessary, in a data set at a later time, without impacting performance.

~~[00119]~~[00120] As previously mentioned, some data collected about a process is constant and does not change over the lifetime of the process (e.g., attributes), and some data changes (e.g., statistical information and other variable information). Constant data can be transmitted (210) once, when the agent first becomes aware of the process. And, if any changes to the constant data are detected (e.g., a process changes its parent), a refreshed version of the data can be transmitted (210) as applicable.

~~[00120]~~[00121] In some examples, an agent may collect variable data (e.g., data that may change over the lifetime of the process). In some examples, variable data can be transmitted (210) at periodic (or other) intervals. Alternatively, variable data may be transmitted in substantially real time as it is collected. In some examples, the variable data may indicate a thread count for a process, a total virtual memory used by the process, the total resident memory used by the process, the total time spent by the process executing in user space, and/or the total time spent by the process executing in kernel space. In some examples, the data may include a hash that may be used within data platform 12 to join process creation time attributes with runtime attributes to construct a full dataset.

~~[00121]~~[00122] Below are additional examples of data that an agent, such as agent 112, can collect and provide to data platform 12.

~~[00122]~~[00123] 1. User Data

~~[00123]~~[00124] Core User Data: user name, UID (user ID), primary group, other groups, home directory.

~~[00124]~~[00125] Failed Login Data: IP address, hostname, username, count.

~~[00125]~~[00126] User Login Data: user name, hostname, IP address, start time, TTY (terminal), UID (user ID), GID (group ID), process, end time.

~~[00126]~~[00127] 2. Machine Data

~~[00127]~~[00128] Dropped Packet Data: source IP address, destination IP address, destination port, protocol, count.

~~[00128]~~[00129] Machine Data: hostname, domain name, architecture, kernel, kernel release, kernel version, OS, OS version, OS description, CPU, memory, model number, number of cores, last boot time, last boot reason, tags (e.g., Cloud provider tags such as AWS, GCP, or Azure tags), default router, interface name, interface hardware address, interface IP address and mask, promiscuous mode.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00029

~~[00129]~~[00130] 3. Network Data

~~[00130]~~[00131] Network Connection Data: source IP address, destination IP address, source port, destination port, protocol, start time, end time, incoming and outgoing bytes, source process, destination process, direction of connection, histograms of packet length, inter packet delay, session lengths, etc.

~~[00131]~~[00132] Listening Ports in Server: source IP address, port number, protocol, process.

~~[00132]~~[00133] Dropped Packet Data: source IP address, destination IP address, destination port, protocol, count.

~~[00133]~~[00134] Arp Data: source hardware address, source IP address, destination hardware address, destination IP address.

~~[00134]~~[00135] DNS Data: source IP address, response code, response string, question (request), packet length, final answer (response).

~~[00135]~~[00136] 4. Application Data

~~[00136]~~[00137] Package Data: exe path, package name, architecture, version, package path, checksums (MD5, SHA-1, SHA-256), size, owner, owner ID.

~~[00137]~~[00138] Application Data: command line, PID (process ID), start time, UID (user ID), EUID (effective UID), PPID (parent process ID), PGID (process group ID), SID (session ID), exe path, username, container ID.

~~[00138]~~[00139] 5. Container Data

~~[00139]~~[00140] Container Image Data: image creation time, parent ID, author, container type, repo, (AWS) tags, size, virtual size, image version.

~~[00140]~~[00141] Container Data: container start time, container type, container name, container ID, network mode, privileged, PID mode, IP addresses, listening ports, volume map, process ID.

~~[00141]~~[00142] 6. File Data

~~[00142]~~[00143] File path, file data hash, symbolic links, file creation data, file change data, file metadata, file mode.

~~[00143]~~[00144] As mentioned above, an agent, such as agent 112, can be deployed in a container (e.g., a Docker container), and can also be used to collect information about containers. Collection about a container can be performed by an agent irrespective of whether the agent is itself deployed in a container or not (as the agent can be deployed in a container running in a privileged mode that allows for monitoring).

~~[00144]~~[00145] Agents can discover containers (e.g., for monitoring) by listening for container create events (e.g., provided by Docker), and can also perform periodic ordered discovery scans to determine whether containers are running on a node. When a container is discovered, the agent can obtain attributes of the container, e.g., using standard Docker API calls (e.g., to obtain IP

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00030

addresses associated with the container, whether there's a server running inside, what port it is listening on, associated PIDs, etc.). Information such as the parent process that started the container can also be collected, as can information about the image (which comes from the Docker repository).

~~[00145]~~[00146] In various embodiments, agents may use namespaces to determine whether a process is associated with a container. Namespaces are a feature of the Linux kernel that can be used to isolate resources of a collection of processes. Examples of namespaces include process ID (PID) namespaces, network namespaces, and user namespaces. Given a process, the agent can perform a fast lookup to determine whether the process is part of the namespace the container claims to be its namespace.

~~[00146]~~[00147] As mentioned, agents can be configured to report certain types of information (e.g., attribute information) once, when the agent first becomes aware of a process. In various embodiments, such static information is not reported again (or is reported once a day, every twelve hours, etc.), unless it changes (e.g., a process changes its parent, changes its owner, or a SHA-1 of the binary associated with the process changes).

~~[00147]~~[00148] In contrast to static/attribute information, certain types of data change constantly (e.g., network-related data). In various embodiments, agents are configured to report a list of current connections every minute (or other appropriate time interval). In that connection list will be connections that started in that minute interval, connections that ended in that minute interval, and connections that were ongoing throughout the minute interval (e.g., a one minute slice of a one hour connection).

~~[00148]~~[00149] In various embodiments, agents are configured to collect/compute statistical information about connections (e.g., at the one minute level of granularity and or at any other time interval). Examples of such information include, for the time interval, the number of bytes transferred, and in which direction. Another example of information collected by an agent about a connection is the length of time between packets. For connections that span multiple time intervals (e.g., a seven minute connection), statistics may be calculated for each minute of the connection. Such statistical information (for all connections) can be reported (e.g., to a data aggregator) once a minute.

~~[00149]~~[00150] In various embodiments, agents are also configured to maintain histogram data for a given network connection, and provide the histogram data (e.g., in the Apache Avro data exchange format) under the Connection event type data. Examples of such histograms include: 1. a packet length histogram (packet_len_hist), which characterizes network packet distribution; 2. a session length histogram (session_len_hist), which characterizes a network session length; 3. a session time histogram (session_time_hist), which characterizes a network session time; and 4. a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00031

session switch time histogram (session_switch_time_hist), which characterizes network session switch time (i.e., incoming->outgoing and vice versa). For example, histogram data may include one or more of the following fields: 1. count, which provides a count of the elements in the sampling; 2. sum, which provides a sum of elements in the sampling; 3. max, which provides the highest value element in the sampling; 4. std_dev, which provides the standard deviation of elements in the sampling; and 5. buckets, which provides a discrete sample bucket distribution of sampling data (if applicable).

~~[00150]~~[00151] For some protocols (e.g., HTTP), typically, a connection is opened, a string is sent, a string is received, and the connection is closed. For other protocols (e.g., NFS), both sides of the connection engage in a constant chatter. Histograms allow data platform 12 to model application behavior (e.g., using machine learning techniques), for establishing baselines, and for detecting deviations. As one example, suppose that a given HTTP server typically sends/receives 1,000 bytes (in each direction) whenever a connection is made with it. If a connection generates 500 bytes of traffic, or 2,000 bytes of traffic, such connections would be considered within the typical usage pattern of the server. Suppose, however, that a connection is made that results in 10G of traffic. Such a connection is anomalous and can be flagged accordingly.

~~[00151]~~[00152] Returning to Fig. 1D, as previously mentioned, data aggregator 114 may be configured to provide information (e.g., collected from nodes 108 by agents) to data platform 12. Data aggregator 128 may be similarly configured to provide information to data platform 12. As shown in Fig. 1D, both aggregator 114 and aggregator 128 may connect to a load balancer 130, which accepts connections from aggregators (and/or as applicable, agents), as well as other devices, such as computer 126 (e.g., when it communicates with web app 120), and supports fair balancing. In various embodiments, load balancer 130 is a reverse proxy that load balances accepted connections internally to various microservices (described in more detail below), allowing for services provided by data platform 12 to scale up as more agents are added to the environment and/or as more entities subscribe to services provided by data platform 12. Example ways to implement load balancer 130 include, but are not limited to, using HaProxy, using nginx, and using elastic load balancing (ELB) services made available by Amazon.

~~[00152]~~[00153] Agent service 132 is a microservice that is responsible for accepting data collected from agents (e.g., provided by aggregator 114). In various embodiments, agent service 132 uses a standard secure protocol, such as HTTPS to communicate with aggregators (and as applicable agents), and receives data in an appropriate format such as Apache Avro. When agent service 132 receives an incoming connection, it can perform a variety of checks, such as to see whether the data is being provided by a current customer, and whether the data is being provided

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00032

in an appropriate format. If the data is not appropriately formatted (and/or is not provided by a current customer), it may be rejected.

~~[00153]~~[00154] If the data is appropriately formatted, agent service 132 may facilitate copying the received data to a streaming data stable storage using a streaming service (e.g., Amazon Kinesis and/or any other suitable streaming service. Once the ingesting into the streaming service is complete, service 132 may send an acknowledgement to the data provider (e.g., data aggregator 114). If the agent does not receive such an acknowledgement, it is configured to retry sending the data to data platform 12. One way to implement agent service 132 is as a REST API server framework (e.g., Java DropWizard), configured to communicate with Kinesis (e.g., using a Kinesis library).

~~[00154]~~[00155] In various embodiments, data platform 12 uses one or more streams (e.g., Kinesis streams) for all incoming customer data (e.g., including data provided by data aggregator 114 and data aggregator 128), and the data is sharded based on the node (also referred to herein as a "machine") that originated the data (e.g., node 116 vs. node 122), with each node having a globally unique identifier within data platform 12. Multiple instances of agent service 132 can write to multiple shards.

~~[00155]~~[00156] Kinesis is a streaming service with a limited period (e.g., 1-7 days). To persist data longer than a day, the data may be copied to long term storage 42 (e.g., S3). Data loader 136 is a microservice that is responsible for picking up data from a data stream (e.g., a Kinesis stream) and persisting it in long term storage 42. In one example embodiment, files collected by data loader 136 from the Kinesis stream are placed into one or more buckets, and segmented using a combination of a customer identifier and time slice. Given a particular time segment, and a given customer identifier, the corresponding file (stored in long term storage) contains five minutes (or another appropriate time slice) of data collected at that specific customer from all of the customer's nodes. Data loader 136 can be implemented in any appropriate programming language, such as Java or C, and can be configured to use a Kinesis library to interface with Kinesis. In various embodiments, data loader 136 uses the Amazon Simple Queue Service (SQS) (e.g., to alert DB loader 140 that there is work for it to do).

~~[00156]~~[00157] DB loader 140 is a microservice that is responsible for loading data into an appropriate data store 30, such as SnowflakeDB or Amazon Redshift, using individual per-customer databases. In particular, DB loader 140 is configured to periodically load data into a set of raw tables from files created by data loader 136 as per above. DB loader 140 manages throughput, errors, etc., to make sure that data is loaded consistently and continuously. Further, DB loader 140 can read incoming data and load into data store 30 data that is not already present in tables of data store 30 (also referred to herein as a database). DB loader 140 can be

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00033

implemented in any appropriate programming language, such as Java or C, and an SQL framework such as jOOQ (e.g., to manage SQLs for insertion of data), and SQL/JDBC libraries. In some examples, DB loader 140 may use Amazon S3 and Amazon Simple Queue Service (SQS) to manage files being transferred to and from data store 30.

~~[00157]~~[00158] Customer data included in data store 30 can be augmented with data from additional data sources, such as AWS CloudTrail and/or other types of external tracking services. To this end, data platform may include a tracking service analyzer 144, which is another microservice. Tracking service analyzer 144 may pull data from an external tracking service (e.g., Amazon CloudTrail) for each applicable customer account, as soon as the data is available. Tracking service analyzer 144 may normalize the tracking data as applicable, so that it can be inserted into data store 30 for later querying/analysis. Tracking service analyzer 144 can be written in any appropriate programming language, such as Java or C. Tracking service analyzer 144 also makes use of SQL/JDBC libraries to interact with data store 30 to insert/query data.

~~[00158]~~[00159] As described herein, data platform 12 can model activities that occur within datacenters, such as datacenters 104 and 106. The model may be stable over time, and differences, even subtle ones (e.g., between a current state of the datacenter and the model) can be surfaced. The ability to surface such anomalies can be particularly beneficial in datacenter environments where rogue employees and/or external attackers may operate slowly (e.g., over a period of months), hoping that the elastic nature of typical resource use (e.g., virtualized servers) will help conceal their nefarious activities.

~~[00159]~~[00160] Using techniques described herein, data platform 12 can automatically discover entities (which may implement compute assets 16) deployed in a given datacenter. Examples of entities include workloads, applications, processes, machines, virtual machines, containers, files, IP addresses, domain names, and users. The entities may be grouped together logically (into analysis groups) based on behaviors, and temporal behavior baselines can be established. In particular, using techniques described herein, periodic graphs can be constructed (also referred to herein as polygraphs), in which the nodes are applicable logical entities, and the edges represent behavioral relationships between the logical entities in the graph. Baselines can be created for every node and edge.

~~[00160]~~[00161] Communication (e.g., between applications/nodes) is one example of a behavior. A model of communications between processes is an example of a behavioral model. As another example, the launching of applications is another example of a behavior that can be modeled. The baselines may be periodically updated (e.g., hourly) for every entity. Additionally or alternatively, the baselines may be continuously updated in substantially real-time as data is collected by agents. Deviations from the expected normal behavior can then be detected and

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00034

automatically reported (e.g., as anomalies or threats detected). Such deviations may be due to a desired change, a misconfiguration, or malicious activity. As applicable, data platform 12 can score the detected deviations (e.g., based on severity and threat posed). Additional examples of analysis groups include models of machine communications, models of privilege changes, and models of insider behaviors (monitoring the interactive behavior of human users as they operate within the datacenter).

~~[00161]~~[00162] Two example types of information collected by agents are network level information and process level information. As previously mentioned, agents may collect information about every connection involving their respective nodes. And, for each connection, information about both the server and the client may be collected (e.g., using the connection-to-process identification techniques described above). DNS queries and responses may also be collected. The DNS query information can be used in logical entity graphing (e.g., collapsing many different IP addresses to a single service -- e.g., s3.amazon.com). Examples of process level information collected by agents include attributes (user ID, effective user ID, and command line). Information such as what user/application is responsible for launching a given process and the binary being executed (and its SHA-256 values) may also be provided by agents.

~~[00162]~~[00163] The dataset collected by agents across a datacenter can be very large, and many resources (e.g., virtual machines, IP addresses, etc.) are recycled very quickly. For example, an IP address and port number used at a first point in time by a first process on a first virtual machine may very rapidly be used (e.g., an hour later) by a different process/virtual machine.

~~[00163]~~[00164] A dataset (and elements within it) can be considered at both a physical level, and a logical level, as illustrated in Fig. 2B. In particular, Fig. 2B illustrates an example 5-tuple of data 210 collected by an agent, represented physically (216) and logically (217). The 5-tuple includes a source address 211, a source port 212, a destination address 213, a destination port 214, and a protocol 215. In some cases, port numbers (e.g., 212, 214) may be indicative of the nature of a connection (e.g., with certain port usage standardized). However, in many cases, and in particular in datacenters, port usage is ephemeral. For example, a Docker container can listen on an ephemeral port, which is unrelated to the service it will run. When another Docker container starts (for the same service), the port may well be different. Similarly, particularly in a virtualized environment, IP addresses may be recycled frequently (and are thus also potentially ephemeral) or could be NATed, which makes identification difficult.

~~[00164]~~[00165] A physical representation of the 5-tuple is depicted in region 216. A process 218 (executing on machine 219) has opened a connection to machine 220. In particular, process 218 is in communication with process 221. Information such as the number of packets exchanged between the two machines over the respective ports can be recorded.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00035

~~[00165]~~[00166] As previously mentioned, in a datacenter environment, portions of the 5-tuple may change – potentially frequently – but still be associated with the same behavior. Namely, one application (e.g., Apache) may frequently be in communication with another application (e.g., Oracle), using ephemeral datacenter resources. Further, either/both of Apache and Oracle may be multi-homed. This can lead to potentially thousands of 5-tuples (or more) that all correspond to Apache communicating with Oracle within a datacenter. For example, Apache could be executed on a single machine, and could also be executed across fifty machines, which are variously spun up and down (with different IP addresses each time). An alternate representation of the 5-tuple of data 210 is depicted in region 217, and is logical. The logical representation of the 5-tuple aggregates the 5-tuple (along with other connections between Apache and Oracle having other 5-tuples) as logically representing the same connection. By aggregating data from raw physical connection information into logical connection information, using techniques described herein, a size reduction of six orders of magnitude in the data set can be achieved.

~~[00166]~~[00167] Fig. 2C depicts a portion of a logical polygraph. Suppose a datacenter has seven instances of the application update_engine 225, executing as seven different processes on seven different machines, having seven different IP addresses, and using seven different ports. The instances of update_engine variously communicate with update.core-os.net 226, which may have a single IP address or many IP addresses itself, over the one hour time period represented in the polygraph. In the example shown in Fig. 2C, update_engine is a client, connecting to the server update.core-os.net, as indicated by arrow 228.

~~[00167]~~[00168] Behaviors of the seven processes are clustered together, into a single summary. As indicated in region 227, statistical information about the connections is also maintained (e.g., number of connections, histogram information, etc.). A polygraph such as is depicted in Fig. 2C can be used to establish a baseline of behavior (e.g., at the one-hour level), allowing for the future detection of deviations from that baseline. As one example, suppose that statistically an update_engine instance transmits data at 11 bytes per second. If an instance were instead to transmit data at 1000 bytes per second, such behavior would represent a deviation from the baseline and could be flagged accordingly. Similarly, changes that are within the baseline (e.g., an eighth instance of update_engine appears, but otherwise behaves as the other instances; or one of the seven instances disappears) are not flagged as anomalous. Further, datacenter events, such as failover, autobalancing, and A-B refresh are unlikely to trigger false alarms in a polygraph, as at the logical level, the behaviors remain the same.

~~[00168]~~[00169] In various embodiments, polygraph data is maintained for every application in a datacenter, and such polygraph data can be combined to make a single datacenter view across all

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00036

such applications. Fig. 2D illustrates a portion of a polygraph for a service that evidences more complex behaviors than are depicted in Fig. 2C. In particular, Fig. 2D illustrates the behaviors of S3 as a service (as used by a particular customer datacenter). Clients within the datacenter variously connect to the S3 service using one of five fully qualified domains (listed in region 230). Contact with any of the domains is aggregated as contact with S3 (as indicated in region 231). Depicted in region 232 are various containers which (as clients) connect with S3. Other containers (which do not connect with S3) are not included. As with the polygraph portion depicted in Fig. 2C, statistical information about the connections is known and summarized, such as the number of bytes transferred, histogram information, etc.

~~[00169]~~[00170] Fig. 2E illustrates a communication polygraph for a datacenter. In particular, the polygraph indicates a one hour summary of approximately 500 virtual machines, which collectively run one million processes, and make 100 million connections in that hour. As illustrated in Fig. 2E, a polygraph represents a drastic reduction in size (e.g., from tracking information on 100 million connections in an hour, to a few hundred nodes and a few hundred edges). Further, as a datacenter scales up (e.g., from using 10 virtual machines to 100 virtual machines as the datacenter uses more workers to support existing applications), the polygraph for the datacenter will tend to stay the same size (with the 100 virtual machines clustering into the same nodes that the 10 virtual machines previously clustered into). As new applications are added into the datacenter, the polygraph may automatically scale to include behaviors involving those applications.

~~[00170]~~[00171] In the particular polygraph shown in Fig. 2E, nodes generally correspond to workers, and edges correspond to communications the workers engage in (with connection activity being the behavior modeled in polygraph 235). Another example polygraph could model other behavior, such as application launching. The communications graphed in Fig. 2E include traffic entering the datacenter, traffic exiting the datacenter, and traffic that stays wholly within the datacenter (e.g., traffic between workers). One example of a node included in polygraph 235 is the sshd application, depicted as node 236. As indicated in Fig. 2E, 421 instances of sshd were executing during the one hour time period of data represented in polygraph 235. As indicated in region 237, nodes within the datacenter communicated with a total of 1349 IP addresses outside of the datacenter (and not otherwise accounted for, e.g., as belonging to a service such as Amazon AWS 238 or Slack 239).

~~[00171]~~[00172] In the following examples, suppose that user B, an administrator of datacenter 106, is interacting with data platform 12 to view visualizations of polygraphs in a web browser (e.g., as served to user B via web app 120). One type of polygraph user B can view is an application-communication polygraph, which indicates, for a given one hour window (or any

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00037

other suitable time interval), which applications communicated with which other applications. Another type of polygraph user B can view is an application launch polygraph. User B can also view graphs related to user behavior, such as an insider behavior graph which tracks user connections (e.g., to internal and external applications, including chains of such behavior), a privilege change graph which tracks how privileges change between processes, and a user login graph, which tracks which (logical) machines a user logs into.

~~[00172]~~[00173] Fig. 2F illustrates an example of an application-communication polygraph for a datacenter (e.g., datacenter 106) for the one hour period of 9am-10am on June 5. The time slice currently being viewed is indicated in region 240. If user B clicks his mouse in region 241, user B will be shown a representation of the application-communication polygraph as generated for the following hour (10am-11am on June 5).

~~[00173]~~[00174] Fig. 2G depicts what is shown in user B's browser after he has clicked on region 241, and has further clicked on region 242. The selection in region 242 turns on and off the ability to compare two time intervals to one another. User B can select from a variety of options when comparing the 9am-10am and 10am-11am time intervals. By clicking region 248, user B will be shown the union of both graphs (i.e., any connections that were present in either time interval). By clicking region 249, user B will be shown the intersection of both graphs (i.e., only those connections that were present in both time intervals).

~~[00174]~~[00175] As shown in Fig. 2G, user B has elected to click on region 250, which depicts connections that are only present in the 9am-10am polygraph in a first color 251, and depicts connections that are only present in the 10am-11am polygraph in a second color 252. Connections present in both polygraphs are omitted from display. As one example, in the 9am-10am polygraph (corresponding to connections made during the 9am-10am time period at datacenter 106), a connection was made by a server to sshd (253) and also to systemd (254). Both of those connections ended prior to 10am and are thus depicted in the first color. As another example, in the 10am-11am polygraph (corresponding to connections made during the 10am-11am time period at datacenter 106), a connection was made from a known bad external IP to nginx (255). The connection was not present during the 9am-10am time slice and thus is depicted in the second color. As yet another example, two different connections were made to a Slack service between 9am and 11am. However, the first was made by a first client during the 9am-10am time slice (256) and the second was made by a different client during the 10am-11am slice (257), and so the two connections are depicted respectively in the first and second colors and blue.

~~[00175]~~[00176] Returning to the polygraph depicted in Fig. 2F, suppose user B enters "etcd" into the search box located in region 244. User B will then be presented with the interface illustrated

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00038

in Fig. 2H. As shown in Fig. 2H, three applications containing the term "etcd" were engaged in communications during the 9am-10am window. One application is etcdctl, a command line client for etcd. As shown in Fig. 2H, a total of three different etcdctl processes were executed during the 9am-10am window, and were clustered together (260). Fig. 2H also depicts two different clusters that are both named etcd2. The first cluster includes (for the 9am-10am window) five members (261) and the second cluster includes (for the same window) eight members (262). The reason for these two distinct clusters is that the two groups of applications behave differently (e.g., they exhibit two distinct sets of communication patterns). Specifically, the instances of etcd2 in cluster 261 only communicate with locksmithctl (263) and other etcd2 instances (in both clusters 261 and 262). The instances of etcd2 in cluster 262 communicate with additional entities, such as etcdctl and Docker containers. As desired, user B can click on one of the clusters (e.g., cluster 261) and be presented with summary information about the applications included in the cluster, as is shown in Fig. 2I (e.g., in region 265). User B can also double click on a given cluster (e.g., cluster 261) to see details on each of the individual members of the cluster broken out.

~~[00176]~~[00177] Suppose user B now clicks on region 245 of the interface shown in Fig. 2F. User B will then be shown an application launch polygraph. Launching an application is another example of a behavior. The launch polygraph models how applications are launched by other applications. Fig. 2J illustrates an example of a portion of a launch polygraph. In particular, user B has typed "find" into region 266, to see how the "find" application is being launched. As shown in Fig. 2J, in the launch polygraph for the 10am-11am time period, find applications (267) are always launched by bash (268), which is in turn always launched by systemd (269). If find is launched by a different application, this would be anomalous behavior.

~~[00177]~~[00178] Fig. 2K illustrates another example of a portion of an application launch polygraph. In Fig. 2K, user B has searched (270) for "python ma" to see how "python marathon_lb" (271) is launched. As shown in Fig. 2K, in each case (during the one hour time slice of 10am-11am), python marathon_lb is launched as a result of a chain of the same seven applications each time. If python marathon_lb is ever launched in a different manner, this indicates anomalous behavior. The behavior could be indicative of malicious activities, but could also be due to other reasons, such as a misconfiguration, a performance-related issue, and/or a failure, etc.

~~[00178]~~[00179] Suppose user B now clicks on region 246 of the interface shown in Fig. 2F. User B will then be shown an insider behavior graph. The insider behavior graph tracks information about behaviors such as processes started by a user interactively using protocols such as ssh or telnet, and any processes started by those processes. As one example, suppose an administrator

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00039

logs into a first virtual machine in datacenter 106 (e.g., using sshd via an external connection he makes from a hotel), using a first set of credentials (e.g., first.last@example.com and an appropriate password). From the first virtual machine, the administrator connects to a second virtual machine (e.g., using the same credentials), then uses the sudo command to change identities to those of another user, and then launches a program. graphs built by data platform 12 can be used to associate the administrator with each of his actions, including launching the program using the identity of another user.

~~[00179]~~[00180] Fig. 2L illustrates an example of a portion of an insider behavior graph. In particular, in Fig. 2L, user B is viewing a graph that corresponds to the time slice of 3pm-4pm on June 1. Fig. 2L illustrates the internal/external applications that users connected to during the one hour time slice. If a user typically communicates with particular applications, that information will become part of a baseline. If the user deviates from his baseline behavior (e.g., using new applications, or changing privilege in anomalous ways), such anomalies can be surfaced.

~~[00180]~~[00181] Fig. 2M illustrates an example of a portion of a privilege change graph, which identifies how privileges are changed between processes. Typically, when a user launches a process (e.g., "ls"), the process inherits the same privileges that the user has. And, while a process can have fewer privileges than the user (i.e., go down in privilege), it is rare (and generally undesirable) for a user to escalate in privilege. Information included in the privilege change graph can be determined by examining the parent of each running process, and determining whether there is a match in privilege between the parent and the child. If the privileges are different, a privilege change has occurred (whether a change up or a change down). The application ntpd is one rare example of a scenario in which a process escalates (272) to root, and then returns back (273). The sudo command is another example (e.g., used by an administrator to temporarily have a higher privilege). As with the other examples, ntpd's privilege change actions, and the legitimate actions of various administrators (e.g., using sudo) will be incorporated into a baseline model by data platform 12. When deviations occur, such as where a new application that is not ntpd escalates privilege, or where an individual that has not previously/does not routinely use sudo does so, such behaviors can be identified as anomalous.

~~[00181]~~[00182] Fig. 2N illustrates an example of a portion of a user login graph, which identifies which users log into which logical nodes. Physical nodes (whether bare metal or virtualized) are clustered into a logical machine cluster, for example, using yet another graph, a machine-server graph, an example of which is shown in Fig. 2O. For each machine, a determination is made as to what type of machine it is, based on what kind(s) of workflows it runs. As one example, some machines run as master nodes (having a typical set of workflows they run, as master nodes) and can thus be clustered as master nodes. Worker nodes are different from master nodes, for

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00040

example, because they run Docker containers, and frequently change as containers move around. Worker nodes can similarly be clustered.

~~[00182]~~[00183] As previously mentioned, the polygraph depicted in Fig. 2E corresponds to activities in a datacenter in which, in a given hour, approximately 500 virtual machines collectively run one million processes, and make 100 million connections in that hour. The polygraph represents a drastic reduction in size (e.g., from tracking information on 100 million connections in an hour, to a few hundred nodes and a few hundred edges). Using techniques described herein, such a polygraph can be constructed (e.g., using commercially available computing infrastructure) in less than an hour (e.g., within a few minutes). Thus, ongoing hourly snapshots of a datacenter can be created within a two hour moving window (i.e., collecting data for the time period 8am-9am, while also generating a snapshot for the time previous time period 7am-8am). The following describes various example infrastructure that can be used in polygraph construction, and also describes various techniques that can be used to construct polygraphs.

~~[00183]~~[00184] Returning to Fig. 1D, embodiments of data platform 12 may be built using any suitable infrastructure as a service (IaaS) (e.g., AWS). For example, data platform 12 can use Simple Storage Service (S3) for data storage, Key Management Service (KMS) for managing secrets, Simple Queue Service (SQS) for managing messaging between applications, Simple Email Service (SES) for sending emails, and Route 53 for managing DNS. Other infrastructure tools can also be used. Examples include: orchestration tools (e.g., Kubernetes or Mesos/Marathon), service discovery tools (e.g., Mesos-DNS), service load balancing tools (e.g., marathon-LB), container tools (e.g., Docker or rkt), log/metric tools (e.g., collectd, fluentd, kibana, etc.), big data processing systems (e.g., Spark, Hadoop, AWS Redshift, Snowflake etc.), and distributed key value stores (e.g., Apache Zookeeper or etcd2).

~~[00184]~~[00185] As previously mentioned, in various embodiments, data platform 12 may make use of a collection of microservices. Each microservice can have multiple instances, and may be configured to recover from failure, scale, and distribute work amongst various such instances, as applicable. For example, microservices are auto-balancing for new instances, and can distribute workload if new instances are started or existing instances are terminated. In various embodiments, microservices may be deployed as self-contained Docker containers. A Mesos-Marathon or Spark framework can be used to deploy the microservices (e.g., with Marathon monitoring and restarting failed instances of microservices as needed). The service etcd2 can be used by microservice instances to discover how many peer instances are running, and used for calculating a hash-based scheme for workload distribution. Microservices may be configured to publish various health/status metrics to either an SQS queue, or etcd2, as applicable. In some examples, Amazon DynamoDB can be used for state management.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00041

~~[00185]~~[00186] Additional information on various microservices used in embodiments of data platform 12 is provided below.

~~[00186]~~[00187] Graph generator 146 is a microservice that may be responsible for generating raw behavior graphs on a per customer basis periodically (e.g., once an hour). In particular, graph generator 146 may generate graphs of entities (as the nodes in the graph) and activities between entities (as the edges). In various embodiments, graph generator 146 also performs other functions, such as aggregation, enrichment (e.g., geolocation and threat), reverse DNS resolution, TF-IDF based command line analysis for command type extraction, parent process tracking, etc.

~~[00187]~~[00188] Graph generator 146 may perform joins on data collected by the agents, so that both sides of a behavior are linked. For example, suppose a first process on a first virtual machine (e.g., having a first IP address) communicates with a second process on a second virtual machine (e.g., having a second IP address). Respective agents on the first and second virtual machines may each report information on their view of the communication (e.g., the PID of their respective processes, the amount of data exchanged and in which direction, etc.). When graph generator performs a join on the data provided by both agents, the graph will include a node for each of the processes, and an edge indicating communication between them (as well as other information, such as the directionality of the communication – i.e., which process acted as the server and which as the client in the communication).

~~[00188]~~[00189] In some cases, connections are process to process (e.g., from a process on one virtual machine within the cloud environment associated with entity A to another process on a virtual machine within the cloud environment associated with entity A). In other cases, a process may be in communication with a node (e.g., outside of entity A) which does not have an agent deployed upon it. As one example, a node within entity A might be in communication with node 172, outside of entity A. In such a scenario, communications with node 172 are modeled (e.g., by graph generator 146) using the IP address of node 172. Similarly, where a node within entity A does not have an agent deployed upon it, the IP address of the node can be used by graph generator in modeling.

~~[00189]~~[00190] Graphs created by graph generator 146 may be written to data store 30 and cached for further processing. A graph may be a summary of all activity that happened in a particular time interval. As each graph corresponds to a distinct period of time, different rows can be aggregated to find summary information over a larger timestamp. In some examples, picking two different graphs from two different timestamps can be used to compare different periods. If necessary, graph generator can parallelize its workload (e.g., where its backlog cannot otherwise be handled within a particular time period, such as an hour, or if is required to process a graph spanning a long time period).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00042

~~[00190]~~[00191] Graph generator 146 can be implemented in any appropriate programming language, such as Java or C, and machine learning libraries, such as Spark's MLLib. Example ways that graph generator computations can be implemented include using SQL or Map-R, using Spark or Hadoop.

~~[00191]~~[00192] SSH tracker 148 is a microservice that may be responsible for following ssh connections and process parent hierarchies to determine trails of user ssh activity. Identified ssh trails are placed by the SSH tracker 148 into data store 30 and cached for further processing.

~~[00192]~~[00193] SSH tracker 148 can be implemented in any appropriate programming language, such as Java or C, and machine libraries, such as Spark's MLLib. Example ways that SSH tracker computations can be implemented include using SQL or Map-R, using Spark or Hadoop.

~~[00193]~~[00194] Threat aggregator 150 is a microservice that may be responsible for obtaining third party threat information from various applicable sources, and making it available to other micro-services. Examples of such information include reverse DNS information, GeoIP information, lists of known bad domains/IP addresses, lists of known bad files etc. As applicable, the threat information is normalized before insertion into data store 30. Threat aggregator 150 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 (e.g., for insertions and queries).

~~[00194]~~[00195] Scheduler 152 is a microservice that may act as a scheduler and that may run arbitrary jobs organized as a directed graph. In some examples, scheduler 152 ensures that all jobs for all customers are able to run during at a given time interval (e.g., every hour). Scheduler 152 may handle errors and retrying for failed jobs, track dependencies, manage appropriate resource levels, and/or scale jobs as needed. Scheduler 152 can be implemented in any appropriate programming language, such as Java or C. A variety of components can also be used, such as open source scheduler frameworks (e.g., Airflow), or AWS services (e.g., the AWS Data pipeline) which can be used for managing schedules.

~~[00195]~~[00196] Graph Behavior Modeler (GBM) 154 is a microservice that may compute polygraphs. In particular, GBM 154 can be used to find clusters of nodes in a graph that should be considered similar based on some set of their properties and relationships to other nodes. As described herein, the clusters and their relationships can be used to provide visibility into a datacenter environment without requiring user specified labels. GBM 154 may track such clusters over time persistently, allowing for changes to be detected and alerts to be generated.

~~[00196]~~[00197] GBM 154 may take as input a raw graph (e.g., as generated by graph generator 146). Nodes are actors of a behavior, and edges are the behavior relationship itself. For example, in the case of communication, example actors include processes, which communicate with other processes. The GBM 154 clusters the raw graph based on behaviors of actors and produces a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00043

summary (the polygraph). The polygraph summarizes behavior at a datacenter level. The GBM also produces "observations" that represent changes detected in the datacenter. Such observations may be based on differences in cumulative behavior (e.g., the baseline) of the datacenter with its current behavior. The GBM 154 can be implemented in any appropriate programming language, such as Java, C, or Golang, using appropriate libraries (as applicable) to handle distributed graph computations (handling large amounts of data analysis in a short amount of time). Apache Spark is another example tool that can be used to compute polygraphs. The GBM can also take feedback from users and adjust the model according to that feedback. For example, if a given user is interested in relearning behavior for a particular entity, the GBM can be instructed to "forget" the implicated part of the polygraph.

~~[00197]~~[00198] GBM runner 156 is a microservice that may be responsible for interfacing with GBM 154 and providing GBM 154 with raw graphs (e.g., using a query language, such as SQL, to push any computations it can to data store 30). GBM runner 156 may also insert polygraph output from GBM 154 to data store 30. GBM runner 156 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 to insert and query data.

~~[00198]~~[00199] Alert generator 158 is a microservice that may be responsible for generating alerts. Alert generator 158 may examine observations (e.g., produced by GBM 154) in aggregate, deduplicate them, and score them. Alerts may be generated for observations with a score exceeding a threshold. Alert generator 158 may also compute (or retrieves, as applicable) data that a customer (e.g., user A or user B) might need when reviewing the alert. Examples of events that can be detected by data platform 12 (and alerted on by alert generator 158) include, but are not limited to the following:

~~[00199]~~[00200] new user: This event may be created the first time a user (e.g., of node 116) is first observed by an agent within a datacenter.

~~[00200]~~[00201] user launched new binary: This event may be generated when an interactive user launches an application for the first time.

~~[00201]~~[00202] new privilege escalation: This event may be generated when user privileges are escalated and a new application is run.

~~[00202]~~[00203] new application or container: This event may be generated when an application or container is seen for the first time.

~~[00203]~~[00204] new external connection: This event may be generated when a connection to an external IP/domain is made from a new application.

~~[00204]~~[00205] new external host or IP: This event may be generated when a new external host or IP is involved in a connection with a datacenter.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00044

~~[00205]~~[00206] new internal connection: This event may be generated when a connection between internal-only applications is seen for the first time.

~~[00206]~~[00207] new external client: This event may be generated when a new external connection is seen for an application which typically does not have external connections.

~~[00207]~~[00208] new parent: This event may be generated when an application is launched by a different parent.

~~[00208]~~[00209] connection to known bad IP/domain: Data platform 12 maintains (or can otherwise access) one or more reputation feeds. If an environment makes a connection to a known bad IP or domain, an event will be generated.

~~[00209]~~[00210] login from a known bad IP/domain: An event may be generated when a successful connection to a datacenter from a known bad IP is observed by data platform 12.

~~[00210]~~[00211] Alert generator 158 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 to insert and query data. In various embodiments, alert generator 158 also uses one or more machine learning libraries, such as Spark's MLLib (e.g., to compute scoring of various observations). Alert generator 158 can also take feedback from users about which kinds of events are of interest and which to suppress.

~~[00211]~~[00212] QsJobServer 160 is a microservice that may look at all the data produced by data platform 12 for an hour, and compile a materialized view (MV) out of the data to make queries faster. The MV helps make sure that the queries customers most frequently run, and data that they search for, can be easily queried and answered. QsJobServer 160 may also precompute and cache a variety of different metrics so that they can quickly be provided as answers at query time. QsJobServer 160 can be implemented using any appropriate programming language, such as Java or C, using SQL/JDBC libraries. In some examples, QsJobServer 160 is able to compute an MV efficiently at scale, where there could be a large number of joins. An SQL engine, such as Oracle, can be used to efficiently execute the SQL, as applicable.

~~[00212]~~[00213] Alert notifier 162 is a microservice that may take alerts produced by alert generator 158 and send them to customers' integrated Security Information and Event Management (SIEM) products (e.g., Splunk, Slack, etc.). Alert notifier 162 can be implemented using any appropriate programming language, such as Java or C. Alert notifier 162 can be configured to use an email service (e.g., AWS SES or pagerduty) to send emails. Alert notifier 162 may also provide templating support (e.g., Velocity or Moustache) to manage templates and structured notifications to SIEM products.

~~[00213]~~[00214] Reporting module 164 is a microservice that may be responsible for creating reports out of customer data (e.g., daily summaries of events, etc.) and providing those reports to

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00045

customers (e.g., via email). Reporting module 164 can be implemented using any appropriate programming language, such as Java or C. Reporting module 164 can be configured to use an email service (e.g., AWS SES or pagerduty) to send emails. Reporting module 164 may also provide templating support (e.g., Velocity or Moustache) to manage templates (e.g., for constructing HTML-based email).

~~[00214]~~[00215] Web app 120 is a microservice that provides a user interface to data collected and processed on data platform 12. Web app 120 may provide login, authentication, query, data visualization, etc. features. Web app 120 may, in some embodiments, include both client and server elements. Example ways the server elements can be implemented are using Java DropWizard or Node.Js to serve business logic, and a combination of JSON/HTTP to manage the service. Example ways the client elements can be implemented are using frameworks such as React, Angular, or Backbone. JSON, jQuery, and JavaScript libraries (e.g., underscore) can also be used.

~~[00215]~~[00216] Query service 166 is a microservice that may manage all database access for web app 120. Query service 166 abstracts out data obtained from data store 30 and provides a JSON-based REST API service to web app 120. Query service 166 may generate SQL queries for the REST APIs that it receives at run time. Query service 166 can be implemented using any appropriate programming language, such as Java or C and SQL/JDBC libraries, or an SQL framework such as jOOQ. Query service 166 can internally make use of a variety of types of databases, including a relational database engine 168 (e.g., AWS Aurora) and/or data store 30 to manage data for clients. Examples of tables that query service 166 manages are OLTP tables and data warehousing tables.

~~[00216]~~[00217] Cache 170 may be implemented by Redis and/or any other service that provides a key-value store. Data platform 12 can use cache 170 to keep information for frontend services about users. Examples of such information include valid tokens for a customer, valid cookies of customers, the last time a customer tried to login, etc.

~~[00217]~~[00218] Fig. 3A illustrates an example of a process for detecting anomalies in a network environment. In various embodiments, process 300 is performed by data platform 12. The process begins at 301 when data associated with activities occurring in a network environment (such as entity A's datacenter) is received. One example of such data that can be received at 301 is agent-collected data described above (e.g., in conjunction with process 200).

~~[00218]~~[00219] At 302, a logical graph model is generated, using at least a portion of the monitored activities. A variety of approaches can be used to generate such logical graph models, and a variety of logical graphs can be generated (whether using the same, or different

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00046

approaches). The following is one example of how data received at 301 can be used to generate and maintain a model.

~~[00219]~~[00220] During bootstrap, data platform 12 creates an aggregate graph of physical connections (also referred to herein as an aggregated physical graph) by matching connections that occurred in the first hour into communication pairs. Clustering is then performed on the communication pairs. Examples of such clustering, described in more detail below, include performing Matching Neighbor clustering and similarity (e.g., SimRank) clustering. Additional processing can also be performed (and is described in more detail below), such as by splitting clusters based on application type, and annotating nodes with DNS query information. The resulting graph (also referred to herein as a base graph or common graph) can be used to generate a variety of models, where a subset of node and edge types (described in more detail below) and their properties are considered in a given model. One example of a model is a UID to UID model (also referred to herein as a Uid2Uid model) which clusters together processes that share a username and show similar privilege change behavior. Another example of a model is a CType model, which clusters together processes that share command line similarity. Yet another example of a model is a PType model, which clusters together processes that share behaviors over time.

~~[00220]~~[00221] Each hour (or any other predetermined time interval) after bootstrap, a new snapshot is taken (i.e., data collected about a datacenter in the last hour is processed) and information from the new snapshot is merged with existing data to create and (as additional data is collected/processed) maintain a cumulative graph. The cumulative graph (also referred to herein as a cumulative PType graph and a polygraph) is a running model of how processes behave over time. Nodes in the cumulative graph are PType nodes, and provide information such as a list of all active processes and PIDs in the last hour, the number of historic total processes, the average number of active processes per hour, the application type of the process (e.g., the CType of the PType), and historic CType information/frequency. Edges in the cumulative graph can represent connectivity and provide information such as connectivity frequency. The edges can be weighted (e.g., based on number of connections, number of bytes exchanged, etc.). Edges in the cumulative graph (and snapshots) can also represent transitions.

~~[00221]~~[00222] One approach to merging a snapshot of the activity of the last hour into a cumulative graph is as follows. An aggregate graph of physical connections is made for the connections included in the snapshot (as was previously done for the original snapshot used during bootstrap). And, clustering/splitting is similarly performed on the snapshot's aggregate graph. Next, PType clusters in the snapshot's graph are compared against PType clusters in the cumulative graph to identify commonality.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00047

~~[00222]~~[00223] One approach to determining commonality is, for any two nodes that are members of a given CmdType (described in more detail below), comparing internal neighbors and calculating a set membership Jaccard distance. The pairs of nodes are then ordered by decreasing similarity (i.e., with the most similar sets first). For nodes with a threshold amount of commonality (e.g., at least 66% members in common), any new nodes (i.e., appearing in the snapshot's graph but not the cumulative graph) are assigned the same PType identifier as is assigned to the corresponding node in the cumulative graph. For each node that is not classified (i.e., has not been assigned a PType identifier), a network signature is generated (i.e., indicative of the kinds of network connections the node makes, who the node communicates with, etc.). The following processing is then performed until convergence. If a match of the network signature is found in the cumulative graph, the unclassified node is assigned the PType identifier of the corresponding node in the cumulative graph. Any nodes which remain unclassified after convergence are new PTypes and are assigned new identifiers and added to the cumulative graph as new. As applicable, the detection of a new PType can be used to generate an alert. If the new PType has a new CmdType, a severity of the alert can be increased. If any surviving nodes (i.e., present in both the cumulative graph and the snapshot graph) change PTypes, such change is noted as a transition, and an alert can be generated. Further, if a surviving node changes PType and also changes CmdType, a severity of the alert can be increased.

~~[00223]~~[00224] Changes to the cumulative graph (e.g., a new PType or a new edge between two PTypes) can be used (e.g., at 303) to detect anomalies (described in more detail below). Two example kinds of anomalies that can be detected by data platform 12 include security anomalies (e.g., a user or process behaving in an unexpected manner) and devops/root cause anomalies (e.g., network congestion, application failure, etc.). Detected anomalies can be recorded and surfaced (e.g., to administrators, auditors, etc.), such as through alerts which are generated at 304 based on anomaly detection.

~~[00224]~~[00225] Additional detail regarding processing performed, by various components depicted in Fig. 1D (whether performed individually or in combination), in conjunction with model/polygraph construction (e.g., as performed at 302) are provided below.

~~[00225]~~[00226] As explained above, an aggregated physical graph can be generated on a per customer basis periodically (e.g., once an hour) from raw physical graph information, by matching connections (e.g., between two processes on two virtual machines). In various embodiments, a deterministic fixed approach is used to cluster nodes in the aggregated physical graph (e.g., representing processes and their communications). As one example, Matching Neighbors Clustering (MNC) can be performed on the aggregated physical graph to determine which entities exhibit identical behavior and cluster such entities together.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00048

~~[00226]~~[00227] Fig. 3B depicts a set of example processes (p1, p2, p3, and p4) communicating with other processes (p10 and p11). Fig. 3B is a graphical representation of a small portion of an aggregated physical graph showing (for a given time period, such as an hour) which processes in a datacenter communicate with which other processes. Using MNC, processes p1, p2, and p3 will be clustered together (305), as they exhibit identical behavior (they communicate with p10 and only p10). Process p4, which communicates with both p10 and p11, will be clustered separately.

~~[00227]~~[00228] In MNC, only those processes exhibiting identical (communication) behavior will be clustered. In various embodiments, an alternate clustering approach can also/instead be used, which uses a similarity measure (e.g., constrained by a threshold value, such as a 60% similarity) to cluster items. In some embodiments, the output of MNC is used as input to SimRank, in other embodiments, MNC is omitted.

~~[00228]~~[00229] Fig. 3C depicts a set of example processes (p4, p5, p6) communicating with other processes (p7, p8, p9). As illustrated, most of nodes p4, p5, and p6 communicate with most of nodes p7, p8, and p9 (as indicated in Fig. 3C with solid connection lines). As one example, process p4 communicates with process p7 (310), process p8 (311), and process p9 (312). An exception is process p6, which communicates with processes p7 and p8, but does not communicate with process p9 (as indicated by dashed line 313). If MNC were applied to the nodes depicted in Fig. 3C, nodes p4 and p5 would be clustered (and node p6 would not be included in their cluster).

~~[00229]~~[00230] One approach to similarity clustering is to use SimRank. In an embodiment of the SimRank approach, for a given node v in a directed graph, $I(v)$ and $O(v)$ denote the respective set of in-neighbors and out-neighbors of v. Individual in-neighbors are denoted as $I_i(v)$, for $1 \leq i \leq |I(v)|$, and individual out-neighbors are denoted as $O_i(v)$, for $1 \leq i \leq |O(v)|$. The similarity between two objects a and b can be denoted by $s(a,b) \in [1,0]$. A recursive equation (hereinafter "the SimRank equation") can be written for $s(a,b)$, where, if a=b, then $s(a,b)$ is defined as 1, otherwise, $s(a,b)=\frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$ where C is a constant between 0 and 1. One example value for the decay factor C is 0.8 (and a fixed number of iterations such as five). Another example value for the decay factor C is 0.6 (and/or a different number of iterations). In the event that a or b has no in-neighbors, similarity is set to $s(a,b)=0$, so the summation is defined to be 0 when $I(a)=\emptyset$ or $I(b)=\emptyset$.

~~[00230]~~[00231] The SimRank equations for a graph G can be solved by iteration to a fixed point. Suppose n is the number of nodes in G. For each iteration k, n^2 entries $s_k(*,*)$ are kept, where $s_k(a,b)$ gives the score between a and b on iteration k. Successive computations of $s_{k+1}(*,*)$ are

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00049

made based on $s_k(*,*)$. Starting with $s_0(*,*)$, where each $s_0(a,b)$ is a lower bound on the actual SimRank score $s(a,b)$: $s_0(a,b) = \begin{cases} 1, \text{if } a = b, \\ 0, \text{if } a \neq b. \end{cases}$

~~[00231]~~[00232] The SimRank equation can be used to compute $s_{k+1}(a,b)$ from $s_k(*,*)$ with $s_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s_k(I_i(a), I_j(b))$ for $a \neq b$, and $s_{k+1}(a,b)=1$ for $a=b$. On each iteration k+1, the similarity of (a,b) is updated using the similarity scores of the neighbors of (a,b) from the previous iteration k according to the SimRank equation. The values $s_k(*,*)$ are nondecreasing as k increases.

~~[00232]~~[00233] Returning to Fig. 3C, while MNC would cluster nodes p4 and p5 together (and not include node p6 in their cluster), application of SimRank would cluster nodes p4-p6 into one cluster (314) and also cluster nodes p7-p9 into another cluster (315).

~~[00233]~~[00234] Fig. 3D depicts a set of processes, and in particular server processes s1 and s2, and client processes c1, c2, c3, c4, c5, and c6. Suppose only nodes s1, s2, c1, and c2 are present in the graph depicted in Fig. 3D (and the other nodes depicted are omitted from consideration). Using MNC, nodes s1 and s2 would be clustered together, as would nodes c1 and c2. Performing SimRank clustering as described above would also result in those two clusters (s1 and s2, and c1 and c2). As previously mentioned, in MNC, identical behavior is required. Thus, if node c3 were now also present in the graph, MNC would not include c3 in a cluster with c2 and c1 because node c3 only communicates with node s2 and not node s1. In contrast, a SimRank clustering of a graph that includes nodes s1, s2, c1, c2, and c3 would result (based, e.g., on an applicable selected decay value and number of iterations) in a first cluster comprising nodes s1 and s2, and a second cluster of c1, c2, and c3. As an increasing number of nodes which communicate with server process s2, and do not also communicate with server process s1, are included in the graph (e.g., as c4, c5, and c6 are added), under SimRank, nodes s1 and s2 will become decreasingly similar (i.e., their intersection is reduced).

~~[00234]~~[00235] In various embodiments, SimRank is modified (from what is described above) to accommodate differences between the asymmetry of client and server connections. As one example, SimRank can be modified to use different thresholds for client communications (e.g., an 80% match among nodes c1-c6) and for server communications (e.g., a 60% match among nodes s1 and s2). Such modification can also help achieve convergence in situations such as where a server process dies on one node and restarts on another node.

~~[00235]~~[00236] The application of MNC/SimRank to an aggregated physical graph results in a smaller graph, in which processes which are determined to be sufficiently similar are clustered together. Typically, clusters generated as output of MNC will be underinclusive. For example, for the nodes depicted in Fig. 3C, process p6 will not be included in a cluster with processes p4

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00050

and p5, despite substantial similarity in their communication behaviors. The application of SimRank (e.g., to the output of MNC) helps mitigate the underinclusiveness of MNC, but can result in overly inclusive clusters. As one example, suppose (returning to the nodes depicted in Fig. 3B) that as a result of applying SimRank to the depicted nodes, nodes p1-p4 are all included in a single cluster. Both MNC and SimRank operate agnostically of which application a given process belongs to. Suppose processes p1-p3 each correspond to a first application (e.g., an update engine), and process p4 corresponds to a second application (e.g., sshd). Further suppose process p10 corresponds to contact with AWS. Clustering all four of the processes together (e.g., as a result of SimRank) could be problematic, particularly in a security context (e.g., where granular information useful in detecting threats would be lost).

~~[00236]~~[00237] As previously mentioned, data platform 12 may maintain a mapping between processes and the applications to which they belong. In various embodiments, the output of SimRank (e.g., SimRank clusters) is split based on the applications to which cluster members belong (such a split is also referred to herein as a "CmdType split"). If all cluster members share a common application, the cluster remains. If different cluster members originate from different applications, the cluster members are split along application-type (CmdType) lines. Using the nodes depicted in Fig. 3D as an example, suppose that nodes c1, c2, c3, and c5 all share "update engine" as the type of application to which they belong (sharing a CmdType). Suppose that node c4 belongs to "ssh," and suppose that node c6 belongs to "bash." As a result of SimRank, all six nodes (c1-c6) might be clustered into a single cluster. After a CmdType split is performed on the cluster, however, the single cluster will be broken into three clusters (c1, c2, c3, c5; c4; and c6). Specifically, the resulting clusters comprise processes associated with the same type of application, which exhibit similar behaviors (e.g., communication behaviors). Each of the three clusters resulting from the CmdType split represents, respectively, a node (also referred to herein as a PType) of a particular CmdType. Each PType is given a persistent identifier and stored persistently as a cumulative graph.

~~[00237]~~[00238] A variety of approaches can be used to determine a CmdType for a given process. As one example, for some applications (e.g., sshd), a one-to-one mapping exists between the CmdType and the application/binary name. Thus, processes corresponding to the execution of sshd will be classified using a CmdType of sshd. In various embodiments, a list of common application/binary names (e.g., sshd, apache, etc.) is maintained by data platform 12 and manually curated as applicable. Other types of applications (e.g., Java, Python, and Ruby) are multi-homed, meaning that several very different applications may all execute using the binary name, "java." For these types of applications, information such as command line / execution path information can be used in determining a CmdType. In particular, the subapplication can be used

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00051

as the CmdType of the application, and/or term frequency analysis (e.g., TF/IDF) can be used on command line information to group, for example, any marathon related applications together (e.g., as a python.marathon CmdType) and separately from other Python applications (e.g., as a python.airflow CmdType).

~~[00238]~~[00239] In various embodiments, machine learning techniques are used to determine a CmdType. The CmdType model is constrained such that the execution path for each CmdType is unique. One example approach to making a CmdType model is a random forest based approach. An initial CmdType model is bootstrapped using process parameters (e.g., available within one minute of process startup) obtained using one hour of information for a given customer (e.g., entity A). Examples of such parameters include the command line of the process, the command line of the process's parent(s) (if applicable), the uptime of the process, UID/EUID and any change information, TTY and any change information, listening ports, and children (if any). Another approach is to perform term frequency clustering over command line information to convert command lines into cluster identifiers.

~~[00239]~~[00240] The random forest model can be used (e.g., in subsequent hours) to predict a CmdType for a process (e.g., based on features of the process). If a match is found, the process can be assigned the matching CmdType. If a match is not found, a comparison between features of the process and its nearest CmdType (e.g., as determined using a Levenstein distance) can be performed. The existing CmdType can be expanded to include the process, or, as applicable, a new CmdType can be created (and other actions taken, such as generating an alert). Another approach to handling processes which do not match an existing CmdType is to designate such processes as unclassified, and once an hour, create a new random forest seeded with process information from a sampling of classified processes (e.g., 10 or 100 processes per CmdType) and the new processes. If a given new process winds up in an existing set, the process is given the corresponding CmdType. If a new cluster is created, a new CmdType can be created.

~~[00240]~~[00241] Conceptually, a polygraph represents the smallest possible graph of clusters that preserve a set of rules (e.g., in which nodes included in the cluster must share a CmdType and behavior). As a result of performing MNC, SimRank, and cluster splitting (e.g., CmdType splitting) many processes are clustered together based on commonality of behavior (e.g., communication behavior) and commonality of application type. Such clustering represents a significant reduction in graph size (e.g., compared to the original raw physical graph). Nonetheless, further clustering can be performed (e.g., by iterating on the graph data using the GBM to achieve such a polygraph). As more information within the graph is correlated, more nodes can be clustered together, reducing the size of the graph, until convergence is reached and no further clustering is possible.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00052

~~[00241]~~[00242] Fig. 3E depicts two pairs of clusters. In particular, cluster 320 represents a set of client processes sharing the same CmdType ("a1"), communicating (collectively) with a server process having a CmdType ("a2"). Cluster 322 also represents a set of client processes having a CmdType a1 communicating with a server process having a CmdType a2. The nodes in clusters 320 and 322 (and similarly nodes in 321 and 323) remain separately clustered (as depicted) after MNC/SimRank/CmdType splitting – isolated islands. One reason this could occur is where server process 321 corresponds to processes executing on a first machine (having an IP address of 1.1.1.1). The machine fails and a new server process 323 starts, on a second machine (having an IP address of 2.2.2.2) and takes over for process 321.

~~[00242]~~[00243] Communications between a cluster of nodes (e.g., nodes of cluster 320) and the first IP address can be considered different behavior from communications between the same set of nodes and the second IP address, and thus communications 324 and 325 will not be combined by MNC/SimRank in various embodiments. Nonetheless, it could be desirable for nodes of clusters 320/322 to be combined (into cluster 326), and for nodes of clusters 321/323 to be combined (into cluster 327), as representing (collectively) communications between a1 and a2. One task that can be performed by data platform 12 is to use DNS query information to map IP addresses to logical entities. As will be described in more detail below, GBM 154 can make use of the DNS query information to determine that graph nodes of cluster 320 and graph nodes of cluster 322 both made DNS queries for "appserverabc.example.com," which first resolved to 1.1.1.1 and then to 2.2.2.2, and to combine nodes 320/322 and 321/323 together into a single pair of nodes (326 communicating with 327).

~~[00243]~~[00244] In various embodiments, GBM 154 operates in a batch manner in which it receives as input the nodes and edges of a graph for a particular time period along with its previous state, and generates as output clustered nodes, cluster membership edges, cluster-to-cluster edges, events, and its next state.

~~[00244]~~[00245] GBM 154 may not try to consider all types of entities and their relationships that may be available in a conceptual common graph all at once. Instead, GBM uses a concept of models where a subset of node and edge types and their properties are considered in a given model. Such an approach is helpful for scalability, and also to help preserve detailed information (of particular importance in a security context) – as clustering entities in a more complex and larger graph could result in less useful results. In particular, such an approach allows for different types of relationships between entities to be preserved/more easily analyzed.

~~[00245]~~[00246] While GBM 154 can be used with different models corresponding to different subgraphs, core abstractions remain the same across types of models.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00053

~~[00246]~~[00247] For example, each node type in a GBM model is considered to belong to a class. The class can be thought of as a way for the GBM to split nodes based on the criteria it uses for the model. The class for a node is represented as a string whose value is derived from the node's key and properties depending on the GBM Model. Note that different GBM models may create different class values for the same node. For each node type in a given GBM model, GBM 154 can generate clusters of nodes for that type. A GBM generated cluster for a given member node type cannot span more than one class for that node type. GBM 154 generates edges between clusters that have the same types as the edges between source and destination cluster node types.

~~[00247]~~[00248] Additionally or alternatively, the processes described herein as being used for a particular model can be used (can be the same) across models, and different models can also be configured with different settings.

~~[00248]~~[00249] Additionally or alternatively, the node types and the edge types may correspond to existing types in the common graph node and edge tables but this is not necessary. Even when there is a correspondence, the properties provided to GBM 154 are not limited to the properties that are stored in the corresponding graph table entries. They can be enriched with additional information before being passed to GBM 154.

~~[00249]~~[00250] Logically, the input for a GBM model can be characterized in a manner that is similar to other graphs. Edge triplets can be expressed, for example, as an array of source node type, edge type, and destination node type. And, each node type is associated with node properties, and each edge type is associated with edge properties. Other edge triplets can also be used (and/or edge triplets can be extended) in accordance with various embodiments.

~~[00250]~~[00251] Note that the physical input to the GBM model need not (and does not, in various embodiments) conform to the logical input. For example, the edges in the PtypeConn model correspond to edges between Matching Neighbors (MN) clusters, where each process node has an MN cluster identifier property. In the User ID to User ID model (also referred to herein as the Uid2Uid model), edges are not explicitly provided separately from nodes (as the euid array in the node properties serves the same purpose). In both cases, however, the physical information provides the applicable information necessary for the logical input.

~~[00251]~~[00252] The state input for a particular GBM model can be stored in a file, a database, or other appropriate storage. The state file (from a previous run) is provided, along with graph data, except for when the first run for a given model is performed, or the model is reset. In some cases, no data may be available for a particular model in a given time period, and GBM may not be run for that time period. As data becomes available at a future time, GBM can run using the latest state file as input.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00054

~~[00252]~~[00253] GBM 154 outputs cluster nodes, cluster membership edges, and inter-cluster relationship edges that are stored (in some embodiments) in the graph node tables: node_c, node_cm, and node_icr, respectively. The type names of nodes and edges may conform to the following rules:

~~[00253]~~[00254] * A given node type can be used in multiple different GBM models. The type names of the cluster nodes generated by two such models for that node type will be different. For instance, process type nodes will appear in both PtypeConn and Uid2Uid models, but their cluster nodes will have different type names.

~~[00254]~~[00255] * The membership edge type name is "MemberOf."

~~[00255]~~[00256] * The edge type names for cluster-to-cluster edges will be the same as the edge type names in the underlying node-to-node edges in the input.

~~[00256]~~[00257] The following are example events GBM 154 can generate: new class, new cluster, new edge from class to class, split class (the notion that GBM 154 considers all nodes of a given type and class to be in the same cluster initially and if GBM 154 splits them into multiple clusters, it is splitting a class), new edge from cluster and class, new edge between cluster and cluster, and/or new edge from class to cluster.

~~[00257]~~[00258] One underlying node or edge in the logical input can cause multiple types of events to be generated. Conversely, one event can correspond to multiple nodes or edges in the input. Not every model generates every event type.

~~[00258]~~[00259] Additional information regarding examples of data structures/models that can be used in conjunction with models used by data platform 12 is now provided.

~~[00259]~~[00260] In some examples, a PTypeConn Model clusters nodes of the same class that have similar connectivity relationships. For example, if two processes had similar incoming neighbors of the same class and outgoing neighbors of the same class, they could be clustered.

~~[00260]~~[00261] The node input to the PTypeConn model for a given time period includes non-interactive (i.e., not associated with tty) process nodes that had connections in the time period and the base graph nodes of other types (IP Service Endpoint (IPSep) comprising an IP address and a port), DNS Service Endpoint (DNSSep) and IPAddress) that have been involved in those connections. The base relationship is the connectivity relationship for the following type triplets:

~~[00261]~~[00262] * Process, ConnectedTo, Process

~~[00262]~~[00263] * Process, ConnectedTo, IP Service Endpoint (IPSep)

~~[00263]~~[00264] * Process, ConnectedTo, DNS Service Endpoint (DNSSep)

~~[00264]~~[00265] * IPAddress, ConnectedTo, ProcessProcess, DNS, ConnectedTo, Process

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00055

~~[00265]~~[00266] The edge inputs to this model are the ConnectedTo edges from the MN cluster, instead of individual node-to-node ConnectedTo edges from the base graph. The membership edges created by this model refer to the base graph node type provided in the input.

~~[00266]~~[00267] Class Values:

~~[00267]~~[00268] The class values of nodes are determined as follows depending on the node type (e.g., Process nodes, IPSep nodes, DNSSep nodes, and IP Address nodes).

~~[00268]~~[00269] Process nodes:

~~[00269]~~[00270] if exe_path contains java then "java <cmdline_term_1> ..."

~~[00270]~~[00271] else if exe_path contains python then "python <cmdline_term_1> ..."

~~[00271]~~[00272] else "last_part_of_exe_path"

~~[00272]~~[00273] IPSep nodes:

~~[00273]~~[00274] if IP_internal then "IntIPS"

~~[00274]~~[00275] else if severity = 0 then "<IP_addr>:<protocol>:<port>"

~~[00275]~~[00276] else "<IP_addr>:<port>_BadIP"

~~[00276]~~[00277] DNSSep nodes:

~~[00277]~~[00278] if IP_internal = 1 then "<hostname>"

~~[00278]~~[00279] else if severity = 0 then "<hostname>:<protocol>:port"

~~[00279]~~[00280] else "<hostname>:<port>_BadIP"

~~[00280]~~[00281] IPAddress nodes (will appear only on client side):

~~[00281]~~[00282] if IP_internal = 1 then "IPIntC"

~~[00282]~~[00283] else if severity = 0 then "ExtIPC"

~~[00283]~~[00284] else "ExtBadIPC"

~~[00284]~~[00285] Events:

~~[00285]~~[00286] A new class event in this model for a process node is equivalent to seeing a new CType being involved in a connection for the first time. Note that this does not mean the CType was not seen before. It is possible that it was previously seen but did not make a connection at that time.

~~[00286]~~[00287] A new class event in this model for an IPSep node with IP_internal = 0 is equivalent to seeing a connection to a new external IP address for the first time.

~~[00287]~~[00288] A new class event in this model for a DNSSep node is equivalent to seeing a connection to a new domain for the first time.

~~[00288]~~[00289] A new class event in this model for an IPAddress node with IP_internal = 0 and severity = 0 is equivalent to seeing a connection from any external IP address for the first time.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00056

~~[00289]~~[00290] A new class event in this model for an IPAddress node with IP_internal = 0 and severity > 0 is equivalent to seeing a connection from any bad external IP address for the first time.

~~[00290]~~[00291] A new class to class to edge from a class for a process node to a class for a process node is equivalent to seeing a communication from the source CType making a connection to the destination CType for the first time.

~~[00291]~~[00292] A new class to class to edge from a class for a process node to a class for a DNSSep node is equivalent to seeing a communication from the source CType making a connection to the destination domain name for the first time.

~~[00292]~~[00293] An IntPConn Model may be similar to the PtypeConn Model, except that connection edges between parent/child processes and connections between processes where both sides are not interactive are filtered out.

~~[00293]~~[00294] A Uid2Uid Model may cluster processes with the same username that show similar privilege change behavior. For instance, if two processes with the same username had similar effective user values, launched processes with similar usernames, and were launched by processes with similar usernames, then they could be clustered.

~~[00294]~~[00295] An edge between a source cluster and destination cluster generated by this model means that all of the processes in the source cluster had a privilege change relationship to at least one process in the destination cluster.

~~[00295]~~[00296] The node input to this model for a given time period includes process nodes that are running in that period. The value of a class of process nodes is "<username>".

~~[00296]~~[00297] The base relationship that is used for clustering is privilege change, either by the process changing its effective user ID, or by launching a child process which runs with a different user.

~~[00297]~~[00298] The physical input for this model includes process nodes (only), with the caveat that the complete ancestor hierarchy of process nodes active (i.e., running) for a given time period is provided as input even if an ancestor is not active in that time period. Note that effective user IDs of a process are represented as an array in the process node properties, and launch relationships are available from ppid_hash fields in the properties as well.

~~[00298]~~[00299] A new class event in this model is equivalent to seeing a user for the first time.

~~[00299]~~[00300] A new class to class edge event is equivalent to seeing the source user making a privilege change to the destination user for the first time.

~~[00300]~~[00301] A Ct2Ct Model may cluster processes with the same CType that show similar launch behavior. For instance, if two processes with the same CType have launched processes with similar CTypes, then they could be clustered.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00057

~~[00301]~~[00302] The node input to this model for a given time period includes process nodes that are running in that period. The value class of process nodes is CType (similar to how it is created for the PtypeConn Model).

~~[00302]~~[00303] The base relationship that is used for clustering is a parent process with a given CType launching a child process with another given destination CType.

~~[00303]~~[00304] The physical input for this model includes process nodes (only) with the caveat that the complete ancestor hierarchy active process nodes (i.e., that are running) for a given time period is provided as input even if an ancestor is not active in that time period. Note that launch relationships are available from ppid_hash fields in the process node properties.

~~[00304]~~[00305] An edge between a source cluster and destination cluster generated by this model means that all of the processes in the source cluster launched at least one process in the destination cluster.

~~[00305]~~[00306] A new class event in this model is equivalent to seeing a CType for the first time. Note that the same type of event will be generated by the PtypeConn Model as well.

~~[00306]~~[00307] A new class to class edge event is equivalent to seeing the source CType launching the destination CType for the first time.

~~[00307]~~[00308] An MTypeConn Model may cluster nodes of the same class that have similar connectivity relationships. For example, if two machines had similar incoming neighbors of the same class and outgoing neighbors of the same class, they could be clustered.

~~[00308]~~[00309] A new class event in this model will be generated for external IP addresses or (as applicable) domain names seen for the first time. Note that a new class to class to edge Machine, class to class for an IPSep or DNSName node will also be generated at the same time.

~~[00309]~~[00310] The membership edges generated by this model will refer to Machine, IPAddress, DNSName, and IPSep nodes in the base graph. Though the nodes provided to this model are IPAddress nodes instead of IPSep nodes, the membership edges it generates will refer to IPSep type nodes. Alternatively, the base graph can generate edges between Machine and IPSep node types. Note that the Machine to IPAddress edges have tcp_dst_ports/udp_dst_ports properties that can be used for this purpose.

~~[00310]~~[00311] The node input to this model for a given time period includes machine nodes that had connections in the time period and the base graph nodes of other types (IPAddress and DNSName) that were involved in those connections.

~~[00311]~~[00312] The base relationship is the connectivity relationship for the following type triplets:

~~[00312]~~[00313] * Machine, ConnectedTo, Machine

~~[00313]~~[00314] * Machine, ConnectedTo, IPAddress

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00058

[00314][00315] * Machine, ConnectedTo, DNSName

[00315][00316] * IPAddress, ConnectedTo, Machine, DNS, ConnectedTo, Machine

[00316][00317] The edge inputs to this model are the corresponding ConnectedTo edges in the base graph.

[00317][00318] Class Values:

[00318][00319] Machine:

[00319][00320] The class value for all Machine nodes is "Machine."

[00320][00321] The machine_terms property in the Machine nodes is used, in various embodiments, for labeling machines that are clustered together. If a majority of the machines clustered together share a term in the machine_terms, that term can be used for labeling the cluster.

[00321][00322] IPSep:

[00322][00323] The class value for IPSep nodes is determined as follows:

[00323][00324] if IP_internal then "IntIPS"

[00324][00325] else

[00325][00326] if severity = 0 then "<ip_addr>:<protocol>:<port>"

[00326][00327] else "<IP_addr_BadIP>"

[00327][00328] IPAddress:

[00328][00329] The class value for IpAddress nodes is determined as follows:

[00329][00330] if IP_internal then "IntIPC"

[00330][00331] else

[00331][00332] if severity = 0 then "ExtIPC"

[00332][00333] else "ExtBadIPC"

[00333][00334] DNSName:

[00334][00335] The class value for DNSName nodes is determined as follows:

[00335][00336] if severity = 0 then "<hostname>"

[00336][00337] else then "<hostname>_BadIP"

[00337][00338] An example structure for a New Class Event is now described.

[00338][00339] The key field for this event type looks as follows (using the PtypeConn model as an example):

[00339][00340] {

[00340][00341] "node": {

[00341][00342] "class": {

[00342][00343] "cid": "httpd"

[00343][00344] },

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00059

[00344][00345] "key": {

[00345][00346] "cid": "29654"

[00346][00347] },

[00347][00348] "type": "PtypeConn"

[00348][00349] }

[00349][00350] }

[00350][00351] It contains the class value and also the ID of the cluster where that class value is observed. Multiple clusters can be observed with the same value in a given time period. It contains the class value and also the ID of the cluster where that class value is observed. Multiple clusters can be observed with the same value in a given time period. Accordingly, in some embodiments, GBM 154 generates multiple events of this type for the same class value.

[00351][00352] The properties field looks as follows:

[00352][00353] {

[00353][00354] "set_size": 5

[00354][00355] }

[00355][00356] The set_size indicates the size of the cluster referenced in the keys field.

[00356][00357] Conditions:

[00357][00358] For a given model and time period, multiple NewClass events can be generated if there is more than one cluster in that class. NewNode events will not be generated separately in this case.

[00358][00359] Example New Class to Class Edge Event structure:

[00359][00360] The key field for this event type looks as follows (using the PtypeConn model as an example):

[00360][00361] "edge": {

[00361][00362] "dst_node": {

[00362][00363] "class": {

[00363][00364] "cid": "java war"

[00364][00365] },

[00365][00366] "key": {

[00366][00367] "cid": "27635"

[00367][00368] },

[00368][00369] "type": "PtypeConn"

[00369][00370] },

[00370][00371] "src_node": {

[00371][00372] "class": {

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00060

~~[00372]~~[00373] "cid": "IntIPC"

~~[00373]~~[00374] },

~~[00374]~~[00375] "key": {

~~[00375]~~[00376] "cid": "20881"

~~[00376]~~[00377] },

~~[00377]~~[00378] "type": "PtypeConn"

~~[00378]~~[00379] },

~~[00379]~~[00380] "type": "ConnectedTo"

~~[00380]~~[00381] }

~~[00381]~~[00382] }

~~[00382]~~[00383] The key field contains source and destination class values and also source and destination cluster identifiers (i.e., the src/dst_node:key.cid represents the src/dst cluster identifier).

~~[00383]~~[00384] In a given time period for a given model, an event of this type could involve multiple edges between different cluster pairs that have the same source and destination class values. GBM 154 can generate multiple events in this case with different source and destination cluster identifiers.

~~[00384]~~[00385] The props fields look as follows for this event type:

~~[00385]~~[00386] {

~~[00386]~~[00387] "dst_set_size": 2,

~~[00387]~~[00388] "src_set_size": 1

~~[00388]~~[00389] }

~~[00389]~~[00390] The source and destination sizes represent the sizes of the clusters given in the keys field.

~~[00390]~~[00391] Conditions:

~~[00391]~~[00392] For a given model and time period, multiple NewClassToClass events can be generated if there are more than one pair of clusters in that class pair. NewNodeToNode events are not generated separately in this case.

~~[00392]~~[00393] Combining Events at the Class Level: for a given model and time period, the following example types of events can represent multiple changes in the underlying GBM cluster level graph in terms of multiple new clusters or multiple new edges between clusters:

~~[00393]~~[00394] * NewClass

~~[00394]~~[00395] * NewEdgeClassToClass

~~[00395]~~[00396] * NewEdgeNodeToClass

~~[00396]~~[00397] * NewEdgeClassToNode

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00061

~~[00397]~~[00398] Multiple NewClass events with the same model and class can be output if there are multiple clusters in that new class.

~~[00398]~~[00399] Multiple NewEdgeClassToClass events with the same model and class pair can be output if there are multiple new cluster edges within that class pair.

~~[00399]~~[00400] Multiple NewEdgeNodeToClass events with the same model and destination class can be output if there are multiple new edges from the source cluster to the destination clusters in that destination class (the first time seeing this class as a destination cluster class for the source cluster).

~~[00400]~~[00401] Multiple NewEdgeClassToNode events with the same model and source class can be output if there are multiple new edges from source clusters to the destination clusters in that source class (the first time seeing this class as a source cluster class for the destination cluster).

~~[00401]~~[00402] These events may be combined at the class level and treated as a single event when it is desirable to view changes at the class level, e.g., when one wants to know when there is a new CType.

~~[00402]~~[00403] In some examples, different models may have partial overlap in the types of nodes they use from the base graph. Therefore, they can generate NewClass type events for the same class. NewClass events can also be combined across models when it is desirable to view changes at the class level.

~~[00403]~~[00404] Using techniques herein, actions can be associated with processes and (e.g., by associating processes with users) actions can thus also be associated with extended user sessions. Such information can be used to track user behavior correctly, even where a malicious user attempts to hide his trail by changing user identities (e.g., through lateral movement). Extended user session tracking can also be useful in operational use cases without malicious intent, e.g., where users make original logins with distinct usernames (e.g., "charlie" or "dave") but then perform actions under a common username (e.g., "admin" or "support"). One such example is where multiple users with administrator privileges exist, and they need to gain superuser privilege to perform a particular type of maintenance. It may be desirable to know which operations are performed (as the superuser) by which original user when debugging issues. In the following examples describing extended user session tracking, reference is generally made to using the secure shell (ssh) protocol as implemented by openssh (on the server side) as the mechanism for logins. However, extended user session tracking is not limited to the ssh protocol or a particular limitation and the techniques described herein can be extended to other login mechanisms.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00062

~~[00404]~~[00405] On any given machine, there will be a process that listens for and accepts ssh connections on a given port. This process can run the openssh server program running in daemon mode or it could be running another program (e.g., initd on a Linux system). In either case, a new process running openssh will be created for every new ssh login session and this process can be used to identify an ssh session on that machine. This process is called the "privileged" process in openssh.

~~[00405]~~[00406] After authentication of the ssh session, when an ssh client requests a shell or any other program to be run under that ssh session, a new process that runs that program will be created under (i.e., as a child of) the associated privileged process. If an ssh client requests port forwarding to be performed, the connections will be associated with the privileged process.

~~[00406]~~[00407] In modern operating systems such as Linux and Windows, each process has a parent process (except for the very first process) and when a new process is created the parent process is known. By tracking the parent-child hierarchy of processes, one can determine if a particular process is a descendant of a privileged openssh process and thus if it is associated with an ssh login session.

~~[00407]~~[00408] For user session tracking across machines (or on a single machine with multiple logins) in a distributed environment, it is established when two login sessions have a parent-child relationship. After that, the "original" login session, if any, for any given login session can be determined by following the parent relationship recursively.

~~[00408]~~[00409] Fig. 3F is a representation of a user logging into a first machine and then into a second machine from the first machine, as well as information associated with such actions. In the example of Fig. 3F, a user, Charlie, logs into Machine A (331) from a first IP address (332). As part of the login process, he provides a username (333). Once connected to Machine A, an openssh privileged process (334) is created to handle the connection for the user, and a terminal session is created and a bash process (335) is created as a child. Charlie launches an ssh client (336) from the shell, and uses it to connect (337) to Machine B (338). As with the connection he makes to Machine A, Charlie's connection to Machine B will have an associated incoming IP address (339), in this case, the IP address of Machine A. And, as part of the login process with Machine B, Charlie will provide a username (340) which need not be the same as username 333. An openssh privileged process (341) is created to handle the connection, and a terminal session and child bash process (342) will be created. From the command line of Machine B, Charlie launches a curl command (343), which opens an HTTP connection (2028) to an external Machine C (345).

~~[00409]~~[00410] Fig. 3G is an alternate representation of actions occurring in Fig. 3F, where events occurring on Machine A are indicated along line 350, and events occurring on Machine B

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00063

are indicated along line 351. As shown in Fig. 3G, an incoming ssh connection is received at Machine A (352). Charlie logs in (as user "x") and an ssh privileged process is created to handle Charlie's connection (353). A terminal session is created and a bash process is created (354) as a child of process 353. Charlie wants to ssh to Machine B, and so executes an ssh client on Machine A (355), providing credentials (as user "y") at 356. Charlie logs into Machine B, and an ssh privileged process is created to handle Charlie's connection (357). A terminal session is created and a bash process is created (358) as a child of process 357. Charlie then executes curl (359) to download content from an external domain (via connection 360).

~~[00410]~~[00411] The external domain could be a malicious domain, or it could be benign. Suppose the external domain is malicious (and, e.g., Charlie has malicious intent). It would be advantageous (e.g., for security reasons) to be able to trace the contact with the external domain back to Machine A, and then back to Charlie's IP address. Using techniques described herein (e.g., by correlating process information collected by various agents), such tracking of Charlie's activities back to his original login (330) can be accomplished. In particular, an extended user session can be tracked that associates Charlie's ssh processes together with a single original login and thus original user.

~~[00411]~~[00412] As described herein, software agents (such as agent 112) may run on machines (such as a machine that implements one of nodes 116) and detect new connections, processes, and/or logins. As also previously explained, such agents send associated records to data platform 12 which includes one or more datastores (e.g., data store 30) for persistently storing such data. Such data can be modeled using logical tables, also persisted in datastores (e.g., in a relational database that provides an SQL interface), allowing for querying of the data. Other datastores such as graph oriented databases and/or hybrid schemes can also be used.

~~[00412]~~[00413] The following identifiers are commonly used in the tables:

~~[00413]~~[00414] * MID

~~[00414]~~[00415] * PID_hash

~~[00415]~~[00416] An ssh login session can be identified uniquely by an (MID, PID_hash) tuple. The MID is a machine identifier that is unique to each machine, whether physical or virtual, across time and space. Operating systems use numbers called process identifiers (PIDs) to identify processes running at a given time. Over time processes may die and new processes may be started on a machine or the machine itself may restart. The PID is not necessarily unique across time in that the same PID value can be reused for different processes at different times. In order to track process descendants across time, one should therefore account for time as well. In order to be able to identify a process on a machine uniquely across time, another number called a PID_hash is generated for the process. In various embodiments, the PID_hash is generated using

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00064

a collision-resistant hash function that takes the PID, start time, and (in various embodiments, as applicable) other properties of a process.

~~[00416]~~[00417] Input data collected by agents comprises the input data model and is represented by the following logical tables:

~~[00417]~~[00418] * connections

~~[00418]~~[00419] * processes

~~[00419]~~[00420] * logins

~~[00420]~~[00421] A connections table may maintain records of TCP/IP connections observed on each machine. Example columns included in a connections table are as follows:

Column Name	Description
MID	Identifier of the machine that the connection was observed on.
start_time	Connection start time.
PID_hash	Identifier of the process that was associated with the connection.
src_IP_addr	Source IP address (the connection was initiated from this IP address).
src_port	Source port.
dst_IP_addr	Destination IP address (the connection was made to this IP address).
dst_port	Destination port.
Prot	Protocol (TCP or UDP).
Dir	Direction of the connection (incoming or outgoing) with respect to this machine.

~~[00421]~~[00422] The source fields (IP address and port) correspond to the side from which the connection was initiated. On the destination side, the agent associates an ssh connection with the privileged ssh process that is created for that connection.

~~[00422]~~[00423] For each connection in the system, there will be two records in the table, assuming that the machines on both sides of the connection capture the connection. These records can be matched based on equality of the tuple (src_IP_addr, src_port, dst_IP_addr, dst_port, Prot) and proximity of the start_time fields (e.g., with a one minute upper threshold between the start_time fields).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00065

~~[00423]~~[00424] A processes table maintains records of processes observed on each machine. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the process was observed on.
PID_hash	Identifier of the process.
start_time	Start time of the process.
exe_path	The executable path of the process.
PPID_hash	Identifier of the parent process.

~~[00424]~~[00425] A logins table may maintain records of logins to machines. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with login.
login_time	Time of login.
login_username	Username used in login.

~~[00425]~~[00426] Output data generated by session tracking is represented with the following logical tables:

~~[00426]~~[00427] * login-local-descendant

~~[00427]~~[00428] * login-connection

~~[00428]~~[00429] * login-lineage

~~[00429]~~[00430] Using data in these tables, it is possible to determine descendant processes of a given ssh login session across the environment (i.e., spanning machines). Conversely, given a process, it is possible to determine if it is an ssh login descendant as well as the original ssh login session for it if so.

~~[00430]~~[00431] A login-local-descendant table maintains the local (i.e., on the same machine) descendant processes of each ssh login session. It may have the following columns:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00066

Column Name	Description
MID	Identifier of the machine that the login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with login.
login_time	Time of login.
login_username	Username used in login.

~~[00431]~~[00432] A login-connections table may maintain the connections associated with ssh logins. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the process was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with the login.
login_time	Time of login.
login_username	The username used in the login.
src_IP_addr	Source IP address (connection was initiated from this IP address).
src_port	Source port.
dst_IP_addr	Destination IP address (connection was made to this IP address).
dst_port	Destination port.

~~[00432]~~[00433] A login-lineage table may maintain the lineage of ssh login sessions. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the ssh login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with the login.
parent_MID	Identifier of the machine that the parent ssh login was observed on.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00067

parent_sshd_PID_hash	Identifier of the sshd privileged process associated with the parent login.
origin_MID	Identifier of the machine that the origin ssh login was observed on.
origin_sshd_PID_hash	Identifier of the sshd privileged process associated with the origin login.

~~[00433]~~[00434] The parent_MID and parent_sshd_PID_hash columns can be null if there is no parent ssh login. In that case, the (MID, sshd_PID_hash) tuple will be the same as the (origin_MID, origin_sshd_PID_hash) tuple.

~~[00434]~~[00435] Fig. 3H illustrates an example of a process for performing extended user tracking. In various embodiments, process 361 is performed by data platform 12. The process begins at 362 when data associated with activities occurring in a network environment (such as entity A's datacenter) is received. One example of such data that can be received at 362 is agent-collected data described above (e.g., in conjunction with process 200). At 363, the received network activity is used to identify user login activity. And, at 364, a logical graph that links the user login activity to at least one user and at least one process is generated (or updated, as applicable). Additional detail regarding process 361, and in particular, portions 363 and 364 of process 361 are described in more detail below (e.g., in conjunction with discussion of Fig. 3J).

~~[00435]~~[00436] Fig. 3I depicts a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection. The scenario depicted in Fig. 3I is used to describe an example of processing that can be performed on data collected by agents to generate extended user session tracking information. Fig. 3I is an alternate depiction of the information shown in Figs. 3F and 3G.

~~[00436]~~[00437] At time t1 (365), a first ssh connection is made to Machine A (366) from an external source (367) by a user having a username of "X." In the following example, suppose the external source has an IP address of 1.1.1.10 and uses source port 10000 to connect to Machine A (which has an IP address of 2.2.2.20 and a destination port 22). External source 367 is considered an external source because its IP address is outside of the environment being monitored (e.g., is a node outside of entity A's datacenter, connecting to a node inside of entity A's datacenter).

~~[00437]~~[00438] A first ssh login session LS1 is created on machine A for user X. The privileged openssh process for this login is A1 (368). Under the login session LS1, the user creates a bash shell process with PID_hash A2 (369).

~~[00438]~~[00439] At time t2 (370), inside the bash shell process A2, the user runs an ssh program under a new process A3 (371) to log in to machine B (372) with a different username ("Y"). In

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00068

particular, an ssh connection is made from source IP address 2.2.2.20 and source port 10001 (Machine A's source information) to destination IP address 2.2.2.21 and destination port 22 (Machine B's destination information).

~~[00439]~~[00440] A second ssh login session LS2 is created on machine B for user Y. The privileged openssh process for this login is B1 (373). Under the login session LS2, the user creates a bash shell process with PID_hash B2 (374).

~~[00440]~~[00441] At time t3 (376), inside the bash shell process B2, the user runs a curl command under a new process B3 (377) to download a file from an external destination (378). In particular, an HTTPS connection is made from source IP address 2.2.2.21 and source port 10002 (Machine B's source information) to external destination IP address 3.3.3.30 and destination port 443 (the external destination's information).

~~[00441]~~[00442] Using techniques described herein, it is possible to determine the original user who initiated the connection to external destination 378, which in this example is a user having the username X on machine A (where the extended user session can be determined to start with ssh login session LS1).

~~[00442]~~[00443] Based on local descendant tracking, the following determinations can be on machine A and B without yet having performed additional processing (described in more detail below):

~~[00443]~~[00444] * A3 is a descendant of A1 and thus associated with LS1.

~~[00444]~~[00445] * The connection to the external domain from machine B is initiated by B3.

~~[00445]~~[00446] * B3 is a descendant of B1 and is thus associated with LS2.

~~[00446]~~[00447] * Connection to the external domain is thus associated with LS2.

~~[00447]~~[00448] An association between A3 and LS2 can be established based on the fact that LS2 was created based on an ssh connection initiated from A3. Accordingly, it can be determined that LS2 is a child of LS1.

~~[00448]~~[00449] To determine the user responsible for making the connection to the external destination (e.g., if it were a known bad destination), first, the process that made the connection would be traced, i.e., from B3 to LS2. Then LS2 would be traced to LS1 (i.e., LS1 is the origin login session for LS2). Thus the user for this connection is the user for LS1, i.e., X. As represented in Fig. 3I, one can visualize the tracing by following the links (in the reverse direction of arrows) from external destination 378 to A1 (368).

~~[00449]~~[00450] In the example scenario, it is assumed that both ssh connections occur in the same analysis period. However, the approaches described herein will also work for connections and processes that are created in different time periods.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00069

~~[00450]~~[00451] Fig. 3J illustrates an example of a process for performing extended user tracking. In various embodiments, process 380 is performed periodically (e.g., once an hour in a batch fashion) by ssh tracker 148 to generate new output data. In general, batch processing allows for efficient analysis of large volumes of data. However, the approach can be adapted, as applicable, to process input data on a record-by-record fashion while maintaining the same logical data processing flow. As applicable the results of a given portion of process 380 are stored for use in a subsequent portion.

~~[00451]~~[00452] The process begins at 381 when new ssh connection records are identified. In particular, new ssh connections started during the current time period are identified by querying the connections table. The query uses filters on the start_time and dst_port columns. The values of the range filter on the start_time column are based on the current time period. The dst_port column is checked against ssh listening port(s). By default, the ssh listening port number is 22. However, as this could vary across environments, the port(s) that openssh servers are listening to in the environment can be determined by data collection agents dynamically and used as the filter value for the dst_port as applicable. In the scenario depicted in Fig. 3I, the query result will generate the records shown in Fig. 3K. Note that for the connection between machine A and B, the two machines are likely to report start_time values that are not exactly the same but close enough to be considered matching (e.g., within one minute or another appropriate amount of time). In the above table, they are shown to be the same for simplicity.

~~[00452]~~[00453] At 382, ssh connection records reported from source and destination sides of the same connection are matched. The ssh connection records (e.g., returned from the query at 381) are matched based on the following criteria:

~~[00453]~~[00454] * The five tuples (src_IP, dst_IP, IP_prot, src_port, dst_port) of the connection records must match.

~~[00454]~~[00455] * The delta between the start times of the connections must be within a limit that would account for the worst case clock difference expected between two machines in the environment and typical connection setup latency.

~~[00455]~~[00456] * If there are multiple matches possible, then the match with the smallest time delta is chosen.

~~[00456]~~[00457] Note that record 390 from machine A for the incoming connection from the external source cannot be matched with another record as there is an agent only on the destination side for this connection. Example output of portion 382 of process 380 is shown in Fig. 3L. The values in the dst_PID_hash column (391) are that of the sshd privileged process associated with ssh logins.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00070

~~[00457]~~[00458] At 383, new logins during the current time period are identified by querying the logins table. The query uses a range filter on the login_time column with values based on the current time period. In the example depicted in Fig. 3I, the query result will generate the records depicted in Fig. 3M.

~~[00458]~~[00459] At 384, matched ssh connection records created at 382 and new login records created at 383 are joined to create new records that will eventually be stored in the login-connection table. The join condition is that dst_MID of the matched connection record is equal to the MID of the login record and the dst_PID_hash of the matched connection record is equal to the sshd_PID_hash of the login record. In the example depicted in Fig. 3I, the processing performed at 384 will generate the records depicted in Fig. 3N.

~~[00459]~~[00460] At 385, login-local-descendant records in the lookback time period are identified. It is possible that a process that is created in a previous time period makes an ssh connection in the current analysis batch period. Although not depicted in the example illustrated in Fig. 3I, consider a case where bash process A2 does not create ssh process A3 right away but instead that the ssh connection A3 later makes to machine B is processed in a subsequent time period than the one where A2 was processed. While processing this subsequent time period in which processes A3 and B1 are seen, knowledge of A2 would be useful in establishing that B1 is associated with A3 (via ssh connection) which is associated with A2 (via process parentage) which in turn would be useful in establishing that the parent of the second ssh login is the first ssh login. The time period for which look back is performed can be limited to reduce the amount of historical data that is considered. However, this is not a requirement (and the amount of look back can be determined, e.g., based on available processing resources). The login local descendants in the lookback time period can be identified by querying the login-local-descendant table. The query uses a range filter on the login_time column where the range is from start_time_of_current_period - lookback_time to start_time_of_current_period. (No records as a result of performing 385 on the scenario depicted in Fig. 3I are obtained, as only a single time period is applicable in the example scenario.)

~~[00460]~~[00461] At 386, new processes that are started in the current time period are identified by querying the processes table. The query uses a range filter on the start_time column with values based on the current time period. In the example depicted in Fig. 3I, the processing performed at 386 will generate the records depicted in Fig. 3O.

~~[00461]~~[00462] At 387, new login-local-descendant records are identified. The purpose is to determine whether any of the new processes in the current time period are descendants of an ssh login process and if so to create records that will be stored in the login-local-descendant table for them. In order to do so, the parent-child relationships between the processes are recursively

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00071

followed. Either a top down or bottom up approach can be used. In a top down approach, the ssh local descendants in the lookback period identified at 385, along with new ssh login processes in the current period identified at 384 are considered as possible ancestors for the new processes in the current period identified at 386.

~~[00462]~~[00463] Conceptually, the recursive approach can be considered to include multiple sub-steps where new processes that are identified to be ssh local descendants in the current sub-step are considered as ancestors for the next step. In the example scenario depicted in Fig. 3I, the following descendancy relationships will be established in two sub-steps:

~~[00463]~~[00464] Sub-step 1:

~~[00464]~~[00465] Process A2 is a local descendant of LS1 (i.e., MID = A, sshd_PID_hash = A1) because it is a child of process A1 which is the login process for LS1.

~~[00465]~~[00466] Process B2 is a local descendant of LS2 (i.e., MID = B, sshd_PID_hash = B1) because it is a child of process B1 which is the login process for LS2.

~~[00466]~~[00467] Sub-step 2:

~~[00467]~~[00468] Process A3 is a local descendant of LS1 because it is a child of process A2 which is associated to LS1 in sub-step 1.

~~[00468]~~[00469] Process B3 is a local descendant of LS2 because it is a child of process B1 which is associated to LS2 in sub-step 1.

~~[00469]~~[00470] Implementation portion 387 can use a datastore that supports recursive query capabilities, or, queries can be constructed to process multiple conceptual sub-steps at once. In the example depicted in Fig. 3I, the processing performed at 387 will generate the records depicted in Fig. 3P. Note that the ssh privileged processes associated with the logins are also included as they are part of the login session.

~~[00470]~~[00471] At 388, the lineage of new ssh logins created in the current time period is determined by associating their ssh connections to source processes that may be descendants of other ssh logins (which may have been created in the current period or previous time periods). In order to do so, first an attempt is made to join the new ssh login connections in the current period (identified at 384) with the combination of the login local descendants in the lookback period (identified at 385) and the login local descendants in the current time period (identified at 386). This will create adjacency relationships between child and parent logins. In the example depicted in Fig. 3I, the second ssh login connection will be associated with process A3 and an adjacency relationship between the two login sessions will be created (as illustrated in Fig. 3Q).

~~[00471]~~[00472] Next, the adjacency relationships are used to find the original login sessions. While not shown in the sample scenario, there could be multiple ssh logins in a chain in the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00072

current time period, in which case a recursive approach (as in 387) could be used. At the conclusion of portion 388, the login lineage records depicted in Fig. 3R will be generated.

~~[00472]~~[00473] Finally, at 389, output data is generated. In particular, the new login-connection, login-local-descendant, and login-lineage records generated at 384, 387, and 388 are inserted into their respective output tables (e.g., in a transaction manner).

~~[00473]~~[00474] An alternate approach to matching TCP connections between machines running an agent is for the client to generate a connection GUID and send it in the connection request (e.g., the SYN packet) it sends and for the server to extract the GUID from the request. If two connection records from two machines have the same GUID, they are for the same connection. Both the client and server will store the GUID (if it exists) in the connection records they maintain and report. On the client side, the agent can configure the network stack (e.g. using IP tables functionality on Linux) to intercept an outgoing TCP SYN packet and modify it to add the generated GUID as a TCP option. On the server side, the agent already extracts TCP SYN packets and thus can look for this option and extract the GUID if it exists.

~~[00474]~~[00475] Example graph-based user tracking and threat detection embodiments associated with data platform 12 will now be described. Administrators and other users of network environments (e.g., entity A's datacenter 104) often change roles to perform tasks. As one example, suppose that at the start of a workday, an administrator (hereinafter "Joe Smith") logs in to a console, using an individualized account (e.g., username=joe.smith). Joe performs various tasks as himself (e.g., answering emails, generating status reports, writing code, etc.). For other tasks (e.g., performing updates), Joe may require different/additional permission than his individual account has (e.g., root privileges). One way Joe can gain access to such permissions is by using sudo, which will allow Joe to run a single command with root privileges. Another way Joe can gain access to such permissions is by su or otherwise logging into a shell as root. After gaining root privileges, another thing that Joe can do is switch identities. As one example, to perform administrative tasks, Joe may use "su help" or "su database-admin" to become (respectively) the help user or the database-admin user on a system. He may also connect from one machine to another, potentially changing identities along the way (e.g., logging in as joe.smith at a first console, and connecting to a database server as database-admin). When he's completed various administrative tasks, Joe can relinquish his root privileges by closing out of any additional shells created, reverting back to a shell created for user joe.smith.

~~[00475]~~[00476] While there are many legitimate reasons for Joe to change his identity throughout the day, such changes may also correspond to nefarious activity. Joe himself may be nefarious, or Joe's account (joe.smith) may have been compromised by a third party (whether an "outsider" outside of entity A's network, or an "insider"). Using techniques described herein, the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00073

behavior of users of the environment can be tracked (including across multiple accounts and/or multiple machines) and modeled (e.g., using various graphs described herein). Such models can be used to generate alerts (e.g., to anomalous user behavior). Such models can also be used forensically, e.g., helping an investigator visualize various aspects of a network and activities that have occurred, and to attribute particular types of actions (e.g., network connections or file accesses) to specific users.

~~[00476]~~[00477] In a typical day in a datacenter, a user (e.g., Joe Smith) will log in, run various processes, and (optionally) log out. The user will typically log in from the same set of IP addresses, from IP addresses within the same geographical area (e.g., city or country), or from historically known IP addresses/geographical areas (i.e., ones the user has previously/occasionally used). A deviation from the user's typical (or historical) behavior indicates a change in login behavior. However, it does not necessarily mean that a breach has occurred. Once logged into a datacenter, a user may take a variety of actions. As a first example, a user might execute a binary/script. Such binary/script might communicate with other nodes in the datacenter, or outside of the datacenter, and transfer data to the user (e.g., executing "curl" to obtain data from a service external to the datacenter). As a second example, the user can similarly transfer data (e.g., out of the datacenter), such as by using POST. As a third example, a user might change privilege (one or more times), at which point the user can send/receive data as per above. As a fourth example, a user might connect to a different machine within the datacenter (one or more times), at which point the user can send/receive data as per the above.

~~[00477]~~[00478] In various embodiments, the above information associated with user behavior is broken into four tiers. The tiers represent example types of information that data platform 12 can use in modeling user behavior:

~~[00478]~~[00479] 1. The user's entry point (e.g., domains, IP addresses, and/or geolocation information such as country/city) from which a user logs in.

~~[00479]~~[00480] 2. The login user and machine class.

~~[00480]~~[00481] 3. Binaries, executables, processes, etc. a user launches.

~~[00481]~~[00482] 4. Internal servers with which the user (or any of the user's processes, child processes, etc.) communicates, and external contacts (e.g., domains, IP addresses, and/or geolocation information such as country/city) with which the user communicates (i.e., transfers data).

~~[00482]~~[00483] In the event of a security breach, being able to concretely answer questions about such information can be very important. And, collectively, such information is useful in providing an end-to-end path (e.g., for performing investigations).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00074

~~[00483]~~[00484] In the following example, suppose a user ("UserA") logs into a machine ("Machine01") from a first IP address ("IP01"). Machine01 is inside a datacenter. UserA then launches a script ("runnable.sh") on Machine01. From Machine01, UserA next logs into a second machine ("Machine02") via ssh, also as UserA, also within the datacenter. On Machine02, UserA again launches a script ("new_runnable.sh"). On Machine02, UserA then changes privilege, becoming root on Machine02. From Machine02, UserA (now as root) logs into a third machine ("Machine03") in the datacenter via ssh, as root on Machine03. As root on Machine03, the user executes a script ("collect_data.sh") on Machine03. The script internally communicates (as root) to a MySQL-based service internal to the datacenter, and downloads data from the MySQL-based service. Finally, as root on Machine03, the user externally communicates with a server outside the datacenter ("External01"), using a POST command. To summarize what has occurred, in this example, the source/entry point is IP01. Data is transferred to an external server External01. The machine performing the transfer to External01 is Machine03. The user transferring the data is "root" (on Machine03), while the actual user (hiding behind root) is UserA.

~~[00484]~~[00485] In the above scenario, the "original user" (ultimately responsible for transmitting data to External01) is UserA, who logged in from IP01. Each of the processes ultimately started by UserA, whether started at the command line (tty) such as "runnable.sh" or started after an ssh connection such as "new_runnable.sh," and whether as UserA, or as a subsequent identity, are all examples of child processes which can be arranged into a process hierarchy.

~~[00485]~~[00486] As previously mentioned, machines can be clustered together logically into machine clusters. One approach to clustering is to classify machines based on information such as the types of services they provide/binaries they have installed upon them/processes they execute. Machines sharing a given machine class (as they share common binaries/services/etc.) will behave similarly to one another. Each machine in a datacenter can be assigned to a machine cluster, and each machine cluster can be assigned an identifier (also referred to herein as a machine class). One or more tags can also be assigned to a given machine class (e.g., database_servers_west or prod_web_frontend). One approach to assigning a tag to a machine class is to apply term frequency analysis (e.g., TF/IDF) to the applications run by a given machine class, selecting as tags those most unique to the class. Data platform 12 can use behavioral baselines taken for a class of machines to identify deviations from the baseline (e.g., by a particular machine in the class).

~~[00486]~~[00487] Fig. 3S illustrates an example of a process for detecting anomalies. In various embodiments, process 392 is performed by data platform 12. As explained above, a given session will have an original user. And, each action taken by the original user can be tied back to the original user, despite privilege changes and/or lateral movement throughout a datacenter. Process

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00075

392 begins at 393 when log data associated with a user session (and thus an original user) is received. At 394, a logical graph is generated, using at least a portion of the collected data. When an anomaly is detected (395), it can be recorded, and as applicable, an alert is generated (396). The following are examples of graphs that can be generated (e.g., at 394), with corresponding examples of anomalies that can be detected (e.g., at 395) and alerted upon (e.g., at 396).

~~[00487]~~[00488] Fig. 4A illustrates a representation of an embodiment of an insider behavior graph. In the example of Fig. 4A, each node in the graph can be: (1) a cluster of users; (2) a cluster of launched processes; (3) a cluster of processes/servers running on a machine class; (4) a cluster of external IP addresses (of incoming clients); or (5) a cluster of external servers based on DNS/IP/etc. As depicted in Fig. 4A, graph data is vertically tiered into four tiers. Tier 0 (400) corresponds to entry point information (e.g., domains, IP addresses, and/or geolocation information) associated with a client entering the datacenter from an external entry point. Entry points are clustered together based on such information. Tier 1 (401) corresponds to a user on a machine class, with a given user on a given machine class represented as a node. Tier 2 (402) corresponds to launched processes, child processes, and/or interactive processes. Processes for a given user and having similar connectivity (e.g., sharing the processes they launch and the machines with which they communicate) are grouped into nodes. Finally, Tier 3 (403) corresponds to the services/servers/domains/IP addresses with which processes communicate. A relationship between the tiers can be stated as follows: Tier 0 nodes log in to tier 1 nodes. Tier 1 nodes launch tier 2 nodes. Tier 2 nodes connect to tier 3 nodes.

~~[00488]~~[00489] The inclusion of an original user in both Tier 1 and Tier 2 allows for horizontal tiering. Such horizontal tiering ensures that there is no overlap between any two users in Tier 1 and Tier 2. Such lack of overlap provides for faster searching of an end-to-end path (e.g., one starting with a Tier 0 node and terminating at a Tier 3 node). Horizontal tiering also helps in establishing baseline insider behavior. For example, by building an hourly insider behavior graph, new edges/changes in edges between nodes in Tier 1 and Tier 2 can be identified. Any such changes correspond to a change associated with the original user. And, any such changes can be surfaced as anomalous and alerts can be generated.

~~[00489]~~[00490] As explained above, Tier 1 corresponds to a user (e.g., user "U") logging into a machine having a particular machine class (e.g., machine class "M"). Tier 2 is a cluster of processes having command line similarity (e.g., CType "C"), having an original user "U," and running as a particular effective user (e.g., user "U1"). The value of U1 may be the same as U (e.g., joe.smith in both cases), or the value of U1 may be different (e.g., U=joe.smith and U1=root). Thus, while an edge may be present from a Tier 1 node to a Tier 2 node, the effective

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00076

user in the Tier 2 node may or may not match the original user (while the original user in the Tier 2 node will match the original user in the Tier 1 node).

~~[00490]~~[00491] A change from a user U into a user U1 can take place in a variety of ways. Examples include where U becomes U1 on the same machine (e.g., via su), and also where U sshes to other machine(s). In both situations, U can perform multiple changes, and can combine approaches. For example, U can become U1 on a first machine, ssh to a second machine (as U1), become U2 on the second machine, and ssh to a third machine (whether as user U2 or user U3). In various embodiments, the complexity of how user U ultimately becomes U3 (or U5, etc.) is hidden from a viewer of an insider behavior graph, and only an original user (e.g., U) and the effective user of a given node (e.g., U5) are depicted. As applicable (e.g., if desired by a viewer of the insider behavior graph), additional detail about the path (e.g., an end-to-end path of edges from user U to user U5) can be surfaced (e.g., via user interactions with nodes).

~~[00491]~~[00492] Fig. 4B illustrates an example of a portion of an insider behavior graph (e.g., as rendered in a web browser). In the example shown, node 405 (the external IP address, 52.32.40.231) is an example of a Tier 0 node, and represents an entry point into a datacenter. As indicated by directional arrows 406 and 407, two users, "aruneli_prod" and "harish_prod," both made use of the source IP 52.32.40.231 when logging in between 5pm and 6pm on Sunday July 30 (408). Nodes 409 and 410 are examples of Tier 1 nodes, having aruneli_prod and harish_prod as associated respective original users. As previously mentioned, Tier 1 nodes correspond to a combination of a user and a machine class. In the example depicted in Fig. 4B, the machine class associated with nodes 409 and 410 is hidden from view to simplify visualization, but can be surfaced to a viewer of interface 404 (e.g., when the user clicks on node 409 or 410).

~~[00492]~~[00493] Nodes 414-423 are examples of Tier 2 nodes -- processes that are launched by users in Tier 1 and their child, grandchild, etc. processes. Note that also depicted in Fig. 4B is a Tier 1 node 411 that corresponds to a user, "root," that logged in to a machine cluster from within the datacenter (i.e., has an entry point within the datacenter). Nodes 425-1 and 425-2 are examples of Tier 3 nodes -- internal/external IP addresses, servers, etc., with which Tier 2 nodes communicate.

~~[00493]~~[00494] In the example shown in Fig. 4B, a viewer of interface 404 has clicked on node 423. As indicated in region 426, the user running the marathon container is "root." However, by following the directional arrows in the graph backwards from node 423 (i.e. from right to left), the viewer can determine that the original user, responsible for node 423, is "aruneli_prod," who logged into the datacenter from IP 52.32.40.231.

~~[00494]~~[00495] The following are examples of changes that can be tracked using an insider behavior graph model:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00077

~~[00495]~~[00496] * A user logs in from a new IP address.

~~[00496]~~[00497] * A user logs in from a geolocation not previously used by that user.

~~[00497]~~[00498] * A user logs into a new machine class.

~~[00498]~~[00499] * A user launches a process not previously used by that user.

~~[00499]~~[00500] * A user connects to an internal server to which the user has not previously connected.

~~[00500]~~[00501] * An original user communicates with an external server (or external server known to be malicious) with which that user has not previously communicated.

~~[00501]~~[00502] * A user communicates with an external server which has a geolocation not previously used by that user.

~~[00502]~~[00503] Such changes can be surfaced as alerts, e.g., to help an administrator determine when/what anomalous behavior occurs within a datacenter. Further, the behavior graph model can be used (e.g., during forensic analysis) to answer questions helpful during an investigation. Examples of such questions include:

~~[00503]~~[00504] * Was there any new login activity (Tier 0) in the timeframe being investigated? As one example, has a user logged in from an IP address with unknown geolocation information? Similarly, has a user started communicating externally with a new Tier 3 node (e.g., one with unknown geolocation information).

~~[00504]~~[00505] * Has there been any suspicious login activity (Tier 0) in the timeframe being investigated? As one example, has a user logged in from an IP address that corresponds to a known bad IP address as maintained by Threat aggregator 150? Similarly, has there been any suspicious Tier 3 activity?

~~[00505]~~[00506] * Were any anomalous connections made within the datacenter during the timeframe being investigated? As one example, suppose a given user ("Frank") typically enters a datacenter from a particular IP address (or range of IP addresses), and then connects to a first machine type (e.g., bastion), and then to a second machine type (e.g., database_prod). If Frank has directly connected to database_prod (instead of first going through bastion) during the timeframe, this can be surfaced using the insider graph.

~~[00506]~~[00507] * Who is (the original user) responsible for running a particular process?

~~[00507]~~[00508] An example of an insider behavior graph being used in an investigation is depicted in Figs. 4C and 4D. Fig. 4C depicts a baseline of behavior for a user, "Bill." As shown in Fig. 4C, Bill typically logs into a datacenter from the IP address, 71.198.44.40 (427). He typically makes use of ssh (428), and sudo (429), makes use of a set of typical applications (430) and connects (as root) with the external service, api.lacework.net (431).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00078

~~[00508]~~[00509] Suppose Bill's credentials are compromised by a nefarious outsider ("Eve"). Fig. 4D depicts an embodiment of how the graph depicted in Fig. 4C would appear once Eve begins exfiltrating data from the datacenter. Eve logs into the datacenter (using Bill's credentials) from 52.5.66.8 (432). As Bill, Eve escalates her privilege to root (e.g., via su), and then becomes a different user, Alex (e.g., via su alex). As Alex, Eve executes a script, "sneak.sh" (433), which launches another script, "post.sh" (434), which contacts external server 435 which has an IP address of 52.5.66.7, and transmits data to it. Edges 436-439 each represent changes in Bill's behavior. As previously mentioned, such changes can be detected as anomalies and associated alerts can be generated. As a first example, Bill logging in from an IP address he has not previously logged in from (436) can generate an alert. As a second example, while Bill does typically make use of sudo (429), he has not previously executed sneak.sh (433) or post.sh (434) and the execution of those scripts can generate alerts as well. As a third example, Bill has not previously communicated with server 435, and an alert can be generated when he does so (439). Considered individually, each of edges 436-439 may indicate nefarious behavior, or may be benign. As an example of a benign edge, suppose Bill begins working from a home office two days a week. The first time he logs in from his home office (i.e., from an IP address that is not 71.198.44.40), an alert can be generated that he has logged in from a new location. Over time, however, as Bill continues to log in from his home office but otherwise engages in typical activities, Bill's graph will evolve to include logins from both 71.198.44.40 and his home office as baseline behavior. Similarly, if Bill begins using a new tool in his job, an alert can be generated the first time he executes the tool, but over time will become part of his baseline.

~~[00509]~~[00510] In some cases, a single edge can indicate a serious threat. For example, if server 432 (or 435) is included in a known bad IP listing, edge 436 (or 439) indicates compromise. An alert that includes an appropriate severity level (e.g., "threat level high") can be generated. In other cases, a combination of edges could indicate a threat (where a single edge might otherwise result in a lesser warning). In the example shown in Fig. 4D, the presence of multiple new edges is indicative of a serious threat. Of note, even though "sneak.sh" and "post.sh" were executed by Alex, because data platform 12 also keeps track of an original user, the compromise of user B's account will be discovered.

~~[00510]~~[00511] Fig. 4E illustrates a representation of an embodiment of a user login graph. In the example of Fig. 4E, tier 0 (440) clusters source IP addresses as belonging to a particular country (including an "unknown" country) or as a known bad IP. Tier 1 (441) clusters user logins, and tier 2 (442) clusters type of machine class into which a user is logging in. The user login graph tracks the typical login behavior of users. By interacting with a representation of the graph, answers to questions such as the following can be obtained:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00079

~~[00511]~~[00512] * Where is a user logging in from?

~~[00512]~~[00513] * Have any users logged in from a known bad address?

~~[00513]~~[00514] * Have any non-developer users accessed development machines?

~~[00514]~~[00515] * Which machines does a particular user access?

~~[00515]~~[00516] Examples of alerts that can be generated using the user login graph include:

~~[00516]~~[00517] * A user logs in from a known bad IP address.

~~[00517]~~[00518] * A user logs in from a new country for the first time.

~~[00518]~~[00519] * A new user logs into the datacenter for the first time.

~~[00519]~~[00520] * A user accesses a machine class that the user has not previously accessed.

~~[00520]~~[00521] One way to track privilege changes in a datacenter is by monitoring a process hierarchy of processes. To help filter out noisy commands/processes such as "su –u," the hierarchy of processes can be constrained to those associated with network activity. In a *nix system, each process has two identifiers assigned to it, a process identifier (PID) and a parent process identifier (PPID). When such a system starts, the initial process is assigned a PID 0. Each user process has a corresponding parent process.

~~[00521]~~[00522] Using techniques described herein, a graph can be constructed (also referred to herein as a privilege change graph) which models privilege changes. In particular, a graph can be constructed which identifies where a process P1 launches a process P2, where P1 and P2 each have an associated user U1 and U2, with U1 being an original user, and U2 being an effective user. In the graph, each node is a cluster of processes (sharing a CType) executed by a particular (original) user. As all the processes in the cluster belong to the same user, a label that can be used for the cluster is the user's username. An edge in the graph, from a first node to a second node, indicates that a user of the first node changed its privilege to the user of the second node.

~~[00522]~~[00523] Fig. 4F illustrates an example of a privilege change graph. In the example shown in Fig. 4F, each node (e.g., nodes 444 and 445) represents a user. Privilege changes are indicated by edges, such as edge 446.

~~[00523]~~[00524] As with other graphs, anomalies in graph 443 can be used to generate alerts. Three examples of such alerts are as follows:

~~[00524]~~[00525] * New user entering the datacenter. Any time a new user enters the datacenter and runs a process, the graph will show a new node, with a new CType. This indicates a new user has been detected within the datacenter. Fig. 4F is a representation of an example of an interface that depicts such an alert. Specifically, as indicated in region 447, an alert for the time period 1pm-2pm on June 8 was generated. The alert identifies that a new user, Bill (448) executed a process.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00080

~~[00525]~~[00526] *Privilege change. As explained above, a new edge, from a first node (user A) to a second node (user B) indicates that user A has changed privilege to user B.

~~[00526]~~[00527] * Privilege escalation. Privilege escalation is a particular case of privilege change, in which the first user becomes root.

~~[00527]~~[00528] An example of an anomalous privilege change and an example of an anomalous privilege escalation are each depicted in graph 450 of Fig. 4G. In particular, as indicated in region 451, two alerts for the time period 2pm-3pm on June 8 were generated (corresponding to the detection of the two anomalous events). In region 452, root has changed privilege to the user "daemon," which root has not previously done. This anomaly is indicated to the user by highlighting the daemon node (e.g., outlining it in a particular color, e.g., red). As indicated by edge 453, Bill has escalated his privilege to the user root (which can similarly be highlighted in region 454). This action by Bill represents a privilege escalation.

~~[00528]~~[00529] An Extensible query interface for dynamic data compositions and filter applications will now be described.

~~[00529]~~[00530] As described herein, datacenters are highly dynamic environments. And, different customers of data platform 12 (e.g., entity A vs. entity B) may have different/disparate needs/requirements of data platform 12, e.g., due to having different types of assets, different applications, etc. Further, as time progresses, new software tools will be developed, new types of anomalous behavior will be possible (and should be detectable), etc. In various embodiments, data platform 12 makes use of predefined relational schema (including by having different predefined relational schema for different customers). However, the complexity and cost of maintaining/updating such predefined relational schema can rapidly become problematic – particularly where the schema includes a mix of relational, nested, and hierarchical (graph) datasets. In other embodiments, the data models and filtering applications used by data platform 12 are extensible. As will be described in more detail below, in various embodiments, data platform 12 supports dynamic query generation by automatic discovery of join relations via static or dynamic filtering key specifications among composable data sets. This allows a user of data platform 12 to be agnostic to modifications made to existing data sets as well as creation of new data sets. The extensible query interface also provides a declarative and configurable specification for optimizing internal data generation and derivations.

~~[00530]~~[00531] As will also be described in more detail below, data platform 12 is configured to dynamically translate user interactions (e.g., received via web app 120) into SQL queries (and without the user needing to know how to write queries). Such queries can then be performed (e.g., by query service 166) against any compatible backend (e.g., data store 30).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00081

~~[00531]~~[00532] Fig. 4H illustrates an example of a user interacting with a portion of an interface. When a user visits data platform 12 (e.g., via web app 120 using a browser), data is extracted from data store 30 as needed (e.g., by query service 166), to provide the user with information, such as the visualizations depicted variously herein). As the user continues to interact with such visualizations (e.g., clicking on graph nodes, entering text into search boxes, navigating between tabs (e.g., tab 455 vs. 465)), such interactions act as triggers that cause query service 166 to continue to obtain information from data store 30 as needed (and as described in more detail below).

~~[00532]~~[00533] In the example shown in Fig. 4H, user A is viewing a dashboard that provides various information about entity A users (455), during the time period March 2 at midnight – March 25 at 7pm (which she selected by interacting with region 456). Various statistical information is presented to user A in region 457. Region 458 presents a timeline of events that occurred during the selected time period. User A has opted to list only the critical, high, and medium events during the time period by clicking on the associated boxes (459-461). A total of 55 low severity, and 155 info-only events also occurred during the time period. Each time user A interacts with an element in Fig. 4H (e.g., clicks on box 461, clicks on link 464-1, or clicks on tab 465), her actions are translated/formalized into filters on the data set and used to dynamically generate SQL queries. The SQL queries are generated transparently to user A (and also to a designer of the user interface shown in Fig. 4H).

~~[00533]~~[00534] User A notes in the timeline (462) that a user, Harish, connected to a known bad server (examplebad.com) using wget, an event that has a critical severity level. User A can click on region 463 to expand details about the event inline (which will display, for example, the text "External connection made to known bad host examplebad.com at port 80 from application 'wget' running on host dev1.lacework.internal as user harish") directly below line 462. User A can also click on link 464-1, which will take her to a dossier for the event (depicted in Fig. 4I). As will be described in more detail below, a dossier is a template for a collection of visualizations.

~~[00534]~~[00535] As shown in interface 466, the event of Harish using wget to contact examplebad.com on March 16 was assigned an event ID of 9291 by data platform 12 (467). For convenience to user A, the event is also added to her dashboard in region 476 as a bookmark (468). A summary of the event is depicted in region 469. By interacting with boxes shown in region 470, user A can see a timeline of related events. In this case, user A has indicated that she would like to see other events involving the wget application (by clicking box 471). Events of critical and medium security involving wget occurred during the one hour window selected in region 472.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00082

~~[00535]~~[00536] Region 473 automatically provides user A with answers to questions that may be helpful to have answers to while investigating event 9291. If user A clicks on any of the links in the event description (474), she will be taken to a corresponding dossier for the link. As one example, suppose user A clicks on link 475. She will then be presented with interface 477 shown in Fig. 4J.

~~[00536]~~[00537] Interface 477 is an embodiment of a dossier for a domain. In this example, the domain is "examplebad.com," as shown in region 478. Suppose user A would like to track down more information about interactions entity A resources have made with examplebad.com between January 1 and March 20. She selects the appropriate time period in region 479 and information in the other portions of interface 477 automatically update to provide various information corresponding to the selected time frame. As one example, user A can see that contact was made with examplebad.com a total of 17 times during the time period (480), as well as a list of each contact (481). Various statistical information is also included in the dossier for the time period (482). If she scrolls down in interface 477, user A will be able to view various polygraphs associated with examplebad.com, such as an application-communication polygraph (483).

~~[00537]~~[00538] Data stored in data store 30 can be internally organized as an activity graph. In the activity graph, nodes are also referred to as Entities. Activities generated by Entities are modeled as directional edges between nodes. Thus, each edge is an activity between two Entities. One example of an Activity is a "login" Activity, in which a user Entity logs into a machine Entity (with a directed edge from the user to the machine). A second example of an Activity is a "launch" Activity, in which a parent process launches a child process (with a directed edge from the parent to the child). A third example of an Activity is a "DNS query" Activity, in which either a process or a machine performs a query (with a directed edge from the requestor to the answer, e.g., an edge from a process to www.example.com). A fourth example of an Activity is a network "connected to" Activity, in which processes, IP addresses, and listen ports can connect to each other (with a directed edge from the initiator to the server).

~~[00538]~~[00539] As will be described in more detail below, query service 166 provides either relational views or graph views on top of data stored in data store 30. Typically, a user will want to see data filtered using the activity graph. For example, if an entity was not involved in an activity in a given time period, that entity should be filtered out of query results. Thus, a request to show "all machines" in a given time frame will be interpreted as "show distinct machines that were active" during the time frame.

~~[00539]~~[00540] Query service 166 relies on three main data model elements: fields, entities, and filters. As used herein, a field is a collection of values with the same type (logical and physical).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00083

A field can be represented in a variety of ways, including: 1. a column of relations (table/view), 2. a return field from another entity, 3. an SQL aggregation (e.g., COUNT, SUM, etc.), 4. an SQL expression with the references of other fields specified, and 5. a nested field of a JSON object. As viewed by query service 166, an entity is a collection of fields that describe a data set. The data set can be composed in a variety of ways, including: 1. a relational table, 2. a parameterized SQL statement, 3. DynamicSQL created by a Java function, and 4. join/project/aggregate/subclass of other entities. Some fields are common for all entities. One example of such a field is a "first observed" timestamp (when first use of the entity was detected). A second example of such a field is the entity classification type (e.g., one of: 1. Machine (on which an agent is installed), 2. Process, 3. Binary, 4. UID, 5. IP, 6. DNS Information, 7. ListenPort, and 8. PType). A third example of such a field is a "last observed" timestamp.

~~[00540]~~[00541] A filter is an operator that: 1. takes an entity and field values as inputs, 2. a valid SQL expression with specific reference(s) of entity fields, or 3. is a conjunct/disjunct of filters. As will be described in more detail below, filters can be used to filter data in various ways, and limit data returned by query service 166 without changing the associated data set.

~~[00541]~~[00542] As mentioned above, a dossier is a template for a collection of visualizations. Each visualization (e.g., the box including chart 484) has a corresponding card, which identifies particular target information needed (e.g., from data store 30) to generate the visualization. In various embodiments, data platform 12 maintains a global set of dossiers/cards. Users of data platform 12 such as user A can build their own dashboard interfaces using preexisting dossiers/cards as components, and/or they can make use of a default dashboard (which incorporates various of such dossiers/cards).

~~[00542]~~[00543] A JSON file can be used to store multiple cards (e.g., as part of a query service catalog). A particular card is represented by a single JSON object with a unique name as a field name.

~~[00543]~~[00544] Each card may be described by the following named fields:

~~[00544]~~[00545] TYPE: the type of the card. Example values include:

~~[00545]~~[00546] * Entity (the default type)

~~[00546]~~[00547] * SQL

~~[00547]~~[00548] * Filters

~~[00548]~~[00549] * DynamicSQL

~~[00549]~~[00550] * graphFilter

~~[00550]~~[00551] * graph

~~[00551]~~[00552] * Function

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00084

~~[00552]~~[00553] * Template

~~[00553]~~[00554] PARAMETERS: a JSON array object that contains an array of parameter objects with the following fields:

~~[00554]~~[00555] * name (the name of the parameter)

~~[00555]~~[00556] * required (a Boolean flag indicating whether the parameter is required or not)

~~[00556]~~[00557] * default (a default value of the parameter)

~~[00557]~~[00558] * props (a generic JSON object for properties of the parameter. Possible values are: "utype" (a user defined type), and "scope" (an optional property to configure a namespace of the parameter))

~~[00558]~~[00559] * value (a value for the parameter - non-null to override the default value defined in nested source entities)

~~[00559]~~[00560] SOURCES: a JSON array object explicitly specifying references of input entities. Each source reference has the following attributes:

~~[00560]~~[00561] * name (the card/entity name or fully-qualified Table name)

~~[00561]~~[00562] * type (required for base Table entity)

~~[00562]~~[00563] * alias (an alias to access this source entity in other fields (e.g., returns, filters, groups, etc))

~~[00563]~~[00564] RETURNS: a required JSON array object of a return field object. A return field object can be described by the following attributes:

~~[00564]~~[00565] * field (a valid field name from a source entity)

~~[00565]~~[00566] * expr (a valid SQL scalar expression. References to input fields of source entities are specified in the format of #{Entity.Field}. Parameters can also be used in the expression in the format of ${ParameterName})

~~[00566]~~[00567] * type (the type of field, which is required for return fields specified by expr. It is also required for all return fields of an Entity with an SQL type)

~~[00567]~~[00568] * alias (the unique alias for return field)

~~[00568]~~[00569] * aggr (possible aggregations are: COUNT, COUNT_DISTINCT, DISTINCT, MAX, MIN, AVG, SUM, FIRST_VALUE, LAST_VALUE)

~~[00569]~~[00570] * case (JSON array object represents conditional expressions "when" and "expr")

~~[00570]~~[00571] * fieldsFrom, and, except (specification for projections from a source entity with excluded fields)

~~[00571]~~[00572] * props (general JSON object for properties of the return field. Possible properties include: "filterGroup," "title," "format," and "utype")

~~[00572]~~[00573] PROPS: generic JSON objects for other entity properties

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00085

~~[00573]~~[00574] SQL: a JSON array of string literals for SQL statements. Each string literal can contain parameterized expressions ${ParameterName} and/or composable entity by #{EntityName}

~~[00574]~~[00575] GRAPH: required for graph entity. Has the following required fields:

~~[00575]~~[00576] * source (including "type," "props," and "keys")

~~[00576]~~[00577] * target (including "type," "props," and "keys")

~~[00577]~~[00578] * edge (including "type" and "props")

~~[00578]~~[00579] JOINS: a JSON array of join operators. Possible fields for a join operator include:

~~[00579]~~[00580] * type (possible join types include: "loj" - Left Outer Join, "join" - Inner Join, "in" - Semi Join, "implicit" - Implicit Join)

~~[00580]~~[00581] * left (a left hand side field of join)

~~[00581]~~[00582] * right (a right hand side field of join)

~~[00582]~~[00583] * keys (key columns for multi-way joins)

~~[00583]~~[00584] * order (a join order of multi-way joins)

~~[00584]~~[00585] FKEYS: a JSON array of FilterKey(s). The fields for a FilterKey are:

~~[00585]~~[00586] * type (type of FilterKey)

~~[00586]~~[00587] * fieldRefs (reference(s) to return fields of an entity defined in the sources field)

~~[00587]~~[00588] * alias (an alias of the FilterKey, used in implicit join specification)

~~[00588]~~[00589] FILTERS: a JSON array of filters (conjunct). Possible fields for a filter include:

~~[00589]~~[00590] * type (types of filters, including: "eq" - equivalent to SQL =, "ne" - equivalent to SQL <>, "ge" - equivalent to SQL >=, "gt" - equivalent to SQL >, "le" - equivalent to SQL <=, "lt" - equivalent to SQL <, "like" - equivalent to SQL LIKE, "not_like" - equivalent to SQL NOT LIKE, "rlike" - equivalent to SQL RLIKE (Snowflake specific), "not_rlike" - equivalent to SQL NOT RLIKE (Snowflake specific), "in" - equivalent to SQL IN, "not_in" - equivalent to SQL NOT IN)

~~[00590]~~[00591] * expr (generic SQL expression)

~~[00591]~~[00592] * field (field name)

~~[00592]~~[00593] * value (single value)

~~[00593]~~[00594] * values (for both IN and NOT IN)

~~[00594]~~[00595] ORDERS: a JSON array of ORDER BY for returning fields. Possible attributes for the ORDER BY clause include:

~~[00595]~~[00596] * field (field ordinal index (1 based) or field alias)

~~[00596]~~[00597] * order (asc/desc, default is ascending order)

~~[00597]~~[00598] GROUPS: a JSON array of GROUP BY for returning fields. Field attributes are:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00086

~~[00598]~~[00599] * field (ordinal index (1 based) or alias from the return fields)

~~[00599]~~[00600] LIMIT: a limit for the number of records to be returned

~~[00600]~~[00601] OFFSET: an offset of starting position of returned data. Used in combination with limit for pagination.

~~[00601]~~[00602] Suppose customers of data platform 12 (e.g., entity A and entity B) request new data transformations or a new aggregation of data from an existing data set (as well as a corresponding visualization for the newly defined data set). As mentioned above, the data models and filtering applications used by data platform 12 are extensible. Thus, two example scenarios of extensibility are (1) extending the filter data set, and (2) extending a FilterKey in the filter data set.

~~[00602]~~[00603] Data platform 12 includes a query service catalog that enumerates cards available to users of data platform 12. New cards can be included for use in data platform 12 by being added to the query service catalog (e.g., by an operator of data platform 12). For reusability and maintainability, a single external-facing card (e.g., available for use in a dossier) can be composed of multiple (nested) internal cards. Each newly added card (whether external or internal) will also have associated FilterKey(s) defined. A user interface (UI) developer can then develop a visualization for the new data set in one or more dossier templates. The same external card can be used in multiple dossier templates, and a given external card can be used multiple times in the same dossier (e.g., after customization). Examples of external card customization include customization via parameters, ordering, and/or various mappings of external data fields (columns).

~~[00603]~~[00604] As mentioned above, a second extensibility scenario is one in which a FilterKey in the filter data set is extended (i.e., existing template functions are used to define a new data set). As also mentioned above, data sets used by data platform 12 are composable/reusable/extensible, irrespective of whether the data sets are relational or graph data sets. One example data set is the User Tracking polygraph, which is generated as a graph data set (comprising nodes and edges). Like other polygraphs, User Tracking is an external data set that can be visualized both as a graph (via the nodes and edges) and can also be used as a filter data set for other cards, via the cluster identifier (CID) field.

~~[00604]~~[00605] As mentioned above, as users such as user A navigate through/interact with interfaces provided by data platform 12 (e.g., as shown in Fig. 4H), such interactions trigger query service 166 to generate and perform queries against data store 30. Dynamic composition of filter datasets can be implemented using FilterKeys and FilterKey Types. A FilterKey can be defined as a list of columns and/or fields in a nested structure (e.g., JSON). Instances of the same FilterKey Type can be formed as an Implicit Join Group. The same instance of a FilterKey can

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00087

participate in different Implicit Join Groups. A list of relationships among all possible Implicit Join Groups is represented as a Join graph for the entire search space to create a final data filter set by traversing edges and producing Join Path(s).

~~[00605]~~[00606] Each card (e.g., as stored in the query service catalog and used in a dossier) can be introspected by a /card/describe/CardID REST request.

~~[00606]~~[00607] At runtime (e.g., whenever it receives a request from web app 120), query service 166 parses the list of implicit joins and creates a Join graph to manifest relationships of FilterKeys among Entities. A Join graph (an example of which is depicted in Fig. 4K) comprises a list of Join Link(s). A Join Link represents each implicit join group by the same FilterKey type. A Join Link maintains a reverse map (Entity-to-FilterKey) of FilterKeys and their Entities. As previously mentioned, Entities can have more than one FilterKey defined. The reverse map guarantees one FilterKey per Entity can be used for each JoinLink. Each JoinLink also maintains a list of entities for the priority order of joins. Each JoinLink is also responsible for creating and adding directional edge(s) to graphs. An edge represents a possible join between two Entities.

~~[00607]~~[00608] At runtime, each Implicit Join uses the Join graph to find all possible join paths. The search of possible join paths starts with the outer FilterKey of an implicit join. One approach is to use a shortest path approach, with breadth first traversal and subject to the following criteria:

~~[00608]~~[00609] * Use the priority order list of Join Links for all entities in the same implicit join group.

~~[00609]~~[00610] * Stop when a node (Entity) is reached which has local filter(s).

~~[00610]~~[00611] * Include all join paths at the same level (depth).

~~[00611]~~[00612] * Exclude join paths based on the predefined rules (path of edges).

~~[00612]~~[00613] Fig. 4L illustrates an example of a process for dynamically generating and executing a query. In various embodiments, process 485 is performed by data platform 12. The process begins at 486 when a request is received to filter information associated with activities within a network environment. One example of such a request occurs in response to user A clicking on tab 465. Another example of such a request occurs in response to user A clicking on link 464-1. Yet another example of such a request occurs in response to user A clicking on link 464-2 and selecting (e.g., from a dropdown) an option to filter (e.g., include, exclude) based on specific criteria that she provides (e.g., an IP address, a username, a range of criteria, etc.).

~~[00613]~~[00614] At 487, a query is generated based on an implicit join. One example of processing that can be performed at 487 is as follows. As explained above, one way dynamic composition of filter datasets can be implemented is by using FilterKeys and FilterKey Types. And, instances of the same FilterKey Type can be formed as an Implicit Join Group. A Join graph for the entire search space can be constructed from a list of all relationships among all

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00088

possible Join Groups. And, a final data filter set can be created by traversing edges and producing one or more Join Paths. Finally, the shortest path in the join paths is used to generate an SQL query string.

~~[00614]~~[00615] One approach to generating an SQL query string is to use a query building library (authored in an appropriate language such as Java). For example, a common interface "sqlGen" may be used in conjunction with process 485 is as follows. First, a card/entity is composed by a list of input cards/entities, where each input card recursively is composed by its own list of input cards. This nested structure can be visualized as a tree of query blocks(SELECT) in standard SQL constructs. SQL generation can be performed as the traversal of the tree from root to leaf entities (top-down), calling the sqlGen of each entity. Each entity can be treated as a subclass of the Java class(Entity). An implicit join filter (EntityFilter) is implemented as a subclass of Entity, similar to the right hand side of a SQL semi-join operator. Unlike the static SQL semi-join construct, it is conditionally and recursively generated even if it is specified in the input sources of the JSON specification. Another recursive interface can also be used in conjunction with process 485, preSQLGen, which is primarily the entry point for EntityFilter to run a search and generate nested implicit join filters. During preSQLGen recursive invocations, the applicability of implicit join filters is examined and pushed down to its input subquery list. Another top-down traversal, pullUpCachable, can be used to pull up common sub-query blocks, including those dynamically generated by preSQLGen, such that SELECT statements of those cacheable blocks are generated only once at top-level WITH clauses. A recursive interface, sqlWith, is used to generate nested subqueries inside WITH clauses. The recursive calls of a sqlWith function can generate nested WITH clauses as well. An sqlFrom function can be used to generate SQL FROM clauses by referencing those subquery blocks in the WITH clauses. It also produces INNER/OUTER join operators based on the joins in the specification. Another recursive interface, sqlWhere, can be used to generate conjuncts and disjuncts of local predicates and semi-join predicates based on implicit join transformations. Further, sqlProject, sqlGroupBy, sqlOrderBy, and sqlLimitOffset can respectively be used to translate the corresponding directives in JSON spec to SQL SELECT list, GROUP BY, ORDER BY, and LIMIT/OFFSET clauses.

~~[00615]~~[00616] Returning to process 485, at 488, the query (generated at 487) is used to respond to the request. As one example of the processing performed at 488, the generated query is used to query data store 30 and provide (e.g., to web app 120) fact data formatted in accordance with a schema (e.g., as associated with a card associated with the request received at 486).

~~[00616]~~[00617] Although the examples described herein largely relate to embodiments where data is collected from agents and ultimately stored in a data store such as those provided by

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00089

Snowflake, in other embodiments data that is collected from agents and other sources may be stored in different ways. For example, data that is collected from agents and other sources may be stored in a data warehouse, data lake, data mart, and/or any other data store.

~~[00617]~~[00618] A data warehouse may be embodied as an analytic database (e.g., a relational database) that is created from two or more data sources. Such a data warehouse may be leveraged to store historical data, often on the scale of petabytes. Data warehouses may have compute and memory resources for running complicated queries and generating reports. Data warehouses may be the data sources for business intelligence ('BI') systems, machine learning applications, and/or other applications. By leveraging a data warehouse, data that has been copied into the data warehouse may be indexed for good analytic query performance, without affecting the write performance of a database (e.g., an Online Transaction Processing ('OLTP') database). Data warehouses also enable the joining data from multiple sources for analysis. For example, a sales OLTP application probably has no need to know about the weather at various sales locations, but sales predictions could take advantage of that data. By adding historical weather data to a data warehouse, it would be possible to factor it into models of historical sales data.

~~[00618]~~[00619] Data lakes, which store files of data in their native format, may be considered as "schema on read" resources. As such, any application that reads data from the lake may impose its own types and relationships on the data. Data warehouses, on the other hand, are "schema on write," meaning that data types, indexes, and relationships are imposed on the data as it is stored in the EDW. "Schema on read" resources may be beneficial for data that may be used in several contexts and poses little risk of losing data. "Schema on write" resources may be beneficial for data that has a specific purpose, and good for data that must relate properly to data from other sources. Such data stores may include data that is encrypted using homomorphic encryption, data encrypted using privacy-preserving encryption, smart contracts, non-fungible tokens, decentralized finance, and other techniques.

~~[00619]~~[00620] Data marts may contain data oriented towards a specific business line whereas data warehouses contain enterprise-wide data. Data marts may be dependent on a data warehouse, independent of the data warehouse (e.g., drawn from an operational database or external source), or a hybrid of the two. In embodiments described herein, different types of data stores (including combinations thereof) may be leveraged. Such data stores may be proprietary or may be embodied as vendor provided products or services such as, for example, Google BigQuery, Druid, Amazon Redshift, IBM Db2, Dremio, Databricks Lakehouse Platform, Cloudera, Azure Synapse Analytics, and others.

~~[00620]~~[00621] The deployments (e.g., a customer's cloud deployment) that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00090

include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may be provisioned, deployed, and/or managed using infrastructure as code ('IaC'). IaC involves the managing and/or provisioning of infrastructure through code instead of through manual processes. With IaC, configuration files may be created that include infrastructure specifications. IaC can be beneficial as configurations may be edited and distributed, while also ensuring that environments are provisioned in a consistent manner. IaC approaches may be enabled in a variety of ways including, for example, using IaC software tools such as Terraform by HashiCorp. Through the usage of such tools, users may define and provide data center infrastructure using JavaScript Object Notation ('JSON'), YAML, proprietary formats, or some other format. In some embodiments, the configuration files may be used to emulate a cloud deployment for the purposes of analyzing the emulated cloud deployment using the systems described herein. Likewise, the configuration files themselves may be used as inputs to the systems described herein, such that the configuration files may be inspected to identify vulnerabilities, misconfigurations, violations of regulatory requirements, or other issues. In fact, configuration files for multiple cloud deployments may even be used by the systems described herein to identify best practices, to identify configuration files that deviate from typical configuration files, to identify configuration files with similarities to deployments that have been determined to be deficient in some way, or the configuration files may be leveraged in some other ways to detect vulnerabilities, misconfigurations, violations of regulatory requirements, or other issues prior to deploying an infrastructure that is described in the configuration files. In some embodiments the techniques described herein may be use in multi-cloud, multi-tenant, cross-cloud, cross-tenant, cross-user, industry cloud, digital platform, and other scenarios depending on specific need or situation.

~~[00621]~~[00622] In some embodiments, the deployments that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may be monitored to determine the extent to which a particular component has experienced "drift" relative to its associated IaC configuration. Discrepancies between how cloud resources were defined in an IaC configuration file and how they are currently configured in runtime may be identified and remediation workflows may be initiated to generate an alert, reconfigure the deployment, or take some other action. Such discrepancies may occur for a variety of reasons. Such discrepancies may occur, for example, due to maintenance operations being performed, due to incident response tasks being carried out, or for some other reason. Readers will appreciate that while IaC helps avoid initial misconfigurations of a deployment by codifying and enforcing resource creation, resource configuration, security policies, and so on,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00091

the systems described herein may prevent unwanted drift from occurring during runtime and after a deployment has been created in accordance with an IaC configuration.

~~[00622]~~[00623] In some embodiments, the deployments (e.g., a customer's cloud deployment) that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may also be provisioned, deployed, and/or managed using security as code ('SaC'). SaC extends IaC concepts by defining cybersecurity policies and/or standards programmatically, so that the policies and/or standards can be referenced automatically in the configuration scripts used to provision cloud deployments. Stated differently, SaC can automate policy implementation and cloud deployments may even be compared with the policies to prevent "drift." For example, if a policy is created where all personally identifiable information ('PII') or personal health information ('PHI') must be encrypted when it is stored, that policy is translated into a process that is automatically launched whenever a developer submits code, and code that violates the policy may be automatically rejected.

~~[00623]~~[00624] In some embodiments, SaC may be implemented by initially classifying workloads (e.g., by sensitivity, by criticality, by deployment model, by segment). Policies that can be instantiated as code may subsequently be designed. For example, compute-related policies may be designed, access-related policies may be designed, application-related policies may be designed, network-related policies may be designed, data-related policies may be designed, and so on. Security as code may then be instantiated through architecture and automation, as successful implementation of SaC can benefit from making key architectural-design decisions and executing the right automation capabilities. Next, operating model protections may be built and supported. For example, an operating model may "shift left" to maximize self-service and achieve full-life-cycle security automation (e.g., by standardizing common development toolchains, CI/CD pipelines, and the like). In such an example, security policies and access controls may be part of the pipeline, automatic code review and bug/defect detection may be performed, automated build processes may be performed, vulnerability scanning may be performed, checks against a risk-control framework may be made, and other tasks may be performed all before deploying an infrastructure or components thereof.

~~[00624]~~[00625] The systems described herein may be useful in analyzing, monitoring, evaluating, or otherwise observing a GitOps environment. In a GitOps environment, Git may be viewed as the one and only source of truth. As such, GitOps may require that the desired state of infrastructure (e.g., a customer's cloud deployment) be stored in version control such that the entire audit trail of changes to such infrastructure can be viewed or audited. In a GitOps

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00092

environment, all changes to infrastructure are embodied as fully traceable commits that are associated with committer information, commit IDs, time stamps, and/or other information. In such an embodiment, both an application and the infrastructure (e.g., a customer's cloud deployment) that supports the execution of the application are therefore versioned artifacts and can be audited using the gold standards of software development and delivery. Readers will appreciate that while the systems described herein are described as analyzing, monitoring, evaluating, or otherwise observing a GitOps environment, in other embodiments other source control mechanisms may be utilized for creating infrastructure, making changes to infrastructure, and so on. In these embodiments, the systems described herein may similarly be used for analyzing, monitoring, evaluating, or otherwise observing such environments.

~~[00625]~~[00626] As described in other portions of the present disclosure, the systems described herein may be used to analyze, monitor, evaluate, or otherwise observe a customer's cloud deployment. While securing traditional datacenters requires managing and securing an IP-based perimeter with networks and firewalls, hardware security modules ('HSMs'), security information and event management ('SIEM') technologies, and other physical access restrictions, such solutions are not particularly useful when applied to cloud deployments. As such, the systems described herein may be configured to interact with and even monitor other solutions that are appropriate for cloud deployments such as, for example, "zero trust" solutions.

~~[00626]~~[00627] A zero trust security model (a.k.a., zero trust architecture) describes an approach to the design and implementation of IT systems. A primary concept behind zero trust is that devices should not be trusted by default, even if they are connected to a managed corporate network such as the corporate LAN and even if they were previously verified. Zero trust security models help prevent successful breaches by eliminating the concept of trust from an organization's network architecture. Zero trust security models can include multiple forms of authentication and authorization (e.g., machine authentication and authorization, human/user authentication and authorization) and can also be used to control multiple types of accesses or interactions (e.g., machine-to-machine access, human-to-machine access).

~~[00627]~~[00628] In some embodiments, the systems described herein may be configured to interact with zero trust solutions in a variety of ways. For example, agents that collect input data for the systems described herein (or other components of such systems) may be configured to access various machines, applications, data sources, or other entity through a zero trust solution, especially where local instances of the systems described herein are deployed at edge locations. Likewise, given that zero trust solutions may be part of a customer's cloud deployment, the zero trust solution itself may be monitored to identify vulnerabilities, anomalies, and so on. For example, network traffic to and from the zero trust solution may be analyzed, the zero trust

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00093

solution may be monitored to detect unusual interactions, log files generated by the zero trust solution may be gathered and analyzed, and so on.

~~[00628]~~[00629] In some embodiments, the systems described herein may leverage various tools and mechanisms in the process of performing its primary tasks (e.g., monitoring a cloud deployment). For example, Linux eBPF is mechanism for writing code to be executed in the Linux kernel space. Through the usage of eBPF, user mode processes can hook into specific trace points in the kernel and access data structures and other information. For example, eBPF may be used to gather information that enables the systems described herein to attribute the utilization of networking resources or network traffic to specific processes. This may be useful in analyzing the behavior of a particular process, which may be important for observability/SIEM.

~~[00629]~~[00630] The systems described may be configured to collect security event logs (or any other type of log or similar record of activity) and telemetry in real time for threat detection, for analyzing compliance requirements, or for other purposes. In such embodiments, the systems described herein may analyze telemetry in real time (or near real time), as well as historical telemetry, to detect attacks or other activities of interest. The attacks or activities of interest may be analyzed to determine their potential severity and impact on an organization. In fact, the attacks or activities of interest may be reported, and relevant events, logs, or other information may be stored for subsequent examination.

~~[00630]~~[00631] In one embodiment, systems described herein may be configured to collect security event logs (or any other type of log or similar record of activity) and telemetry in real time to provide customers with a SIEM or SIEM-like solution. SIEM technology aggregates event data produced by security devices, network infrastructure, systems, applications, or other source. Centralizing all of the data that may be generated by a cloud deployment may be challenging for a traditional SIEM, however, as each component in a cloud deployment may generate log data or other forms of machine data, such that the collective amount of data that can be used to monitor the cloud deployment can grow to be quite large. A traditional SIEM architecture, where data is centralized and aggregated, can quickly result in large amounts of data that may be expensive to store, process, retain, and so on. As such, SIEM technologies may frequently be implemented such that silos are created to separate the data.

~~[00631]~~[00632] In some embodiments of the present disclosure, data that is ingested by the systems described herein may be stored in a cloud-based data warehouse such as those provided by Snowflake and others. Given that companies like Snowflake offer data analytics and other services to operate on data that is stored in their data warehouses, in some embodiments one or more of the components of the systems described herein may be deployed in or near Snowflake as part of a secure data lake architecture (a.k.a., a security data lake architecture, a security data

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00094

lake/warehouse). In such an embodiment, components of the systems described herein may be deployed in or near Snowflake to collect data, transform data, analyze data for the purposes of detecting threats or vulnerabilities, initiate remediation workflows, generate alerts, or perform any of the other functions that can be performed by the systems described herein. In such embodiments, data may be received from a variety of sources (e.g., EDR or EDR-like tools that handle endpoint data, cloud access security broker ('CASB') or CASB-like tools that handle data describing interactions with cloud applications, Identity and Access Management ('IAM') or IAM-like tools, and many others), normalized for storage in a data warehouse, and such normalized data may be used by the systems described herein. In fact, the systems described herein may actually implement the data sources (e.g., an EDR tool, a CASB tool, an IAM tool) described above.

~~[00632]~~[00633] In some embodiments one data source that is ingested by the systems described herein is log data, although other forms of data such as network telemetry data (flows and packets) and/or many other forms of data may also be utilized. In some embodiments, event data can be combined with contextual information about users, assets, threats, vulnerabilities, and so on, for the purposes of scoring, prioritization and expediting investigations. In some embodiments, input data may be normalized, so that events, data, contextual information, or other information from disparate sources can be analyzed more efficiently for specific purposes (e.g., network security event monitoring, user activity monitoring, compliance reporting). The embodiments described here offer real-time analysis of events for security monitoring, advanced analysis of user and entity behaviors, querying and long-range analytics for historical analysis, other support for incident investigation and management, reporting (for compliance requirements, for example), and other functionality.

~~[00633]~~[00634] In some embodiments, the systems described herein may be part of an application performance monitoring ('APM') solution. APM software and tools enable the observation of application behavior, observation of its infrastructure dependencies, observation of users and business key performance indicators ('KPIs') throughout the application's life cycle, and more. The applications being observed may be developed internally, as packaged applications, as software as a service ('SaaS'), or embodied in some other ways. In such embodiments, the systems described herein may provide one or more of the following capabilities:

~~[00634]~~[00635] The ability to operate as an analytics platform that ingests, analyzes, and builds context from traces, metrics, logs, and other sources.

~~[00635]~~[00636] Automated discovery and mapping of an application and its infrastructure components.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00095

~~[00636]~~[00637] Observation of an application's complete transactional behavior, including interactions over a data communications network.

~~[00637]~~[00638] Monitoring of applications running on mobile (native and browser) and desktop devices.

~~[00638]~~[00639] Identification of probable root causes of an application's performance problems and their impact on business outcomes.

~~[00639]~~[00640] Integration capabilities with automation and service management tools.

~~[00640]~~[00641] Analysis of business KPIs and user journeys (for example, login to check-out).

~~[00641]~~[00642] Domain-agnostic analytics capabilities for integrating data from third-party sources.

~~[00642]~~[00643] Endpoint monitoring to understand the user experience and its impact on business outcomes.

~~[00643]~~[00644] Support for virtual desktop infrastructure ('VDI') monitoring.

~~[00644]~~[00645] In embodiments where the systems described herein are used for APM, some components of the system may be modified, other components may be added, some components may be removed, and other components may remain the same. In such an example, similar mechanisms as described elsewhere in this disclosure may be used to collect information from the applications, network resources used by the application, and so on. The graph based modelling techniques may also be leveraged to perform some of the functions mentioned above, or other functions as needed.

~~[00645]~~[00646] In some embodiments, the systems described herein may be part of a solution for developing and/or managing artificial intelligence ('AI') or machine learning ('ML') applications. For example, the systems described herein may be part of an AutoML tool that automate the tasks associated with developing and deploying ML models. In such an example, the systems described herein may perform various functions as part of an AutoML tool such as, for example, monitoring the performance of a series of processes, microservices, and so on that are used to collectively form the AutoML tool. In other embodiments, the systems described herein may perform other functions as part of an AutoML tool or may be used to monitor, analyze, or otherwise observe an environment that the AutoML tool is deployed within.

~~[00646]~~[00647] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include other forms of AI/ML tools. For example, the systems described herein may manage, analyze, or otherwise observe deployments that include AI services. AI services are, like other resources in an as-a-service model, ready-made models and AI applications that are consumable as services and made available through APIs. In such an example, rather than using their own data to build and train models for common

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00096

activities, organizations may access pre-trained models that accomplish specific tasks. Whether an organization needs natural language processing ('NLP'), automatic speech recognition ('ASR'), image recognition, or some other capability, AI services simply plug-and-play into an application through an API. Likewise, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include other forms of AI/ML tools such as Amazon Sagemaker (or other cloud machine-learning platform that enables developers to create, train, and deploy ML models) and related services such as Data Wrangler (a service to accelerate data prep for ML) and Pipelines (a CI/CD service for ML).

~~[00647]~~[00648] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include various data services. For example, data services may include secure data sharing services, data marketplace services, private data exchanges services, and others. Secure data sharing services can allow access to live data from its original location, where those who are granted access to the data simply reference the data in a controlled and secure manner, without latency or contention from concurrent users. Because changes to data are made to a single version, data remains up-to-date for all consumers, which ensures data models are always using the latest version of such data. Data marketplace services operate as a single location to access live, ready-to-query data (or data that is otherwise ready for some other use). A data marketplace can even include a "feature stores," which can allow data scientists to repurpose existing work. For example, once a data scientist has converted raw data into a metric (e.g., costs of goods sold), this universal metric can be found quickly and used by other data scientists for quick analysis against that data.

~~[00648]~~[00649] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include distributed training engines or similar mechanisms such as, for example, such as tools built on Dask. Dask is an open source library for parallel computing that is written in Python. Dask is designed to enable data scientists to improve model accuracy faster, as Dask enables data scientists can do everything in Python end-to-end, which means that they no longer need to convert their code to execute in environments like Apache Spark. The result is reduced complexity and increased efficiency. The systems described herein may also be used to manage, analyze, or otherwise observe deployments that include technologies such as RAPIDS (an open source Python framework which is built on top of Dask). RAPIDS optimizes compute time and speed by providing data pipelines and executing data science code entirely on graphics processing units (GPUs) rather than CPUs. Multi-cluster, shared data architecture, DataFrames, Java user-defined functions (UDF) are supported to enable trained models to run within a data warehouse.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00097

~~[00649]~~[00650] In some embodiments, the systems described herein may be leveraged for the specific use case of detecting and/or remediating ransomware attacks and/or other malicious action taken with respect to data, systems, and/or other resources associated with one or more entities. Ransomware is a type of malware from cryptovirology that threatens to publish the victim's data or perpetually block access to such data unless a ransom is paid. In such embodiments, ransomware attacks may be carried out in a manner such that patterns (e.g., specific process-to-process communications, specific data access patterns, unusual amounts of encryption/re-encryption activities) emerge, where the systems described herein may monitor for such patterns. Alternatively, ransomware attacks may involve behavior that deviates from normal behavior of a cloud deployment that is not experiencing a ransomware attack, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a ransomware attack.

~~[00650]~~[00651] In some embodiments, particular policies may be put in place the systems described herein may be configured to enforce such policies as part of an effort to thwart ransomware attacks. For example, particular network sharing protocols (e.g., Common Internet File System ('CIFS'), Network File System ('NFS')) may be avoided when implementing storage for backup data, policies that protect backup systems may be implemented and enforced to ensure that usable backups are always available, multifactor authentication for particular accounts may be utilized and accounts may be configured with the minimum privilege required to function, isolated recovery environments may be created and isolation may be monitored and enforced to ensure the integrity of the recovery environment, and so on. As described in the present disclosure, the systems described herein may be configured to explicitly enforce such policies or may be configured to detect unusual activity that represents a violation of such policies, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a violation of a particular policy.

~~[00651]~~[00652] Readers will appreciate that ransomware attacks are often deployed as part of a larger attack that may involve, for example:

~~[00652]~~[00653] Penetration of the network through means such as, for example, stolen credentials and remote access malware.

~~[00653]~~[00654] Stealing of credentials for critical system accounts, including subverting critical administrative accounts that control systems such as backup, Active Directory ('AD'), DNS, storage admin consoles, and/or other key systems.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00098

~~[00654]~~[00655] Attacks on a backup administration console to turn off or modify backup jobs, change retention policies, or even provide a roadmap to where sensitive application data is stored.

~~[00655]~~[00656] Data theft attacks.

~~[00656]~~[00657] As a result of the many aspects that are part of a ransomware attack, embodiments of the present disclosure may be configured as follows:

~~[00657]~~[00658] The systems may include one or more components that detect malicious activity based on the behavior of a process.

~~[00658]~~[00659] The systems may include one or more components that store indicator of compromise ('IOC') or indicator of attack ('IOA') data for retrospective analysis.

~~[00659]~~[00660] The systems may include one or more components that detect and block fileless malware attacks.

~~[00660]~~[00661] The systems may include one or more components that remove malware automatically when detected.

~~[00661]~~[00662] The systems may include a cloud-based, SaaS-style, multitenant infrastructure.

~~[00662]~~[00663] The systems may include one or more components that identify changes made by malware and provide the recommended remediation steps or a rollback capability.

~~[00663]~~[00664] The systems may include one or more components that detect various application vulnerabilities and memory exploit techniques.

~~[00664]~~[00665] The systems may include one or more components that continue to collect suspicious event data even when a managed endpoint is outside of an organization's network.

~~[00665]~~[00666] The systems may include one or more components that perform static, on-demand malware detection scans of folders, drives, devices, or other entities.

~~[00666]~~[00667] The systems may include data loss prevention (DLP) functionality.

~~[00667]~~[00668] In some embodiments, the systems described herein may manage, analyze, or otherwise observe deployments that include deception technologies. Deception technologies allow for the use of decoys that may be generated based on scans of true network areas and data. Such decoys may be deployed as mock networks running on the same infrastructure as the real networks, but when an intruder attempts to enter the real network, they are directed to the false network and security is immediately notified. Such technologies may be useful for detecting and stopping various types of cyber threats such as, for example, Advanced Persistent Threats ('APTs'), malware, ransomware, credential dumping, lateral movement and malicious insiders. To continue to outsmart increasingly sophisticated attackers, these solutions may continuously deploy, support, refresh and respond to deception alerts.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00099

~~[00668]~~[00669] In some embodiments, the systems described herein may manage, analyze, or otherwise observe deployments that include various authentication technologies, such as multi-factor authentication and role-based authentication. In fact, the authentication technologies may be included in the set of resources that are managed, analyzed, or otherwise observed as interactions with the authentication technologies may monitored. Likewise, log files or other information retained by the authentication technologies may be gathered by one or more agents and used as input to the systems described herein.

~~[00669]~~[00670] In some embodiments, the systems described herein may be leveraged for the specific use case of detecting supply chain attacks. More specifically, the systems described herein may be used to monitor a deployment that includes software components, virtualized hardware components, and other components of an organization's supply chain such that interactions with an outside partner or provider with access to an organization's systems and data can be monitored. In such embodiments, supply chain attacks may be carried out in a manner such that patterns (e.g., specific interactions between internal and external systems) emerge, where the systems described herein may monitor for such patterns. Alternatively, supply chain attacks may involve behavior that deviates from normal behavior of a cloud deployment that is not experiencing a supply chain attack, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a supply chain attack.

~~[00670]~~[00671] In some embodiments, the systems described herein may be leveraged for other specific use cases such as, for example, detecting the presence of (or preventing infiltration from) cryptocurrency miners (e.g., bitcoin miners), token miners, hashing activity, non-fungible token activity, other viruses, other malware, and so on. As described in the present disclosure, the systems described herein may monitor for such threats using known patterns or by detecting unusual activity, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a particular type of threat, intrusion, vulnerability, and so on.

~~[00671]~~[00672] The systems described herein may also be leveraged for endpoint protection, such the systems described herein form all of or part of an endpoint protection platform. In such an embodiment, agents, sensors, or similar mechanisms may be deployed on or near managed endpoints such as computers, servers, virtualized hardware, internet of things ('IotT') devices, mobile devices, phones, tablets, watches, other personal digital devices, storage devices, thumb drives, secure data storage cards, or some other entity. In such an example, the endpoint protection platform may provide functionality such as:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00100

~~[00672]~~[00673] Prevention and protection against security threats including malware that uses file-based and fileless exploits.

~~[00673]~~[00674] The ability to apply control (allow/block) to access of software, scripts, processes, microservices, and so on.

~~[00674]~~[00675] The ability to detect and prevent threats using behavioral analysis of device activity, application activity, user activity, and/or other data.

~~[00675]~~[00676] The ability for facilities to investigate incidents further and/or obtain guidance for remediation when exploits evade protection controls

~~[00676]~~[00677] The ability to collect and report on inventory, configuration and policy management of the endpoints.

~~[00677]~~[00678] The ability to manage and report on operating system security control status for the monitored endpoints.

~~[00678]~~[00679] The ability to scan systems for vulnerabilities and report/manage the installation of security patches.

~~[00679]~~[00680] The ability to report on internet, network and/or application activity to derive additional indications of potentially malicious activity.

~~[00680]~~[00681] Example embodiments are described in which policy enforcement, threat detection, or some other function is carried out by the systems described herein by detecting unusual activity, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a particular type of threat, intrusion, vulnerability, and so on. Although these examples are largely described in terms of identifying unusual activity, in these examples the systems described herein may be configured to learn what constitutes 'normal activity' – where 'normal activity' is activity observed, modeled, or otherwise identified in the absence of a particular type of threat, intrusion, vulnerability, and so on. As such, detecting 'unusual activity' may alternatively be viewed as detecting a deviation from 'normal activity' such that 'unusual activity' does not need to be identified and sought out. Instead, deviations from 'normal activity' may be assumed to be 'unusual activity'.

~~[00681]~~[00682] Readers will appreciate that while specific examples of the functionality that the systems described herein can provide are included in the present disclosure, such examples are not to be interpreted as limitations as to the functionality that the systems described herein can provide. Other functionality may be provided by the systems described herein, all of which are within the scope of the present disclosure. For the purposes of illustration and not as a limitation, additional examples can include governance, risk, and compliance ('GRC'), threat detection and

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00101

incident response, identity and access management, network and infrastructure security, data protection and privacy, identity and access management ("IAM"), and many others.

~~[00682]~~[00683] In order to provide the functionality described above, the systems described herein or the deployments that are monitored by such systems may implement a variety of techniques. For example, the systems described herein or the deployments that are monitored by such systems may tag data and logs to provide meaning or context, persistent monitoring techniques may be used to monitor a deployment at all times and in real time, custom alerts may be generated based on rules, tags, and/or known baselines from one or more polygraphs, and so on.

~~[00683]~~[00684] Although examples are described above where data may be collected from one or more agents, in some embodiments other methods and mechanisms for obtaining data may be utilized. For example, some embodiments may utilize agentless deployments where no agent (or similar mechanism) is deployed on one or more customer devices, deployed within a customer's cloud deployment, or deployed at another location that is external to the data platform. In such embodiments, the data platform may acquire data through one or more APIs such as the APIs that are available through various cloud services. For example, one or more APIs that enable a user to access data captured by Amazon CloudTrail may be utilized by the data platform to obtain data from a customer's cloud deployment without the use of an agent that is deployed on the customer's resources. In some embodiments, agents may be deployed as part of a data acquisition service or tool that does not utilize a customer's resources or environment. In some embodiments, agents (deployed on a customer's resources or elsewhere) and mechanisms in the data platform that can be used to obtain data from through one or more APIs such as the APIs that are available through various cloud services may be utilized. In some embodiments, one or more cloud services themselves may be configured to push data to some entity (deployed anywhere), which may or may not be an agent. In some embodiments, other data acquisition techniques may be utilized, including combinations and variations of the techniques described above, each of which is within the scope of the present disclosure.

~~[00684]~~[00685] Readers will appreciate that while specific examples of the cloud deployments that may be monitored, analyzed, or otherwise observed by the systems described herein have been provided, such examples are not to be interpreted as limitations as to the types of deployments that may be monitored, analyzed, or otherwise observed by the systems described herein. Other deployments may be monitored, analyzed, or otherwise observed by the systems described herein, all of which are within the scope of the present disclosure. For the purposes of illustration and not as a limitation, additional examples can include multi-cloud deployments, on-premises environments, hybrid cloud environments, sovereign cloud environments,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00102

heterogeneous environments, DevOps environments, DevSecOps environments, GitOps environments, quantum computing environments, data fabrics, composable applications, composable networks, decentralized applications, and many others.

~~[00685]~~[00686] Readers will appreciate that while specific examples of the types of data that may be collected, transformed, stored, and/or analyzed by the systems described herein have been provided, such examples are not to be interpreted as limitations as to the types of data that may be collected, transformed, stored, and/or analyzed by the systems described herein. Other types of data can include, for example, data collected from different tools (e.g., DevOps tools, DevSecOps, GitOps tools), different forms of network data (e.g., routing data, network translation data, message payload data, Wi-Fi data, Bluetooth data, personal area networking data, payment device data, near field communication data, metadata describing interactions carried out over a network, and many others), data describing processes executing in a container, lambda, EC2 instance, virtual machine, or other execution environment), information describing the execution environment itself, and many other types of data.

~~[00686]~~[00687] In some embodiments, one or more of the components described above may be deployed using a secure access service edge ('SASE') model or similar model. In a SASE model, the services, functionality, or components described above may be deployed at edge devices (or relatively close to such edge device) such as a user's laptop, tablet, smartphone, or other device. In such a way, network security controls may be delivered on such edge devices. SASE capabilities may be delivered as a service based upon the identity of the entity, real-time context, enterprise security/compliance policies and continuous assessment of risk/trust throughout the sessions, where the identity of entities can be associated with people, groups of people, devices, applications, services, IoT systems or edge computing locations, and so on. In these embodiments, one or more of the components described above may be deployed at or near the edge devices, and the edge devices may even include local applications that are configured to utilize one or more of the components described above where the components are not deployed on the edge devices themselves.

~~[00687]~~[00688] For further explanation, Fig. 5 sets forth a flowchart illustrating an example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The cloud deployments 508, 514 may be similar to the cloud deployments described above, where a particular cloud deployment can include a variety of components 510, 512 such as one or more applications, one or more data sources, networking resources, processing resources, and other resources. Such components 510, 512 may, in some embodiments, be deployed in the cloud deployments 508, 514 using one or more as-a-service models where software, infrastructure,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00103

platforms, databases, and other components as delivered as services. Configuring cloud deployments 508, 514 based on learnings obtained by monitoring other cloud deployments may be carried out using the systems described above. As such, one or more of the steps depicted in Fig. 5 may be performed by the systems described above.

~~[00688]~~[00689] The example method depicted in Figure 5 includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508. Determining 502 normal behavior for one or more components 510 in a first cloud deployment 508 may be carried out, for example, as described in greater detail above (at times described as identifying 'normal activity') by the systems described above (also referred to herein as a 'data platform').

~~[00689]~~[00690] The example method depicted in Figure 5 also includes determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514. Determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514 may also be carried out, for example, as described in greater detail above (at times described as identifying 'normal activity') by the systems described above.

~~[00690]~~[00691] In some embodiments, a customer-specific data platform may be used to analyze, monitor, or otherwise observe a particular customer's cloud deployment (or some other deployment). Within such a cloud deployment, various clusters may exist. For example, a collection of microservices may form a cluster by virtue of those microservices communicating only (or mostly) with each other. Likewise, one or more cloud computing instances (e.g., one or more EC2 instances) and a database may form a cluster by virtue of the EC2 instances accessing the database as the only source of data utilized by the EC2 instances. Using the techniques and mechanisms described above, such clusters may be identified. Although clusters may be identified and characteristics associated with the cluster may be learned, limited insights may be gained if only a particular customer's cloud deployment is analyzed, monitored, or otherwise observed. In accordance with embodiments of the present disclosure, cross-customer analysis may be leveraged to gain deeper insights than would be gained if only a single customer's cloud deployment is analyzed, monitored, or otherwise observed.

~~[00691]~~[00692] In some embodiments, cross-customer analysis may be carried out by gathering information related to cloud deployments (or some other deployments) for multiple customers and comparing such information. Using the example described above, information describing clusters identified in a first customer's cloud deployment may be compared to information describing clusters identified in a second customer's cloud deployment for the purposes of identifying similar or identical clusters in each customer's cloud deployment.

~~[00692]~~[00693] Consider an example in which each customer's deployment included a web server that was deployed in one or more EC2 instances. In such an example, a particular cluster

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00104

that represents the web server may be identified in each customer's deployment. For example, a first cluster in the first customer's deployment may represent a first web server and a second cluster in the second customer's deployment may represent a second web server. Because the first cluster and the second cluster would have similar characteristics (e.g., each cluster receives data communications using HTTP or HTTPS or any other suitable communication protocol, each cluster communicates with a web browser, each cluster requires similar computing resources, and so on), the first cluster and the second cluster may be identified as being identical clusters by comparing the characteristics of each cluster that each cluster. This process may be repeated across the cloud deployments for many customers such that a collection of 'web server' clusters (in this example) may be identified.

~~[00693]~~[00694] Readers will appreciate that although the example described above relates to an embodiment where a collection of 'web server' clusters are identified in different customer's cloud deployments, identifying the nature or type (e.g., a web server) of the clusters is not required. In fact, by comparing the characteristics of different clusters to each other, similar or identical clusters may be identified even if the exact nature/type of those clusters is not known. For example, a comparison of the characteristics of multiple clusters may only reveal that the cluster are identical, even if such a comparison does not reveal that clusters are 'web server' clusters. Multiple clusters that have been identified as being identical (or sufficiently similar as measured by a threshold) will be referred to throughout the remainder of this document as a "cluster set," where the clusters that are members of the cluster set may be deployed across multiple customer's cloud deployments.

~~[00694]~~[00695] In some embodiments, information describing each member of the cluster set may be utilized to identify distributions across the cluster set. Distributions may be identified for traditional resource consumption metrics such as, for example, CPU usage, memory usage, network bandwidth usage, and others. A distribution may reveal, for example, that all members of the cluster set utilize between 10-60 Mb/s of network bandwidth, with the vast majority of members of the cluster set utilize between 45-60 Mb/s of network bandwidth. Readers will appreciate that distributions may also be identified for other quantifiable characteristics of each cluster. Such quantifiable characteristics can include, for example, the failure rate of a cluster or particular components thereof, an identification of communication protocols used by a cluster or particular components thereof, an identification of the types of communications endpoints that a cluster or particular components thereof communicate with (e.g., endpoints that reside on the public internet v. endpoints that are in a private network), characteristics that can be classified by a binary value (e.g., does any component in the cluster perform privileged operations),

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00105

information describing the various privileges that are given to a particular cluster, and many more.

~~[00695]~~[00696] In some embodiments, the distributions may be used to identify 'normal' behavior for a particular cluster. Consider an example in which the cluster set is identified, where each member of the cluster set represents a payroll system deployed in a particular customer's cloud deployment. In such an example, a distribution may be identified which indicates that all members of the cluster communicate with (and has privileged access to) the same set of cloud services, including: 1) a cloud database service (e.g., Amazon Aurora, Microsoft Azure SQL Database, Amazon Relational Database Service, Google Cloud SQL, Amazon DynamoDB), 2) a vendor provided SaaS offering that provides bill payment services, and 3) a vendor provided SaaS offering that provides accounting services. In such an example, by looking at each member of the cluster set and identifying that each member communicates with the same set of cloud services, a baseline may be established that identifies 'normal' behavior for each member of the cluster set, at least with respect to the specific characteristic (i.e., what cloud services are utilized by members) that the distribution is based on. As such, if monitoring a particular cluster revealed that some member of the cluster set accessed a source code repository cloud service (e.g., GitHub Enterprise on AWS), this sort of access would be outside of the typical distribution for this cluster set and could serve as the basis for raising an alert, denying access to the service, or initiating some other alerting/remediation workflow. Readers will appreciate that many distributions may be created for each cluster set, where each distribution is based on one or more characteristics of the members of the cluster set.

~~[00696]~~[00697] The example method depicted in Figure 5 also includes recommending 506, based on the normal behavior for one or more components 512 in one or more other cloud deployments 514, a change to the first cloud deployment 508. Recommending 506 a change to the first cloud deployment 508 may be carried out, for example, in response to determining that the normal behavior in one or more other cloud deployments 514 differs from the normal behavior for one or more components 510 in a first cloud deployment 508. In such an example, changes to the first cloud deployment 508 may be recommended that (if implemented) would cause the first cloud deployment 508 to be more similar to the other cloud deployments 514. For example, if the normal behavior in one or more other cloud deployments 514 indicates that all computing resources (e.g., virtual machines, container, serverless computing resources) communicate with each other using a particular secure data communications protocol and the normal behavior for one or more components 510 in a first cloud deployment 508 is for computing resources to communicate using some other data communications protocol, a change

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00106

may be recommended that involves reconfiguring the computing resources to communicate using the particular secure data communications protocol.

~~[00697]~~[00698] Readers will appreciate that in some embodiments the mere fact that normal behavior in a first cloud deployment 508 deviates from normal behavior in one or more other cloud deployments 514 may be sufficient rationale for recommending 506 a change to the first cloud deployment 508. That is, a departure from normality and standard practices alone may result in recommending 506 a change to the first cloud deployment 508. In other embodiments, recommending 506 a change to the first cloud deployment 508 may only occur where the normal behavior in one or more other cloud deployments 514 is determined to be superior to the normal behavior in the first cloud deployment 508. For example, recommending 506 a change to the first cloud deployment 508 may only be carried where the normal behavior in one or more other cloud deployments 514 is representative of a stronger security posture than the normal behavior in the first cloud deployment 508.

~~[00698]~~[00699] Consider an example in which a particular threat was detected (in part by detecting a deviation from normal behavior for one or more components 512) in a particular customer's cloud deployment 514, where the threat turned out to be a ransomware attack, which may in some embodiments include an encryption component and/or a data theft or leakage component. In such an example, if an identical (or sufficiently similar, following a general recognized pattern or 'fingerprint') threat is detected in the first customer's cloud deployment 508 (in part by detecting a similar deviation from deviation from normal behavior for one or more components 510), information describing the remedial actions (e.g., disabling encryption, increasing the frequency of backups, locking down a backup system, blocking transmission of data exernally, etc.) that were taken by the particular customer 514 may even recommended 506 as changes to be made to the first cloud deployment 508. Furthermore, if many customers had experienced the same attack and the data platform could determine with sufficient certainty that the first cloud deployment 508 was experiencing the same attack, workflows may be automatically initiated to carry out various remedial actions.

~~[00699]~~[00700] For further explanation, Fig. 6 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 6 is similar to the example depicted in Fig. 5, as the example depicted in Fig. 6 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00107

deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

~~[00700]~~[00701] In some embodiments, customers (and their corresponding deployments) may be modeled into logical groups such that cross customer learning could be carried out only across customers in the same logical group, or other customers in the same logical group may be given a greater weighting for the purposes of cross customer learning. For example, a logical group may be tied to a customer's business function (e.g., the customer is a financial company, the customer is a health care company, the customer is a services company, the customer has inventory to manage, the customer sells to retail customers, and so on). Likewise, a logical group may be tied to a customer's cloud deployment (e.g., the deployment uses a particular combination of resources, the deployment uses a particular set of cloud-services, the deployment utilizes availability zones and regions in a particular way, and so on). In such an example, a particular customer may be associated with multiple logical groups and cross customer learning for the particular customer may only involve other customers that are in the same (or sufficiently similar) logical groups. As such, the particular customer's resources (e.g., people, cloud deployment) may only be included in cluster sets with resources associated with other customers that are in the same (or sufficiently similar) logical groups.

~~[00701]~~[00702] To that end, the example method depicted in Fig. 6 also includes identifying 604, from the other cloud deployments, cloud deployments that are similar to the first cloud deployment 508. Identifying 604 cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by taking an inventory of the various components in each cloud deployment 508, 514 and identifying cloud deployments that have similar components, portions of the cloud deployment that have similar deployments, and so on. Alternatively, identifying 604 cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by identifying cloud deployments that are associated with customers in the same industry as the customer associated with the first cloud deployment 508, by identifying cloud deployments that are associated with a similar intended functionality as the first cloud deployment 508 (e.g., both cloud deployment are intended to provide an online store for an online retailer), and so on. In the examples in the preceding sentence, it may be assumed that two cloud deployments should be relatively similar even without inspecting the cloud deployments themselves. In such examples, similarity may be judge based on the application of rules, policies, heuristics, or similar mechanism. Alternatively, similarity may be detected via machine learning techniques as where information describing the activity of various components in many cloud deployments are fed as input to one or more machine learning models which subsequently identifies clusters across various cloud deployments that are similar.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00108

~~[00702]~~[00703] In the example depicted in Fig. 6, recommending 506 the change to the first cloud deployment 508 may be based on the normal behavior for the cloud deployments that are similar to the first cloud deployment. Recommending 606 the change to the first cloud deployment 508 based on the normal behavior for the cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by ignoring normal behavior for dissimilar cloud deployments and only taking into consideration those cloud deployments that have been identified 604 as being sufficiently similar to the first cloud deployment 508.

~~[00703]~~[00704] The example method depicted in Fig. 6 also includes identifying 602, from the other cloud deployments, one or more highly rated cloud deployments. A cloud deployment may be identified 602 as being 'highly rated' in the sense that fewer than a threshold number of vulnerabilities have been detected in the cloud deployment over some period of time, fewer than a threshold number of outages (or less than a threshold amount of downtime) has been detected in the cloud deployment over some period of time, fewer than a threshold number of regulatory violations (or attempted violations) have been detected in the cloud deployment over some period of time, or identified 602 based on some other standard. In fact, comparisons may be relative (rather than comparing some quantifiable aspect of the other deployments operation) such that, for example, the 15% of cloud deployments with the fewest detected vulnerabilities over some period of time as identified 602 as being 'highly rated'. In other embodiments, identifying 602 one or more highly rated cloud deployments may be carried out through the use of machine learning techniques where information describing the activity of various components in many cloud deployments are fed as input to one or more machine learning models which subsequently identifies deployments that have the best performance against a combination of one or more metrics (e.g., vulnerabilities, availability, cost). In such a way, the best-of-breed cloud deployments may be identified 602.

~~[00704]~~[00705] In the example depicted in Fig. 6, recommending 506 the change to the first cloud deployment 508 may be based on the normal behavior for the highly rated cloud deployments. Recommending 608 the change to the first cloud deployment 508 based on the normal behavior for the highly rated cloud deployments may be carried out, for example, by ignoring normal behavior for cloud deployments that are not highly rated and only taking into consideration those cloud deployments are highly rated.

~~[00705]~~[00706] Readers will appreciate that although the example in Fig. 6 is depicted where identifying 604 cloud deployments that are similar to the first cloud deployment 508 and identifying 602 one or more highly rated cloud deployments are alternatives to each other, in other embodiments both steps may be performed. In such embodiments, recommending 506 the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00109

change to the first cloud deployment 508 may be based on the normal behavior other cloud deployments that are both highly rated and similar to the first cloud deployment 508.

~~[00706]~~[00707] For further explanation, Fig. 7 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 7 is similar to the examples depicted in Fig. 5 and Fig. 6, as the example depicted in Fig. 7 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

~~[00707]~~[00708] The example method depicted in Fig. 7 also includes ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514. Ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514 may be carried out, for example, by identifying how well the cloud deployment 508 compares to other cloud deployments 514 with respect to one or more metrics. For example, cloud deployments may be ranked based on a weighted combination of multiple metrics (e.g., reliability, cost, regulatory compliance) such that the best-of-breed deployments may be identified. In such an example, the ranking of the first cloud deployment 508 relative to the other cloud deployments 514 may be presented to an administrator or other user associated with the first cloud deployment 508 for analysis by the administrator or other user.

~~[00708]~~[00709] The example method depicted in Fig. 7 also includes comparing 704 the trajectory of the first cloud deployment 508 to the trajectory of the other cloud deployments 514. The trajectory of a particular cloud deployment 508, 514 may be determined, for example, by evaluating the particular cloud deployment's 508, 514 performance over time as measured by one or more metrics. For example, the performance of each cloud deployment may be periodically scored based on a weighted combination of multiple metrics (e.g., reliability, cost, regulatory compliance). In such an example, the trajectory for a particular cloud deployment may be determined by determining the extent to which the score associated with a particular cloud deployment is changing over time. For example, if the first cloud deployment's 508 scoring is improving over time while the other cloud deployments' 514 scores are remaining the same over time, the trajectory of the first cloud deployment 508 may be determined to be better than the trajectory of the other cloud deployments 514 (in embodiments where a higher score is considered to be better).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00110

~~[00709]~~[00710] Readers will appreciate that ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514 and comparing 704 the trajectory of the first cloud deployment 508 to the trajectory of the other cloud deployments 514 are just two examples of the sort of analytics that can be performed by comparing the performance of multiple cloud deployments 508, 514. Readers will appreciate that other analytics may also be put in place due to the availability of information describing the performance of multiple cloud deployments 508, 514.

~~[00710]~~[00711] For further explanation, Fig. 8 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 8 is similar to the examples depicted in Figs. 5-7, as the example depicted in Fig. 8 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

~~[00711]~~[00712] The example method depicted in Fig. 8 also includes identifying 802 other cloud deployments 514 to exclude from consideration when recommending changes to the first cloud deployment 508. Readers will appreciate that other cloud deployments 514 may be excluded from consideration when recommending changes to the first cloud deployment 508 for a variety of reasons. For example, another cloud deployment 514 may be so dissimilar to the first cloud deployment 508 that there may be very little to learn that is relevant to the first cloud deployment 508, another cloud deployment 514 may be deficient for a variety of reasons such that there may be very little high quality takeaways from evaluating the other cloud deployment 514 that are relevant to the first cloud deployment 508, another cloud deployment 514 may be a relatively recent deployment such that any takeaways from evaluating the other cloud deployment 514 may not be reliable and proven over time, or for some other reason. In such an example, identifying 802 other cloud deployments 514 to exclude from consideration when recommending changes to the first cloud deployment 508 may be carried out by filtering the set of other cloud deployments 514 according to some criteria, or carried out in some other way. In the example depicted in Fig. 8, recommending 506 the change to the first cloud deployment 508 may therefore include recommending 806 changes based on the other cloud deployments 514 that are not excluded from consideration.

~~[00712]~~[00713] The example method depicted in Fig. 8 also includes identifying 804 a response to a recommended change in one or more other cloud deployments 514. Identifying 804 a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00111

response to a recommended change in one or more other cloud deployments 514 may be carried out, for example, by tracking the status of a recommended change that was recommended for another cloud deployment 514 to determine whether the recommended change was actually implemented. Consider an example in which a recommendation was made to have another cloud deployment 514 switch from using a first IaaS offering to a second IaaS offering in response to detecting some condition in the cloud deployment 514. In such an example, identifying 804 a response to a recommended change in one or more other cloud deployments 514 may be carried out by determining whether the recommended change was actually implemented and a switch was made to utilize the second IaaS offering as part of the cloud deployment 514. Readers will appreciate that by repeatedly identifying 804 responses to recommended changes in one or more other cloud deployments 514 can be used by the systems described above as an indication as to the systems described above are making useful recommendations. For example, if identifying 804 a response to a recommended change in one or more other cloud deployments 514 reveals that a very small percentage of administrators actually implement the recommended change, this may indicate that the recommend change is not particularly valuable and may be used as feedback in determining whether to make the recommended change if the condition that triggered the recommendation is encountered again. Alternatively, if identifying 804 a response to a recommended change in one or more other cloud deployments 514 reveals that a very large percentage of administrators actually implement the recommended change, this may indicate that the recommend change is valuable and may be used as feedback in determining whether to make the recommended change if the condition that triggered the recommendation is encountered again. In such a way, the response from other customer's may be used to drive recommendations that are made regarding the first cloud deployment 508.

~~[00713]~~[00714] In the example depicted in Fig. 8, recommending 506 the change to the first cloud deployment 508 recommending 808 the change to the first cloud deployment 508 based on the response to the recommended change in one or more other cloud deployments 514. Recommending 808 the change to the first cloud deployment 508 based on the response to the recommended change in one or more other cloud deployments 514 may be carried out, for example, only if a predetermined percentage of previous recipients of similar recommendations have actually implemented the recommended change, as described above.

~~[00714]~~[00715] Readers will appreciate that while the examples described in the preceding two paragraphs relate to an example in which the response to a recommended change in one or more other cloud deployments 514 is expressed in terms of whether the recommended change was or was not implemented, in other embodiments the response to a recommended change in one or more other cloud deployments 514 may be measured in other ways. For example, a response to a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00112

recommended change in one or more other cloud deployments 514 may represent whether the recommended change did or did not resolve the condition that caused the recommendation to be generated in the first place. In other embodiments, the response to a recommended change in one or more other cloud deployments 514 may be measured or determined using different criteria.

~~[00715]~~[00716] Although some of the figures described above depict only a single other cloud deployment 514, a single other cloud deployment 514 is included for ease of illustration but in no way represents a limitation of the embodiments described herein. In fact, most embodiments will include multiple other cloud deployments 514 that may be monitored and learned from.

~~[00716]~~[00717] Readers will appreciate that while many of the embodiments described above relate to embodiments where cross-customer learnings are used to generate recommendations for changes to the first cloud deployment 508, in other embodiments the cross-customer learning may be used to initiate, based on the normal behavior for the one or more other cloud deployments, a change to the first cloud deployment. Initiating such changes may be carried out, for example, by initiating one or more remediation workflows that implement the changes. As was the case with recommendations, initiating a change to the first cloud environment may be based on the normal behavior for similar cloud deployments, based on the normal behavior for the highly rated cloud deployments, based on the cloud deployments to exclude from consideration when recommending changes to the first cloud deployment, and so on.

~~[00717]~~[00718] For further explanation, Fig. 9 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 9 may be carried out by the systems described above (also referred to as a 'data platform' above). As such, the systems described above may include computer program instructions executing on computer hardware, virtualized hardware, or some other execution environment (e.g., one or more containers, one or more serverless compute instances), where the computer program instructions carry out the steps described in Fig. 9 when the computer program instructions are executed.

~~[00718]~~[00719] The example depicted in Fig. 9 includes a plurality of cloud deployments, including a first cloud deployment 910 and one or more additional cloud deployments 912, which in this illustration includes three cloud deployments 914a, 914b, 914n, although more or fewer cloud deployments may be included in the set of additional cloud deployments 912 in other embodiments. Such cloud deployments may be embodied as described above and may include a variety of components such as, for example, software applications, storage resources, computing resources, networking resources, and other resources. In each cloud deployment, each of the resources may be delivered as services provided by a public cloud, private cloud, hybrid cloud, and so on.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00113

~~[00719]~~[00720] The example method depicted in Fig. 9 includes identifying 902, for at least a portion of a first cloud deployment 910, one or more additional cloud deployments 912 to utilize for cross-customer learning. Cross-customer learning, as the phrase is used here, can generally be described as the process of learning about the cloud deployments of different customers, different organizations, or some other entity for the benefit of tailoring another entity's cloud deployment. For example, the cloud deployments of Customer A, Customer B, Customer C, Customer D, and Customer F may be evaluated and the information gathered about their deployments may be leveraged to help shape the cloud deployment of Customer F. In such an example, lessons learned by other customers may be used to help guide a particular customer's design, deployment, and management of their cloud deployments, which may be particularly useful for organizations that are pivoting to the cloud from on-premises based deployments.

~~[00720]~~[00721] Identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may be carried out, for example, by identifying cloud deployments for other customers that are in similar industries as the customer associated with the first cloud deployment 910, by identifying highly rated (as described above) cloud deployments, by identifying cloud deployments (or portions thereof) that are similar to the first cloud deployment 910 (or some portion thereof), or in some other way. In some embodiments, identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may be carried out using machine learning techniques as one or more machine learning models may be used to identify cloud deployments that are similar to the first cloud deployment 910, to identify best-of-breed cloud deployments, and so on. In such a way, identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may result in a subset of curated, relevant, and/or exemplary cloud deployments being utilized for cross-customer learning rather than using the entire set of available cloud deployments being utilized for cross-customer learning (although some embodiments could utilize all cloud deployments for cross-customer learning).

~~[00721]~~[00722] The example method depicted in Fig. 9 also includes receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912. The one or more actions associated with the additional cloud deployments 912 may be embodied as actions taken by components within the additional cloud deployments 912. For example, a particular software service accessing a data store may be an example of an action that is associated with the additional cloud deployments 912, especially where the software service and the data store are components of a particular additional cloud deployment 914a. Likewise, a message, request, or other form of data communications that are exchanged between the additional cloud deployments 912 and actors that are external to the additional cloud deployments 912 may be an example of an action that is associated with the additional cloud

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00114

deployments 912. For example, a server that is external to any of the additional cloud deployments 912 may send requests to access some software service within a particular additional cloud deployment 914a.

~~[00722]~~[00723] In other embodiments, the one or more actions associated with the additional cloud deployments 912 may include one or more user interactions with the additional cloud deployments 912. Such user interactions can include a description of which resources are accessed and/or utilized by particular users, particular business units (e.g., the finance department of a business, the engineering department of a business), particular personas (e.g., a system administrator, a software developer, a human resources manager), and so on. The one or more actions associated with the additional cloud deployments 912 may also include, for example, one or more interactions involving external applications, external computing devices, or similar external actor and the additional cloud deployments 912. Such user interactions can include, for example, a description of which resources within the additional cloud deployments 912 are accessed and/or utilized by particular external application, what IP address is associated with an external entity that is communicating with the additional cloud deployments 912, a description of the specific requests that an external actor is issuing to the resources within the additional cloud deployments 912, and so on.

~~[00723]~~[00724] In the example method depicted in Fig. 9, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 may be carried out by receiving (directly or indirectly) such information 916 from the additional cloud deployments 912 themselves. In fact, the additional cloud deployments 912 may be queried for such information. In other embodiments, the information 916 describing one or more actions associated with the additional cloud deployments 912 may be retained by the systems described above as part of monitoring the additional cloud deployments 912. In such an example, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 may be carried out by querying a data repository (e.g., a data warehouse) that contains such information that was gathered while monitoring the additional cloud deployments 912. Readers will appreciate that information 916 may be received 904 describing other types of actions associated with the additional cloud deployments 912.

~~[00724]~~[00725] The example method depicted in Fig. 9 also includes receiving 906 information 918 describing configurations associated with the additional cloud deployments 912. The information 918 describing configurations associated with the additional cloud deployments 912 may be embodied, for example, as information describing how components within a cloud deployment are organized (including which other components they may communication with), information describing what permissions are granted to various users, information describing the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00115

manner in which internal (i.e., occurring entirely within the cloud deployment) data communications and/or external (i.e., occurring at least partially external to the cloud deployment) data communications are carried out, and so on. Such information 918 may be received 906 in a similar manner as described above with respect to step 904.

~~[00725]~~[00726] The example method depicted in Fig. 9 also includes identifying 908, based on the configurations and the one or more actions, one or more configurations to adopt for the first cloud deployment 910. Readers will appreciate that by identifying 908 one or more configurations to adopt for the first cloud deployment 910 in such a way, the configuration of the first cloud deployment 910 may be influenced by the monitoring of the additional cloud deployments 912. In such an example, once the configurations that the first cloud deployment 910 should adopt have been identified 908, the first cloud deployment 910 (or associated entities) may be reconfigured to implement such configurations.

~~[00726]~~[00727] Identifying 908 one or more configurations to adopt for the first cloud deployment 910 may be carried out, for example, by identifying configurations in the additional cloud deployments 912 that were effective in dealing with various threats, securing vulnerabilities, or otherwise contributed to a healthy cloud deployment. Some configurations associated with the additional cloud deployments 912 may have been put in place in response to some action. For example, a configuration to always backup data stored in a first storage service (e.g., AWS S3) to a second, distinct storage service (e.g., AWS Glacier) may have been put in place in response to some attack that was directed to a particular cloud deployment's 914a S3 buckets. In such an embodiment, if the first cloud deployment 910 also leverages S3 as its object store, it may be desirable for the first cloud deployment 910 to adopt a configuration setting that would result in its S3 buckets being backed up to Glacier so that the first cloud deployment 910 can be ready to survive a similar attack that was experienced by the particular cloud deployment 914a.

~~[00727]~~[00728] For further explanation, Fig. 10 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 10 is similar to the example depicted in Fig. 9, as the example depicted in Fig. 10 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

~~[00728]~~[00729] In the example method depicted in Fig. 10, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 can include

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00116

receiving 1002 information describing a security threat to one or more of the additional cloud deployments 912. The security threat may be embodied, for example, as a ransomware attack that was directed to one or more of the additional cloud deployments 912, as a denial of service attack that was directed to one or more of the additional cloud deployments 912, as an SQL injection attack that was directed to one or more of the additional cloud deployments 912, or as some other security threat. The information describing a security threat to one or more of the additional cloud deployments 912 can include, for example, information describing where an attack originated from, information describing the components within the cloud deployment that were targeted by an attack, information describing data access patterns that were associated with an attack, or some other information that would be useful in detecting subsequent security threats.

~~[00729]~~[00730] In the example method depicted in Fig. 10, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can include receiving 1004 information describing configuration settings used to combat the security threat. The configuration settings used to combat the security threat may be embodied, for example, as configuration settings that caused the additional cloud deployments 912 to blacklist certain IP addresses or network domains, as configuration settings that caused the additional cloud deployments 912 to close vulnerabilities that allowed an attack to succeed, or some other configuration that would be useful in detecting/preventing/mitigating a security threat.

~~[00730]~~[00731] In the example method depicted in Fig. 10, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 can also include receiving 1006 information describing a detected vulnerability associated with one or more of the additional cloud deployments 912. The detected vulnerability may be embodied, for example, as a vulnerability to some known security threat, as a vulnerability to some data loss event, as a data breach vulnerability, or as some other vulnerability. The information describing a detected vulnerability of one or more of the additional cloud deployments 912 can include, for example, information describing vulnerable components, information describing data communications protocols, endpoints, or other data communications components that expose vulnerabilities, or some other information that would be useful in detecting a vulnerability.

~~[00731]~~[00732] In the example method depicted in Fig. 10, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can also include receiving 1008 information describing configuration settings used to address the vulnerability. The configuration settings used to address the vulnerability may be embodied, for example, as configuration settings that caused the additional cloud deployments 912 to blacklist certain IP addresses or network domains, as configuration settings that caused certain components within

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00117

the additional cloud deployments 912 to not connect to external data communications networks, as configuration settings that caused certain components to be accessible only by using enhanced authentication protocols, or as some other configuration that would be useful in detecting, preventing, or otherwise mitigating a vulnerability.

~~[00732]~~[00733] For further explanation, Fig. 11 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 11 is similar to the examples depicted in Fig. 9 and Fig. 10, as the example depicted in Fig. 11 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

~~[00733]~~[00734] In the example method depicted in Fig. 11, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can include receiving 1102 information describing permissions for one or more users of the additional cloud deployments 912. The one or more users of the additional cloud deployments 912 may be embodied, for example, as individual users (e.g., Bob Smith), as a collection of user (e.g., Finance Group, Marketing Group), and so on. The one or more users of the additional cloud deployments 912 may also be embodied as different personas within an organization, where each persona is associated with a certain role. For example, there may be 'software developer' personas, 'database administrator' personas, 'customer support' personas, and many others. Receiving 1102 information describing permissions for one or more users of the additional cloud deployments 912 may be carried out, for example, by receiving (directly or indirectly) information from one or more of the additional cloud deployments 912 that described what resources a particular user/persona/group can access, what level of privileges they have with respect to a particular resource, and any other information describing the user/persona/group's privileges. In fact, such information may be retained in a data warehouse during monitoring of the additional cloud deployments 912 and received 1102 by querying such a data warehouse.

~~[00734]~~[00735] The example method depicted in Fig. 11 also includes determining 1104, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment 910 are over-permissioned. A particular user may be over-permissioned in the sense that the user has privileges or permissions beyond those that should be associated with the user. Consider the example of two distinct personas within an organization: 1) a software engineer, and 2) an accounts payable administrator. In such an example, the software engineer might have sufficient privileges to access a code repository, a dev/test environment, a code documentation tool, and so on. The

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00118

accounts payable administrator, however, might have sufficient privileges to access bill paying software, a financial ledger, and so. The software developer having access to bill paying software or the accounts payable administrator having access to a code repository may be examples of the users being over-permissioned, as the user has access to resources that they should not be able to access. Such situations can create vulnerabilities as an over-permissioned user (or a malicious actor using the over-permissioned user's credentials) may intentionally or unintentionally perform some harmful act that they would not be capable of performing if they weren't over-permissioned.

~~[00735]~~[00736] Determining 1104 that one or more users of the first cloud deployment 910 are over-permissioned may be carried out, for example, by examining the information describing permissions for one or more users of the additional cloud deployments 912 to determine whether similar users in the additional cloud deployments 912 have different/fewer privileges. As such, a user in a first cloud deployment 910 may be determined to be over-permissioned if other cloud deployments 912 are giving similar users fewer permissions than such a user has in the first cloud deployment 910.

~~[00736]~~[00737] In the example method depicted in Fig. 11, identifying 908 one or more configurations to adopt for the first cloud deployment 910 can include identifying 1106 a reduced privilege level to give to the one or more users. Identifying 1106 a reduced privilege level to give to the one or more users may be carried out, for example, by giving an over-permissioned user of the first cloud deployment 910 privileges that are similar to those held by similar users in the additional cloud deployments 912.

~~[00737]~~[00738] For further explanation, Fig. 12 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 12 is similar to the examples depicted in Figs. 9-11, as the example depicted in Fig. 12 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

~~[00738]~~[00739] The example method depicted in Fig. 12 also includes receiving 1202 information 1204 describing one or more deployment processes associated with the additional cloud deployments 912. The information 1204 describing one or more deployment processes associated with the additional cloud deployments 912 can include, for example, information describing how a particular component within an additional cloud deployment was created (e.g., who requested that the component be created, when was the request to create the component issued). In addition, in some embodiments such information 1204 can include information

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00119

describing the software development processes that were used to develop software that will be deployed in a particular cloud deployment (e.g., was a code repository used, who has access to the repository, who committed the last change to some code before deployment). In such a way, the information 1204 describing one or more deployment processes associated with the additional cloud deployments 912 can describe the processes that are being implemented in other cloud deployments. Such information 1204 may be used, for example, to determine whether the development and deployment processes associated with the first cloud deployment 910 are abnormal, to use as the basis for making recommended changes to the development and deployment processes associated with the first cloud deployment 910, to initiate remediation workflows to back out changes to the first cloud deployment 910 that deviated from normal activity or violated some policy, and so on.

~~[00739]~~[00740] The example method depicted in Fig. 12 also includes identifying 1206 abnormally configured components in the first cloud deployment 910. Abnormally configured components in the first cloud deployment 910 may be embodied as components that are configured in a way that deviates from typical configurations observed by monitoring the additional cloud deployments 912. Monitoring the additional cloud deployments 912 may reveal, for example, that a connection to a server that is contained in another cloud deployment 914a should always go through an authentication server that is contained in another cloud deployment 914a. In such an example, if an examination of the first cloud deployment 910 reveals that an authentication server has been bypassed by a connection to a server, this may be an indication of a breach or misconfiguration regardless of customer's network topology. In such an example, the component (i.e., the server) may be identified 1206 as being abnormally configured based on its deviation from typical configurations observed in the additional cloud deployments 912.

~~[00740]~~[00741] Although the examples described above relate to embodiments where one or more configurations to adopt for the first cloud deployment 910 are identified 908 using the information described above, in other embodiments such information may be used for other purposes. For example, information gained by observing additional cloud deployments 912 may be leveraged to identify best-of-breed deployments, similar deployments, and so on.

~~[00741]~~[00742] In some embodiments, the distributions that identify 'normal' behavior for a particular cluster set may be used for a variety of purposes. For example, anomaly detection may be performed by identify members of the cluster set that are operating outside of a particular typical distribution. Likewise, best practices may be identified using distributions and members of the cluster set that are not adhering to best practices may be identified if they are operating outside of a particular typical distribution. Vulnerabilities may also be identified using distributions, for example, by identifying members of the cluster set that have over-privileged

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00120

users or components that may be able access things that they should not be able to access. Readers will appreciate that that deviation from an established baseline of normal behavior, normal activity, normal configuration, or other form of normal operation may be indicative of many other things, all of which may be detected through the usage of the distributions described above paired with the monitoring of particular clusters.

~~[00742]~~[00743] In some embodiments, the distributions that identify 'normal' behavior for a particular cluster set may be used to not only detect threats, vulnerabilities, compromise, and things of that nature, but the distributions that identify 'normal' behavior for a particular cluster set may be used to optimize the operation or configuration of a particular cluster. For example, cloud deployments that utilize fewer resources may be examined and characteristics that are common across such the cloud deployments may be identified as being high efficiency characteristics. Likewise, cloud deployments that are the subject of fewer alerts may be examined and characteristics that are common across such the cloud deployments may be identified as being high efficiency characteristics.

~~[00743]~~[00744] In some embodiments, historical information may also be retained and utilized to show the trajectory of a particular one or more clusters. For example, historical information may be used to compare the current state of a particular cluster (as measured by one or more quantifiable characteristics associated with the cluster) with a previous state of the cluster such that trends and/or trajectories may be identified. Consider an example in which a particular characteristic associated with a cluster identifies how many users are accessing the cluster. Historical information associated with the cluster may be compared to current (or most recent) information associated with the cluster to determine, for example, that more or less users are accessing the cluster, that the number of users that are accessing the cluster is increasing or decreasing at a certain rate, and so on. In some situations, some changes may be acceptable (e.g., if the cluster represents an interface to an online store, and increase in the number of users accessing the online store may be perfectly acceptable given an expansion in the organization's customer base) whereas in other situations changes may be troubling (e.g., if the cluster represents a source code repository and more users are accessing the repository in spite of a contraction in the number of developers that are employed by an organization) and my require alerts, further investigation, or some other remediation workflow.

~~[00744]~~[00745] In some embodiments, historical information may also be retained and utilized to show the trajectory of a particular cluster relative to other members of a cluster set. For example, historical information may be used to compare the failure rate of particular cluster (as measured by one or more quantifiable characteristics associated with the cluster) over time with the failure rate over time of other clusters in the cluster set. Such a comparison may reveal, for example,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00121

that a particular cluster was previously failing at a rate that was in line with the failure rate of other members of the cluster set, but that the particular cluster has more recently been failing more/less than other members of the cluster set. As such, through the use of such techniques a determination may be made as to whether a particular cluster is becoming more/less healthy than other members of the cluster set, more/less secure than other members of the cluster set, more/less efficient than other members of the cluster set, more/less reliable than other members of the cluster set, more/less compliant with relevant regulations than other members of the cluster set, slower/faster than other members of the cluster set, and so on. Such an analysis may be part of identifying best-of-breed deployments, best practices, providing remediation actions, providing recommendations, or utilized for a variety of other purposes.

~~[00745]~~[00746] Although the examples described above relate to embodiments where components within a cloud deployment are analyzed, monitored, or otherwise observed, in other embodiments the techniques described herein may be applied to other entities. For example, the techniques described above may be applied to analyze, monitor, or otherwise observe different personas with an organization, different users with an organization, different user groups with an organization, and so on. Using such techniques, 'normal' behaviors for a particular persona (e.g., a database administrator, a network administrator) can be identified, 'normal' behaviors for a particular user group (e.g., users that are part of an organization's finance department, users that are part of an organization's engineering department) can be identified, and so on.

~~[00746]~~[00747] In some embodiments, by identifying 'normal' behaviors (i.e., those behaviors that are consistent with standard distributions for particular characteristics associated with an entity) with different personas, different users, different user groups, or other entities, abnormal behavior may also be identified. Consider an example in which a group of users is identified as a cluster by virtue of those users accessing the same set of applications (e.g., all of the users access a set of applications that are finance-related applications). Further assume that similar clusters are identified for other customers, such that a cluster set may be formed. In such an example, a distribution may be identified for the set of applications that members of the cluster typically access. For example, each member of the cluster set may typically access an accounting application, a spreadsheet application, a payroll application, and so on. If one member of the cluster set consists of one or more users that also access an organization's source code repository, such behavior may be determined to be outside of the typical distribution for members of the cluster set. In response to making some determination, alerts may be generated, access to the source code repository may be blocked for the users in the cluster, a remediation workflow may be initiated, or some other action may be taken.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00122

~~[00747]~~[00748] In some embodiments, the techniques described above may be particularly useful for identifying over-privileged users, user groups, personas, or other entity. Identifying over-privileged users, user groups, personas may be carried out, at least in part, by identifying what set of privileges is 'normal' for a particular entity to have based on evaluating what privileges are given to similar or identical members of a cluster set. That is, an evaluation may be made as to what privileges are given to users, user groups, personas, or other entity by one or more other customers. If the set of privileges given to a particular customer's users, user groups, personas, or other entity are not consistent with (e.g., the set of privileges are much greater than) the privileges given to similar or identical users, user groups, personas, or other entity of another customer (i.e., the other members of the cluster set), a determination may be made that the particular customer's users, user groups, personas, or other entities are over-privileged. In some embodiments, an evaluation as to whether a user, user group, persona, or other entity is over-privileged may also include identifying privileges that the user has and comparing that with the privileges that the user actually utilizes. If some privileges are never used, this may be taken as an indication that the user is over-privileged.

~~[00748]~~[00749] In some embodiments, the cross customer techniques described above may be used to provide additional context to issues identified in a particular customer's cloud deployment, to provide recommendations to a particular customer, or even to drive remediation actions. Consider an example in which a particular threat was detected in a first customer's cloud deployment, where the threat turned out to be a ransomware attack, which may in some embodiments include an encryption component and/or a data theft or leakage component. In such an example, if an identical (or sufficiently similar, following a general recognized pattern or 'fingerprint') threat is detected in a second customer's cloud deployment, additional context may be provided by including information in an alert that is delivered to the second customer that indicates that the threat matches the profile of a ransomware attack that was detected in the first customer's cloud deployment. In fact, information describing the remedial actions (e.g., disabling encryption, increasing the frequency of backups, locking down a backup system, blocking transmission of data externally, etc.) that were taken by the first customer may even be included in the alert to the second customer or otherwise recommended to the second customer. Furthermore, if many customers had experienced the same attack and the data platform could determine with sufficient certainty that the second customer's cloud deployment was experiencing the same attack, workflows may be automatically initiated to carry out various remedial actions.

~~[00749]~~[00750] In some embodiments, the way other customers investigated or responded to a particular alert may also be used when presenting alerts to a particular customer. Consider an

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00123

example in which a particular threat was detected in ten different customer's cloud deployments. In such an example, assume that 9 of the 10 customers ignored the alert. In such an example, the conditions that are indicative of the particular threat are detected in a particular customer's cloud deployment, when raising the alert to the particular customer, information may be included in the alert indicating that most other customers ignored the alert. In such an embodiment, it is the behavior of users of the data platform – not components in a cloud deployment – that is being monitored and evaluated for the benefit of other customers. In other embodiments, information indicating that customers do or do not ignore a particular alert may be used when scoring or ranking the alert. For example, alerts that are ignored by most customers may be ranked as less critical than alerts that are acted upon by most customers. In addition, the particular security stance of a particular customer may be utilized when determining the extent to which their usage of the data platform should be utilized when guiding other customers. If a first customer has relatively poor security practices, for example, the first customer's usage of the data platform may be ignored (or given less weight) for the purposes of guiding other customers. If a second customer has relatively good security practices, however, the second customer's usage of the data platform may be taken into consideration (or given more weight) for the purposes of guiding other customers. Likewise, if users that ignored the alert experienced a security breach while users that did not ignore the alert did not experience a security breach, such an outcome could be taken into consideration when ranking the alert, determining what information to include in the alert, and so on.

~~[00750]~~[00751] In addition to using other customers interactions with the data platform to score, rank, suppress, or provide context for the alerts provided to other customers, the remediation actions taken by other customers may also be used for guiding a particular customer. Consider an example in which a particular threat was detected in ten different customer's cloud deployments. In such an example, assume that 9 of the 10 customers investigated the threat by accessing a particular customer support page. In this example, if the conditions that are indicative of the threat are detected in a particular customer's cloud deployment, information may be included in an alert indicating that most other customers that received the alert investigated the alert by accessing the particular customer support page. Likewise, if other customers successfully resolved the issue, information may be included in an alert indicating the solution that was implemented by other customers.

~~[00751]~~[00752] In some embodiments, the way customers investigated or responded to a particular alert may be used with other information to determine whether the customer's responses were actually correct. Consider an example in which an alert is sent to 10 customers identifying a vulnerability that would allow crypto miners to use the customer's resources for

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00124

solving complex computational problems for the purposes of acquiring cryptocurrency. In such an example, assume that 9 of 10 customers ignored the alert. Further assume that in this example, the 9 customers that ignored the alert subsequently had their cloud resources utilized by hackers for crypto mining, whereas the 1 customer that did not ignore the alert did not suffer such an attack. In such an example, the data platform described above should not cease issuing alerts when detecting this vulnerability by virtue of 9 of 10 customers ignoring the alert. Instead, by taking the ultimate outcome for each customer into consideration, rather than suppressing the alert, the data platforms should respond by taking actions that would make it more likely that customers would not ignore these alerts. For example, the severity level of the alert may be raised, a user of the data platform may be required to confirm receipt of the alert, contextual information could be included in the alert indicating that recipients that do not respond to the alert end up having their systems hijacked by crypto miners, or some other action may be taken.

~~[00752]~~[00753] In some embodiments, a customer's interactions may be analyzed to improve how alerts are delivered to a customer. For example, if an evaluation of customer interactions with the data platform indicates that most customer ignore alerts issued between 11:00 PM – 7:00 AM whereas most customers take action in response to alerts issued between 7:01 AM – 10:59 PM, then alerts that are issued between 11:00 PM – 7:00 AM may be reissued between 7:01 AM – 10:59 PM. Likewise, if a particular class of alerts (e.g., those related to vulnerability threats) are largely acted upon whereas another class of alerts (e.g., those related to a lack of compliance with regulatory requirements) are largely ignored, the manner in which alerts are generated may be altered. For example, alerts that are related to a lack of compliance with regulatory requirements may be issued to additional users, such as some user that is designated as having a compliance persona within a customer's organization. In other embodiments, other aspects of the customer's interactions with the data platform may be used to improve the manner in which the data platform interacts with customers (e.g., what type of devices received alerts, were alerts delivered in a primary or secondary window, do the parties that the alerts were sent to have other obligations at the time of alert as determined from an inspection of their calendar, and so on). While the example described above related to embodiments where the interactions of multiple customers with the data platform is analyzed, in other embodiments a single customer's interaction with the date platform may serve as the basis for altering the way that the data platform interacts with the customer.

~~[00753]~~[00754] In some embodiments, the same or additional cross customer learning techniques may be applied to earlier stages of a software development pipeline and even before an actual cloud deployment is in place for a particular customer. Stated differently, the same or additional cross customer learning techniques may be applied to things other than a deployed system. For

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00125

example, cross customer learning may be applied to development processes, testing processes, deployment processes, and so on

~~[00754]~~[00755] Readers will appreciate that developing, testing, and deploying software in cloud environment comes with a few requirements that were not always present in software development. In particular, software is always expected to be running but the software is also expected to continue to be updated with fixes, new features, or other improvements. As such, it is not desirable to cease running a software application, install a new version, and begin running the new version. As a result of these changes to the software development paradigm, the software development processes, software testing and validation processes, and deployment processes tend to be fairly independent of each other. In some embodiments, each of these processes may be analyzed, monitored, or otherwise observed by the data platforms described herein. Because each of these processes are analyzed, monitored, or otherwise observed by the data platforms described herein, an opportunity exists to leverage cross customer learning to these processes.

~~[00755]~~[00756] In some embodiments, the development processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. For example, interactions with a code repository may be monitored, permissions granted to each user of the code repository may be monitored, the number of people that are checking in and checking out code may be monitored, the extent to which code revisions are documented may be monitored, and so on. In such an example, the development processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

~~[00756]~~[00757] In some embodiments, the testing processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. For example, the number of tests that are run may monitored, the type of tests that are run may be monitored, the number of people that are running tests and evaluating test results may be monitored, the processes through which the results of testing are communicated with developers may be monitored, and so on. In such an example, the testing processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00126

shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

~~[00757]~~[00758] In some embodiments, the results of testing may be carried forward and utilized when an application is deployed. Consider an example in which testing reveals that the code performs some action that appears to create a possible vulnerability, but further testing and validation reveals that a vulnerability is not created. When this piece of code is subsequently deployed, it presumably will perform the same action (i.e., the action that was flagged during testing) that appears to create a possible vulnerability. Rather than raising an alert, initiating some investigative action, initiating a remedial action, or responding in a similar way, by carrying forward the knowledge gained during testing (i.e., that a vulnerability was not, in fact, created) the possible vulnerability may be ignored based on the conclusions reached during testing.

~~[00758]~~[00759] In some embodiments, the deployment processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. In such an example, the deployment processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

~~[00759]~~[00760] In some embodiments, some forms of static analysis may be used (in conjunction with other features of the data platform) to detect anomalies, vulnerabilities, threats, misconfigurations, violations of regulatory requirements, and many other things. Consider an example in which a cloud deployment is deployed using IaC. In such an example, one or more configuration files may be examined, and the state of the cloud deployment may be monitored to identify situations in which the state of the cloud deployment drifts from the configuration of the cloud deployment that was described in the configuration file. As such, the intended configuration of the cloud deployment (at least as expressed in one or more configuration files) may be used as a baseline to measure the current cloud deployment, such that alerts may be issued or other remediation workflows may be initiated when a customer's cloud deployment deviates from its codified state.

~~[00760]~~[00761] In some embodiments, other forms of code other than an IaC configuration may be examined to detect anomalies, vulnerabilities, threats, misconfigurations, violations of regulatory requirements, and many other things. For example, the source code that has been deployed in a customer's environment may be examined to determine all the things that the code could do. This information may be compared to a polygraph for the customer's cloud

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00127

deployment, which identifies all things that a customer's cloud deployment does do, as learned by monitoring and observing the customer's cloud deployment. Consider an example in which a polygraph for a customer's cloud deployment indicates that a first microservice in their cloud deployment only communicates with other internal microservices. In such an example, however, assume that the source code for the first microservice includes a messaging library that it uses to communicate with the other internal microservices. Further assume in this example, however, that examining the source code for the first microservice reveals that the messaging library also includes functions that enable a user of the library to send messages to recipients on an external network using standard internet protocols (e.g., TCP/IP, HTTPS, and so on). In such an example, although the polygraph for the customer's cloud deployment reveals that the first microservice only communicates with other internal microservices, the presence of functions that could be used for communications with external services, machines, and other entities may be undesirable. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if executed, would rise to the level of anomalous or otherwise unusual activity. For example, the data platform may prompt the customer to delete or disable the library functions described above that enable undesirable data communications.

~~[00761]~~[00762] In another example, assume that some source code has all the necessary capabilities (e.g. hashing functions, mathematical calculations, etc.) to mine cryptocurrency, as the mining process may only require performing standard computations. During a crypto miner attack, however, the source code may be subverted such that the code runs in a loop to perform the standard computations required for mining cryptocurrency. An evaluation as to what functions the source code can perform may therefore not reveal anything concerning but monitoring the actual operation of the source code may reveal that the source code is operating in an undesirable way. Likewise, static analysis may reveal things like a non-incrementing counter or other mechanism that would result in an infinite loop or similar operation. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if or when executed, rise to the level of anomalous or otherwise unusual activity.

~~[00762]~~[00763] In another example, assume that some source code has all the necessary capabilities to carry out the steps required for a ransomware attack, as the ransomware attack may only require performing standard operations like reading data, encrypting data, sending data, communicating externally, and so on. During a ransomware attack, however, the source code may be subverted such that the code does far more encryption than would be expected in a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00128

typical code module. An evaluation as to what functions the source code can perform may therefore not reveal anything concerning but monitoring the actual operation of the source code may reveal that the source code is operating in an undesirable way. Likewise, static analysis may reveal that all paths through the source result in data being encrypted, or that all paths through the source code results in some atypical pattern (e.g., equal amounts of reading data, encrypting data, and writing data, or unusual external data flows), which may rise to the level of being unusual or concerning behavior. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if or when executed, rise to the level of anomalous or otherwise unusual activity.

~~[00763]~~[00764] In some embodiments, the data platforms described above may be used to identify discriminates between two entities that may otherwise appear to be similar or identical. In the examples described herein, 'discriminates' may be embodied as characteristics of entities that, when not similar or identical, prevent the entities from being similar or identical. For example, many customers may have many Java processes. If the data platform were to identify the set of things that each Java process does for each customer and identify that set of things as the 'normal' or acceptable set of things that any Java process was allowed to do, this set would be far too large. As such, the data platform may be configured to look at things like command line arguments and know that one or more Java processes with one set of jar files and command line arguments is actually a separate program from one or more Java processes with another set of jar files and command line arguments. In other words, the data platform may identify discriminates (in the example, a first discriminate being distinct command line arguments and a second discriminate may be that the jar files for each process are distinct) between two sets of Java processes to determine that the Java processes are not actually similar or identical entities. As such, any attempt to engage in cross customer learning based on a first set of Java processes in a first customer's environment and a second set of Java processes in a second customer's environment may result in undesirable outcomes. Through the use of discriminates, however, a decision can be reached that these two sets of Java processes are not related and attempts to learn through an examination of these distinct entities would be undesirable.

~~[00764]~~[00765] In some embodiments, the usage of discriminates described above may be extended to other entities. For example, if a first user in a first customer's environment is a database administrator and a second user in a second customer's environment is also a database administrator, these users may initially be determined to be similar enough such that cross customer learning can take place. In such an example, however, if an examination of their activities, privileges, or something else reveals that they are not actually occupying the same

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00129

roles, cross customer learning with respect to these two users may be disabled. In such embodiments, the data platform described above, and models leveraged by such a data platform may be used to identify discriminates that may be used to decouple multiple users, applications, microservices, devices, or other entities that would otherwise be candidates for cross customer learning.

~~[00765]~~[00766] In some embodiments, the data platforms may be configured to analyze, monitor, or otherwise observe environments other than cloud deployments. In fact, the data platforms described here could apply the principles and techniques described herein to any environment such as, for example, an on-premises environment, a hybrid cloud environment, or some special purpose environment. As one example of a special purpose environment, consider an example in which the data platforms described herein are used to analyze, monitor, or otherwise observe a container orchestration environment such as a Kubernetes cluster (which may be deployed on-premises, in a public cloud, or in some other way). In such an example, the data platform may be configured to ingest Kubernetes audit logs via one or more agents or in some other way. Through the ingestion and subsequent analysis of such audit logs, the data platform may model normal behaviors of a Kubernetes cluster, normal behavior of a cluster administrator, and so on. As described above, any deviations from such normal behaviors may result in an alert being generated or some other remediation workflow being initiated. For example, if the ingestion and subsequent analysis of audit logs revealed that workloads are deployed and deleted according to some pattern, a customer deploying or deleting a workload in a manner that is inconsistent with the identified pattern may result in an alert being generated or some other remediation workflow being initiated. In fact, by evaluating audit logs from multiple customer's Kubernetes clusters, cross customer learning can be carried out to help define normal behavior for a Kubernetes cluster, a Kubernetes administrator, or some other entity associated with a Kubernetes cluster. For example, evaluating audit logs from the Kubernetes deployment of multiple customers may reveal that one entity (presumed to be a Kubernetes administrator, or related group of such administrators) is generally responsible for creating and deleting nodes from the cluster whereas another entity (perhaps a developer) is responsible for deploying new versions of the code that is executing in a container that is supported by the cluster. In such an example, if a single entity (or related group of entities) is observed creating nodes and modifying the code that is executing on a node by deploying a new container, an alert may be generated or some other remediation workflow may be initiated as a consequence of observing this atypical pattern.

~~[00766]~~[00767] One or more embodiments may be described herein with the aid of method steps illustrating the performance of specified functions and relationships thereof. The boundaries and

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00130

sequence of these functional building blocks and method steps have been arbitrarily defined herein for convenience of description. Alternate boundaries and sequences can be defined so long as the specified functions and relationships are appropriately performed. Any such alternate boundaries or sequences are thus within the scope and spirit of the claims. Further, the boundaries of these functional building blocks have been arbitrarily defined for convenience of description. Alternate boundaries could be defined as long as the certain significant functions are appropriately performed. Similarly, flow diagram blocks may also have been arbitrarily defined herein to illustrate certain significant functionality.

~~[00767]~~[00768] To the extent used, the flow diagram block boundaries and sequence could have been defined otherwise and still perform the certain significant functionality. Such alternate definitions of both functional building blocks and flow diagram blocks and sequences are thus within the scope and spirit of the claims. One of average skill in the art will also recognize that the functional building blocks, and other illustrative blocks, modules and components herein, can be implemented as illustrated or by discrete components, application specific integrated circuits, processors executing appropriate software and the like or any combination thereof.

~~[00768]~~[00769] While particular combinations of various functions and features of the one or more embodiments are expressly described herein, other combinations of these features and functions are likewise possible. The present disclosure is not limited by the particular examples disclosed herein and expressly incorporates these other combinations.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00131

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/671,199	**07/21/2023 07:15:09 PM ET**	**L0041B1CIP2**

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1070	FILED BY	Kate Hill
PATENT CENTER #	62486427	FILING DATE	02/14/2022
CUSTOMER #	182115	FIRST NAMED INVENTOR	ÚLFAR ERLINGSSON
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Edward Lenart

Documents

TOTAL DOCUMENTS: 3

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
2023-07-21_L0041B1CIP2_SpecificationMarkup.pdf	118	Specification	570 KB
2023-07-21_L0041B1CIP2_SpecificationClean.pdf	117	Specification	685 KB
2023-07-21_L0041B1CIP2_312Amendment.pdf	6	Amendment after Notice of Allowance (Rule 312)	760 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
2023-07-21_L0041B1CIP2_SpecificationMarkup.pdf	AB96A274710509B34E4EDBC1C084A373ED883202BCE8B5C8CC84025D7AEE411682B7A7A0623EDA14CF0CAA095AB425CF17817239D2D1AF21F22EFD4E543C8AC2
2023-07-21_L0041B1CIP2_SpecificationClean.pdf	5D9CC4A2F76F0FDC83758DE63EDD037D3D5B6E6CA7E6647D1C58DDE83622BEE1CEBE277383F183B5B486127BD3B88989BA324FC11AFAD8D3B6008A2BD9852FE5
2023-07-21_L0041B1CIP2_312Amendment.pdf	E9F11CB8DCE97BA29B1DD7A29C200D2B275F4DE42093B18223B25AFB9060B62FFFE9B9B8FF11C4E6645C762085171CBC60B1419E1AB4B39F9CE0F55BCAB8219D

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This is a continuation in-part application for patent entitled to a filing date and claiming the benefit of earlier-filed U.S. Patent Application No. 17/504,311, filed October 18, 2021, herein incorporated by reference in its entirety, and claims priority from U.S. Provisional Patent Application No. 63/243,013, filed September 10, 2021.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The accompanying drawings illustrate various embodiments and are a part of the specification. The illustrated embodiments are merely examples and do not limit the scope of the disclosure. Throughout the drawings, identical or similar reference numbers designate identical or similar elements.

[0003] Fig. 1A shows an illustrative configuration in which a data platform is configured to perform various operations with respect to a cloud environment that includes a plurality of compute assets.

[0004] Fig. 1B shows an illustrative implementation of the configuration of Fig. 1A.

[0005] Fig. 1C illustrates an example computing device.

[0006] Fig. 1D illustrates an example of an environment in which activities that occur within datacenters are modeled.

[0007] Fig. 2A illustrates an example of a process, used by an agent, to collect and report information about a client.

[0008] Fig. 2B illustrates a 5-tuple of data collected by an agent, physically and logically.

[0009] Fig. 2C illustrates a portion of a polygraph.

[0010] Fig. 2D illustrates a portion of a polygraph.

[0011] Fig. 2E illustrates an example of a communication polygraph.

[0012] Fig. 2F illustrates an example of a polygraph.

[0013] Fig. 2G illustrates an example of a polygraph as rendered in an interface.

[0014] Fig. 2H illustrates an example of a portion of a polygraph as rendered in an interface.

[0015] Fig. 2I illustrates an example of a portion of a polygraph as rendered in an interface.

[0016] Fig. 2J illustrates an example of a portion of a polygraph as rendered in an interface.

[0017] Fig. 2K illustrates an example of a portion of a polygraph as rendered in an interface.

[0018] Fig. 2L illustrates an example of an insider behavior graph as rendered in an interface.

[0019] Fig. 2M illustrates an example of a privilege change graph as rendered in an interface.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00134

[0020] Fig. 2N illustrates an example of a user login graph as rendered in an interface.

[0021] Fig. 2O illustrates an example of a machine server graph as rendered in an interface.

[0022] Fig. 3A illustrates an example of a process for detecting anomalies in a network environment.

[0023] Fig. 3B depicts a set of example processes communicating with other processes.

[0024] Fig. 3C depicts a set of example processes communicating with other processes.

[0025] Fig. 3D depicts a set of example processes communicating with other processes.

[0026] Fig. 3E depicts two pairs of clusters.

[0027] Fig. 3F is a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection.

[0028] Fig. 3G is an alternate representation of actions occurring in Fig. 3F.

[0029] Fig. 3H illustrates an example of a process for performing extended user tracking.

[0030] Fig. 3I is a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection.

[0031] Fig. 3J illustrates an example of a process for performing extended user tracking.

[0032] Fig. 3K illustrates example records.

[0033] Fig. 3L illustrates example output from performing an ssh connection match.

[0034] Fig. 3M illustrates example records.

[0035] Fig. 3N illustrates example records.

[0036] Fig. 3O illustrates example records.

[0037] Fig. 3P illustrates example records.

[0038] Fig. 3Q illustrates an adjacency relationship between two login sessions.

[0039] Fig. 3R illustrates example records.

[0040] Fig. 3S illustrates an example of a process for detecting anomalies.

[0041] Fig. 4A illustrates a representation of an embodiment of an insider behavior graph.

[0042] Fig. 4B illustrates an embodiment of a portion of an insider behavior graph.

[0043] Fig. 4C illustrates an embodiment of a portion of an insider behavior graph.

[0044] Fig. 4D illustrates an embodiment of a portion of an insider behavior graph.

[0045] Fig. 4E illustrates a representation of an embodiment of a user login graph.

[0046] Fig. 4F illustrates an example of a privilege change graph.

[0047] Fig. 4G illustrates an example of a privilege change graph.

[0048] Fig. 4H illustrates an example of a user interacting with a portion of an interface.

[0049] Fig. 4I illustrates an example of a dossier for an event.

[0050] Fig. 4J illustrates an example of a dossier for a domain.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00135

[0051] Fig. 4K depicts an example of an Entity Join graph by FilterKey and FilterKey Group (implicit join).

[0052] Fig. 4L illustrates an example process for dynamically generating and executing a query.

[0053] Fig. 5 sets forth a flowchart illustrating an example method of improving developer efficiency and application quality in accordance with some embodiments.

[0054] Fig. 6 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0055] Fig. 7 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0056] Fig. 8 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0057] Fig. 9 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0058] Fig. 10 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0059] Fig. 11 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0060] Fig. 12 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

DETAILED DESCRIPTION

[0061] Various illustrative embodiments are described herein with reference to the accompanying drawings. It will, however, be evident that various modifications and changes may be made thereto, and additional embodiments may be implemented, without departing from the scope of the invention as set forth in the claims. For example, certain features of one embodiment described herein may be combined with or substituted for features of another embodiment described herein. The description and drawings are accordingly to be regarded in an illustrative rather than a restrictive sense.

[0062] Fig. 1A shows an illustrative configuration 10 in which a data platform 12 is configured to perform various operations with respect to a cloud environment 14 that includes a plurality of compute assets 16-1 through 16-N (collectively "compute assets 16). For example, data platform 12 may include data ingestion resources 18 configured to ingest data from cloud environment 14 into data platform 12, data processing resources 20 configured to perform data processing operations with respect to the data, user interface resources 22 configured to provide one or more

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00136

external users and/or compute resources (e.g., computing device 24) with access to an output of data processing resources 20. Each of these resources are described in detail herein.

[0063] Cloud environment 14 may include any suitable network-based computing environment as may serve a particular application. For example, cloud environment 14 may be implemented by one or more compute resources provided and/or otherwise managed by one or more cloud service providers, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, and/or any other cloud service provider configured to provide public and/or private access to network-based compute resources.

[0064] Compute assets 16 may include, but are not limited to, containers (e.g., container images, deployed and executing container instances, etc.), virtual machines, workloads, applications, processes, physical machines, compute nodes, clusters of compute nodes, software runtime environments (e.g., container runtime environments), and/or any other virtual and/or physical compute resource that may reside in and/or be executed by one or more computer resources in cloud environment 14. In some examples, one or more compute assets 16 may reside in one or more datacenters.

[0065] A compute asset 16 may be associated with (e.g., owned, deployed, or managed by) a particular entity, such as a customer or client of cloud environment 14 and/or data platform 12. Accordingly, for purposes of the discussion herein, cloud environment 14 may be used by one or more entities.

[0066] Data platform 12 may be configured to perform one or more data security monitoring and/or remediation services, compliance monitoring services, anomaly detection services, DevOps services, compute asset management services, and/or any other type of data analytics service as may serve a particular implementation. Data platform 12 may be managed or otherwise associated with any suitable data platform provider, such as a provider of any of the data analytics services described herein. The various resources included in data platform 12 may reside in the cloud and/or be located on-premises and be implemented by any suitable combination of physical and/or virtual compute resources, such as one or more computing devices, microservices, applications, etc.

[0067] Data ingestion resources 18 may be configured to ingest data from cloud environment 14 into data platform 12. This may be performed in various ways, some of which are described in detail herein. For example, as illustrated by arrow 26, data ingestion resources 18 may be configured to receive the data from one or more agents deployed within cloud environment 14, utilize an event streaming platform (e.g., Kafka) to obtain the data, and/or pull data (e.g., configuration data) from cloud environment 14. In some examples, data ingestion resources 18 may obtain the data using one or more agentless configurations.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00137

[0068] The data ingested by data ingestion resources 18 from cloud environment 14 may include any type of data as may serve a particular implementation. For example, the data may include data representative of configuration information associated with compute assets 16, information about one or more processes running on compute assets 16, network activity information, information about events (creation events, modification events, communication events, user-initiated events, etc.) that occur with respect to compute assets 16, etc. In some examples, the data may or may not include actual customer data processed or otherwise generated by compute assets 16.

[0069] As illustrated by arrow 28, data ingestion resources 18 may be configured to load the data ingested from cloud environment 14 into a data store 30. Data store 30 is illustrated in Fig. 1A as being separate from and communicatively coupled to data platform 12. However, in some alternative embodiments, data store 30 is included within data platform 12.

[0070] Data store 30 may be implemented by any suitable data warehouse, data lake, data mart, and/or other type of database structure as may serve a particular implementation. Such data stores may be proprietary or may be embodied as vendor provided products or services such as, for example, Snowflake, Google BigQuery, Druid, Amazon Redshift, IBM Db2, Dremio, Databricks Lakehouse Platform, Cloudera, Azure Synapse Analytics, and others.

[0071] Although the examples described herein largely relate to embodiments where data is collected from agents and ultimately stored in a data store such as those provided by Snowflake, in other embodiments data that is collected from agents and other sources may be stored in different ways. For example, data that is collected from agents and other sources may be stored in a data warehouse, data lake, data mart, and/or any other data store.

[0072] A data warehouse may be embodied as an analytic database (e.g., a relational database) that is created from two or more data sources. Such a data warehouse may be leveraged to store historical data, often on the scale of petabytes. Data warehouses may have compute and memory resources for running complicated queries and generating reports. Data warehouses may be the data sources for business intelligence ('BI') systems, machine learning applications, and/or other applications. By leveraging a data warehouse, data that has been copied into the data warehouse may be indexed for good analytic query performance, without affecting the write performance of a database (e.g., an Online Transaction Processing ('OLTP') database). Data warehouses also enable the joining data from multiple sources for analysis. For example, a sales OLTP application probably has no need to know about the weather at various sales locations, but sales predictions could take advantage of that data. By adding historical weather data to a data warehouse, it would be possible to factor it into models of historical sales data.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00138

[0073] Data lakes, which store files of data in their native format, may be considered as "schema on read" resources. As such, any application that reads data from the lake may impose its own types and relationships on the data. Data warehouses, on the other hand, are "schema on write," meaning that data types, indexes, and relationships are imposed on the data as it is stored in an enterprise data warehouse (EDW). "Schema on read" resources may be beneficial for data that may be used in several contexts and poses little risk of losing data. "Schema on write" resources may be beneficial for data that has a specific purpose, and good for data that must relate properly to data from other sources. Such data stores may include data that is encrypted using homomorphic encryption, data encrypted using privacy-preserving encryption, smart contracts, non-fungible tokens, decentralized finance, and other techniques.

[0074] Data marts may contain data oriented towards a specific business line whereas data warehouses contain enterprise-wide data. Data marts may be dependent on a data warehouse, independent of the data warehouse (e.g., drawn from an operational database or external source), or a hybrid of the two. In embodiments described herein, different types of data stores (including combinations thereof) may be leveraged.

[0075] Data processing resources 20 may be configured to perform various data processing operations with respect to data ingested by data ingestion resources 18, including data ingested and stored in data store 30. For example, data processing resources 20 may be configured to perform one or more data security monitoring and/or remediation operations, compliance monitoring operations, anomaly detection operations, DevOps operations, compute asset management operations, and/or any other type of data analytics operation as may serve a particular implementation. Various examples of operations performed by data processing resources 20 are described herein.

[0076] As illustrated by arrow 32, data processing resources 20 may be configured to access data in data store 30 to perform the various operations described herein. In some examples, this may include performing one or more queries with respect to the data stored in data store 30. Such queries may be generated using any suitable query language.

[0077] In some examples, the queries provided by data processing resources 20 may be configured to direct data store 30 to perform one or more data analytics operations with respect to the data stored within data store 30. These data analytics operations may be with respect to data specific to a particular entity (e.g., data residing in one or more silos within data store 30 that are associated with a particular customer) and/or data associated with multiple entities. For example, data processing resources 20 may be configured to analyze data associated with a first entity and use the results of the analysis to perform one or more operations with respect to a second entity.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00139

[0078] One or more operations performed by data processing resources 20 may be performed periodically according to a predetermined schedule. For example, one or more operations may be performed by processing resources 20 every hour or any other suitable time interval. Additionally or alternatively, one or more operations performed by data processing resources 20 may be performed in substantially real-time (or near real-time) as data is ingested into data platform 12. In this manner, the results of such operations (e.g., one or more detected anomalies in the data) may be provided to one or more external entities (e.g., computing device 24 and/or one or more users) in substantially real-time and/or in near real-time.

[0079] User interface resources 22 may be configured to perform one or more user interface operations, examples of which are described herein. For example, user interface resources 22 may be configured to present one or more results of the data processing performed by data processing resources 20 to one or more external entities (e.g., computing device 24 and/or one or more users), as illustrated by arrow 34. As illustrated by arrow 36, user interface resources 22 may access data in data store 30 to perform the one or more user interface operations

[0080] Fig. 1B illustrates an implementation of configuration 10 in which an agent 38 (e.g., agent 38-1 through agent 38-N) is installed on each of compute assets 16. As used herein, an agent may include a self-contained binary and/or other type of code or application that can be run on any appropriate platforms, including within containers and/or other virtual compute assets. Agents 38 may monitor the nodes on which they execute for a variety of different activities, including but not limited to, connection, process, user, machine, and file activities. In some examples, agents 38 can be executed in user space, and can use a variety of kernel modules (e.g., auditd, iptables, netfilter, pcap, etc.) to collect data. Agents can be implemented in any appropriate programming language, such as C or Golang, using applicable kernel APIs.

[0081] Agents 38 may be deployed in any suitable manner. For example, an agent 38 may be deployed as a containerized application or as part of a containerized application. As described herein, agents 38 may selectively report information to data platform 12 in varying amounts of detail and/or with variable frequency.

[0082] Also shown in Fig. 1B is a load balancer 40 configured to perform one or more load balancing operations with respect to data ingestion operations performed by data ingestion resources 18 and/or user interface operations performed by user interface resources 22. Load balancer 40 is shown to be included in data platform 12. However, load balancer 40 may alternatively be located external to data platform 12. Load balancer 40 may be implemented by any suitable microservice, application, and/or other computing resources. In some alternative examples, data platform 12 may not utilize a load balancer such as load balancer 40.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00140

[0083] Also shown in Fig. 1B is long term storage 42 with which data ingestion resources may interface, as illustrated by arrow 44. Long term storage 42 may be implemented by any suitable type of storage resources, such as cloud-based storage (e.g., AWS S3, etc.) and/or on-premises storage and may be used by data ingestion resources 18 as part of the data ingestion process. Examples of this are described herein. In some examples, data platform 12 may not utilize long term storage 42.

[0084] The embodiments described herein can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the principles described herein. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0085] In some examples, a non-transitory computer-readable medium storing computer-readable instructions may be provided in accordance with the principles described herein. The instructions, when executed by a processor of a computing device, may direct the processor and/or computing device to perform one or more operations, including one or more of the operations described herein. Such instructions may be stored and/or transmitted using any of a variety of known computer-readable media.

[0086] A non-transitory computer-readable medium as referred to herein may include any non-transitory storage medium that participates in providing data (e.g., instructions) that may be read and/or executed by a computing device (e.g., by a processor of a computing device). For example, a non-transitory computer-readable medium may include, but is not limited to, any combination of non-volatile storage media and/or volatile storage media. Exemplary non-volatile storage media include, but are not limited to, read-only memory, flash memory, a solid-state drive, a magnetic storage device (e.g. a hard disk, a floppy disk, magnetic tape, etc.), ferroelectric random-access memory ("RAM"), and an optical disc (e.g., a compact disc, a digital video disc, a Blu-ray disc, etc.). Exemplary volatile storage media include, but are not limited to, RAM (e.g., dynamic RAM).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00141

[0087] Fig. 1C illustrates an example computing device 50 that may be specifically configured to perform one or more of the processes described herein. Any of the systems, microservices, computing devices, and/or other components described herein may be implemented by computing device 50.

[0088] As shown in Fig. 1C, computing device 50 may include a communication interface 52, a processor 54, a storage device 56, and an input/output ("I/O") module 58 communicatively connected one to another via a communication infrastructure 60. While an exemplary computing device 50 is shown in Fig. 1C, the components illustrated in Fig. 1C are not intended to be limiting. Additional or alternative components may be used in other embodiments. Components of computing device 50 shown in Fig. 1C will now be described in additional detail.

[0089] Communication interface 52 may be configured to communicate with one or more computing devices. Examples of communication interface 52 include, without limitation, a wired network interface (such as a network interface card), a wireless network interface (such as a wireless network interface card), a modem, an audio/video connection, and any other suitable interface.

[0090] Processor 54 generally represents any type or form of processing unit capable of processing data and/or interpreting, executing, and/or directing execution of one or more of the instructions, processes, and/or operations described herein. Processor 54 may perform operations by executing computer-executable instructions 62 (e.g., an application, software, code, and/or other executable data instance) stored in storage device 56.

[0091] Storage device 56 may include one or more data storage media, devices, or configurations and may employ any type, form, and combination of data storage media and/or device. For example, storage device 56 may include, but is not limited to, any combination of the non-volatile media and/or volatile media described herein. Electronic data, including data described herein, may be temporarily and/or permanently stored in storage device 56. For example, data representative of computer-executable instructions 62 configured to direct processor 54 to perform any of the operations described herein may be stored within storage device 56. In some examples, data may be arranged in one or more databases residing within storage device 56.

[0092] I/O module 58 may include one or more I/O modules configured to receive user input and provide user output. I/O module 58 may include any hardware, firmware, software, or combination thereof supportive of input and output capabilities. For example, I/O module 58 may include hardware and/or software for capturing user input, including, but not limited to, a keyboard or keypad, a touchscreen component (e.g., touchscreen display), a receiver (e.g., an RF or infrared receiver), motion sensors, and/or one or more input buttons.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00142

[0093] I/O module 58 may include one or more devices for presenting output to a user, including, but not limited to, a graphics engine, a display (e.g., a display screen), one or more output drivers (e.g., display drivers), one or more audio speakers, and one or more audio drivers. In certain embodiments, I/O module 58 is configured to provide graphical data to a display for presentation to a user. The graphical data may be representative of one or more graphical user interfaces and/or any other graphical content as may serve a particular implementation.

[0094] Fig. 1D illustrates an example implementation 100 of configuration 10. As such, one more components shown in Fig. 1D may implement one or more components shown in Fig. 1A and/or Fig. 1B. In particular, implementation 100 illustrates an environment in which activities that occur within datacenters are modeled using data platform 12. Using techniques described herein, a baseline of datacenter activity can be modeled, and deviations from that baseline can be identified as anomalous. Anomaly detection can be beneficial in a security context, a compliance context, an asset management context, a DevOps context, and/or any other data analytics context as may serve a particular implementation.

[0095] Two example datacenters (104 and 106) are shown in Fig. 1D, and are associated with (e.g., belong to) entities named entity A and entity B, respectively. A datacenter may include dedicated equipment (e.g., owned and operated by entity A, or owned/leased by entity A and operated exclusively on entity A's behalf by a third party). A datacenter can also include cloud-based resources, such as infrastructure as a service (IaaS), platform as a service (PaaS), and/or software as a service (SaaS) elements. The techniques described herein can be used in conjunction with multiple types of datacenters, including ones wholly using dedicated equipment, ones that are entirely cloud-based, and ones that use a mixture of both dedicated equipment and cloud-based resources.

[0096] Both datacenter 104 and datacenter 106 include a plurality of nodes, depicted collectively as set of nodes 108 and set of nodes 110, respectively, in Fig. 1D. These nodes may implement compute assets 16. Installed on each of the nodes are in-server / in-virtual machine (VM) / embedded in IoT device agents (e.g., agent 112), which are configured to collect data and report it to data platform 12 for analysis. As described herein, agents may be small, self-contained binaries that can be run on any appropriate platforms, including virtualized ones (and, as applicable, within containers). Agents may monitor the nodes on which they execute for a variety of different activities, including: connection, process, user, machine, and file activities. Agents can be executed in user space, and can use a variety of kernel modules (e.g., auditd, iptables, netfilter, pcap, etc.) to collect data. Agents can be implemented in any appropriate programming language, such as C or Golang, using applicable kernel APIs.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00143

[0097] As described herein, agents can selectively report information to data platform 12 in varying amounts of detail and/or with variable frequency. As is also described herein, the data collected by agents may be used by data platform 12 to create polygraphs, which are graphs of logical entities, connected by behaviors. In some embodiments, agents report information directly to data platform 12. In other embodiments, at least some agents provide information to a data aggregator, such as data aggregator 114, which in turn provides information to data platform 12. The functionality of a data aggregator can be implemented as a separate binary or other application (distinct from an agent binary), and can also be implemented by having an agent execute in an "aggregator mode" in which the designated aggregator node acts as a Layer 7 proxy for other agents that do not have access to data platform 12. Further, a chain of multiple aggregators can be used, if applicable (e.g., with agent 112 providing data to data aggregator 114, which in turn provides data to another aggregator (not pictured) which provides data to data platform 12). An example way to implement an aggregator is through a program written in an appropriate language, such as C or Golang.

[0098] Use of an aggregator can be beneficial in sensitive environments (e.g., involving financial or medical transactions) where various nodes are subject to regulatory or other architectural requirements (e.g., prohibiting a given node from communicating with systems outside of datacenter 104). Use of an aggregator can also help to minimize security exposure more generally. As one example, by limiting communications with data platform 12 to data aggregator 114, individual nodes in nodes 108 need not make external network connections (e.g., via Internet 124), which can potentially expose them to compromise (e.g., by other external devices, such as device 118, operated by a criminal). Similarly, data platform 12 can provide updates, configuration information, etc., to data aggregator 114 (which in turn distributes them to nodes 108), rather than requiring nodes 108 to allow incoming connections from data platform 12 directly.

[0099] Another benefit of an aggregator model is that network congestion can be reduced (e.g., with a single connection being made at any given time between data aggregator 114 and data platform 12, rather than potentially many different connections being open between various of nodes 108 and data platform 12). Similarly, network consumption can also be reduced (e.g., with the aggregator applying compression techniques/bundling data received from multiple agents).

[00100] One example way that an agent (e.g., agent 112, installed on node 116) can provide information to data aggregator 114 is via a REST API, formatted using data serialization protocols such as Apache Avro. One example type of information sent by agent 112 to data aggregator 114 is status information. Status information may be sent by an agent periodically (e.g., once an hour or once any other predetermined amount of time). Alternatively, status

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00144

information may be sent continuously or in response to occurrence of one or more events. The status information may include, but is not limited to, a. an amount of event backlog (in bytes) that has not yet been transmitted, b. configuration information, c. any data loss period for which data was dropped, d. a cumulative count of errors encountered since the agent started, e. version information for the agent binary, and/or f. cumulative statistics on data collection (e.g., number of network packets processed, new processes seen, etc.).

[00101] A second example type of information that may be sent by agent 112 to data aggregator 114 is event data (described in more detail herein), which may include a UTC timestamp for each event. As applicable, the agent can control the amount of data that it sends to the data aggregator in each call (e.g., a maximum of 10MB) by adjusting the amount of data sent to manage the conflicting goals of transmitting data as soon as possible, and maximizing throughput. Data can also be compressed or uncompressed by the agent (as applicable) prior to sending the data.

[00102] Each data aggregator may run within a particular customer environment. A data aggregator (e.g., data aggregator 114) may facilitate data routing from many different agents (e.g., agents executing on nodes 108) to data platform 12. In various embodiments, data aggregator 114 may implement a SOCKS 5 caching proxy through which agents can connect to data platform 12. As applicable, data aggregator 114 can encrypt (or otherwise obfuscate) sensitive information prior to transmitting it to data platform 12, and can also distribute key material to agents which can encrypt the information (as applicable). Data aggregator 114 may include a local storage, to which agents can upload data (e.g., pcap packets). The storage may have a key-value interface. The local storage can also be omitted, and agents configured to upload data to a cloud storage or other storage area, as applicable. Data aggregator 114 can, in some embodiments, also cache locally and distribute software upgrades, patches, or configuration information (e.g., as received from data platform 12).

[00103] Various examples associated with agent data collection and reporting will now be described.

[00104] In the following example, suppose that a user (e.g., a network administrator) at entity A (hereinafter "user A") has decided to begin using the services of data platform 12. In some embodiments, user A may access a web frontend (e.g., web app 120) using a computer 126 and enrolls (on behalf of entity A) an account with data platform 12. After enrollment is complete, user A may be presented with a set of installers, pre-built and customized for the environment of entity A, that user A can download from data platform 12 and deploy on nodes 108. Examples of such installers include, but are not limited to, a Windows executable file, an iOS app, a Linux package (e.g., .deb or .rpm), a binary, or a container (e.g., a Docker container). When a user (e.g., a network administrator) at entity B (hereinafter "user B") also signs up for the services of data

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00145

platform 12, user B may be similarly presented with a set of installers that are pre-built and customized for the environment of entity B.

[00105] User A deploys an appropriate installer on each of nodes 108 (e.g., with a Windows executable file deployed on a Windows-based platform or a Linux package deployed on a Linux platform, as applicable). As applicable, the agent can be deployed in a container. Agent deployment can also be performed using one or more appropriate automation tools, such as Chef, Puppet, Salt, and Ansible. Deployment can also be performed using managed/hosted container management/orchestration frameworks such as Kubernetes, Mesos, and/or Docker Swarm.

[00106] In various embodiments, the agent may be installed in the user space (i.e., is not a kernel module), and the same binary is executed on each node of the same type (e.g., all Windows-based platforms have the same Windows-based binary installed on them). An illustrative function of an agent, such as agent 112, is to collect data (e.g., associated with node 116) and report it (e.g., to data aggregator 114). Other tasks that can be performed by agents include data configuration and upgrading.

[00107] One approach to collecting data as described herein is to collect virtually all information available about a node (and, e.g., the processes running on it). Alternatively, the agent may monitor for network connections, and then begin collecting information about processes associated with the network connections, using the presence of a network packet associated with a process as a trigger for collecting additional information about the process. As an example, if a user of node 116 executes an application, such as a calculator application, which does not typically interact with the network, no information about use of that application may be collected by agent 112 and/or sent to data aggregator 114. If, however, the user of node 116 executes an ssh command (e.g., to ssh from node 116 to node 122), agent 112 may collect information about the process and provide associated information to data aggregator 114. In various embodiments, the agent may always collect/report information about certain events, such as privilege escalation, irrespective of whether the event is associated with network activity.

[00108] An approach to collecting information (e.g., by an agent) is as follows, and described in conjunction with process 200 depicted in Fig. 2A. An agent (e.g., agent 112) monitors its node (e.g., node 116) for network activity. One example way that agent 112 can monitor node 116 for network activity is by using a network packet capture tool (e.g., listening using libpcap). As packets are received (201), the agent obtains and maintains (e.g., in an in-memory cache) connection information associated with the network activity (202). Examples of such information include DNS query/response, TCP, UDP, and IP information.

[00109] The agent may also determine a process associated with the network connection (203). One example approach is for the agent to use a kernel network diagnostic API (e.g., netlink_diag)

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00146

to obtain inode/process information from the kernel. Another example approach is for the agent to scan using netstat (e.g., on /proc/net/tcp, /proc/net/tcp6, /proc/net/udp, and /proc/net/udp6) to obtain sockets and relate them to processes. Information such as socket state (e.g., whether a socket is connected, listening, etc.) can also be collected by the agent.

[00110] One way an agent can obtain a mapping between a given inode and a process identifier is to scan within the /proc/pid directory. For each of the processes currently running, the agent examines each of their file descriptors. If a file descriptor is a match for the inode, the agent can determine that the process associated with the file descriptor owns the inode. Once a mapping is determined between an inode and a process identifier, the mapping is cached. As additional packets are received for the connection, the cached process information is used (rather than a new search being performed).

[00111] In some cases, exhaustively scanning for an inode match across every file descriptor may not be feasible (e.g., due to CPU limitations). In various embodiments, searching through file descriptors is accordingly optimized. User filtering is one example of such an optimization. A given socket is owned by a user. Any processes associated with the socket will be owned by the same user as the socket. When matching an inode (identified as relating to a given socket) against processes, the agent can filter through the processes and only examine the file descriptors of processes sharing the same user owner as the socket. In various embodiments, processes owned by root are always searched against (e.g., even when user filtering is employed).

[00112] Another example of an optimization is to prioritize searching the file descriptors of certain processes over others. One such prioritization is to search through the subdirectories of /proc/ starting with the youngest process. One approximation of such a sort order is to search through /proc/ in reverse order (e.g., examining highest numbered processes first). Higher numbered processes are more likely to be newer (i.e., not long-standing processes), and thus more likely to be associated with new connections (i.e., ones for which inode-process mappings are not already cached). In some cases, the most recently created process may not have the highest process identifier (e.g., due to the kernel wrapping through process identifiers).

[00113] Another example prioritization is to query the kernel for an identification of the most recently created process and to search in a backward order through the directories in /proc/ (e.g., starting at the most recently created process and working backwards, then wrapping to the highest value (e.g., 32768) and continuing to work backward from there). An alternate approach is for the agent to keep track of the newest process that it has reported information on (e.g., to data aggregator 114), and begin its search of /proc/ in a forward order starting from the PID of that process.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00147

[00114] Another example prioritization is to maintain, for each user actively using node 116, a list of the five (or any other number) most recently active processes. Those processes are more likely than other processes (less active, or passive) on node 116 to be involved with new connections, and can thus be searched first. For many processes, lower valued file descriptors tend to correspond to non-sockets (e.g., stdin, stdout, stderr). Yet another optimization is to preferentially search higher valued file descriptors (e.g., across processes) over lower valued file descriptors (that are less likely to yield matches).

[00115] In some cases, while attempting to locate a process identifier for a given inode, an agent may encounter a socket that does not correspond to the inode being matched against and is not already cached. The identity of that socket (and its corresponding inode) can be cached, once discovered, thus removing a future need to search for that pair.

[00116] In some cases, a connection may terminate before the agent is able to determine its associated process (e.g., due to a very short-lived connection, due to a backlog in agent processing, etc.). One approach to addressing such a situation is to asynchronously collect information about the connection using the audit kernel API, which streams information to user space. The information collected from the audit API (which can include PID/inode information) can be matched by the agent against pcap/inode information. In some embodiments, the audit API is always used, for all connections. However, due to CPU utilization considerations, use of the audit API can also be reserved for short/otherwise problematic connections (and/or omitted, as applicable).

[00117] Once the agent has determined which process is associated with the network connection (203), the agent can then collect additional information associated with the process (204). As will be described in more detail below, some of the collected information may include attributes of the process (e.g., a process parent hierarchy, and an identification of a binary associated with the process). As will also be described in more detail below, other of the collected information is derived (e.g., session summarization data and hash values).

[00118] The collected information is then transmitted (205), e.g., by an agent (e.g., agent 112) to a data aggregator (e.g., data aggregator 114), which in turn provides the information to data platform 12. In some embodiments, all information collected by an agent may be transmitted (e.g., to a data aggregator and/or to data platform 12). In other embodiments, the amount of data transmitted may be minimized (e.g., for efficiency reasons), using various techniques.

[00119] One approach to minimizing the amount of data flowing from agents (such as agents installed on nodes 108) to data platform 12 is to use a technique of implicit references with unique keys. The keys can be explicitly used by data platform 12 to extract/derive relationships, as necessary, in a data set at a later time, without impacting performance.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00148

[00120] As previously mentioned, some data collected about a process is constant and does not change over the lifetime of the process (e.g., attributes), and some data changes (e.g., statistical information and other variable information). Constant data can be transmitted (210) once, when the agent first becomes aware of the process. And, if any changes to the constant data are detected (e.g., a process changes its parent), a refreshed version of the data can be transmitted (210) as applicable.

[00121] In some examples, an agent may collect variable data (e.g., data that may change over the lifetime of the process). In some examples, variable data can be transmitted (210) at periodic (or other) intervals. Alternatively, variable data may be transmitted in substantially real time as it is collected. In some examples, the variable data may indicate a thread count for a process, a total virtual memory used by the process, the total resident memory used by the process, the total time spent by the process executing in user space, and/or the total time spent by the process executing in kernel space. In some examples, the data may include a hash that may be used within data platform 12 to join process creation time attributes with runtime attributes to construct a full dataset.

[00122] Below are additional examples of data that an agent, such as agent 112, can collect and provide to data platform 12.

[00123] 1. User Data

[00124] Core User Data: user name, UID (user ID), primary group, other groups, home directory.

[00125] Failed Login Data: IP address, hostname, username, count.

[00126] User Login Data: user name, hostname, IP address, start time, TTY (terminal), UID (user ID), GID (group ID), process, end time.

[00127] 2. Machine Data

[00128] Dropped Packet Data: source IP address, destination IP address, destination port, protocol, count.

[00129] Machine Data: hostname, domain name, architecture, kernel, kernel release, kernel version, OS, OS version, OS description, CPU, memory, model number, number of cores, last boot time, last boot reason, tags (e.g., Cloud provider tags such as AWS, GCP, or Azure tags), default router, interface name, interface hardware address, interface IP address and mask, promiscuous mode.

[00130] 3. Network Data

[00131] Network Connection Data: source IP address, destination IP address, source port, destination port, protocol, start time, end time, incoming and outgoing bytes, source process, destination process, direction of connection, histograms of packet length, inter packet delay, session lengths, etc.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00149

[00132] Listening Ports in Server: source IP address, port number, protocol, process.

[00133] Dropped Packet Data: source IP address, destination IP address, destination port, protocol, count.

[00134] Arp Data: source hardware address, source IP address, destination hardware address, destination IP address.

[00135] DNS Data: source IP address, response code, response string, question (request), packet length, final answer (response).

[00136] 4. Application Data

[00137] Package Data: exe path, package name, architecture, version, package path, checksums (MD5, SHA-1, SHA-256), size, owner, owner ID.

[00138] Application Data: command line, PID (process ID), start time, UID (user ID), EUID (effective UID), PPID (parent process ID), PGID (process group ID), SID (session ID), exe path, username, container ID.

[00139] 5. Container Data

[00140] Container Image Data: image creation time, parent ID, author, container type, repo, (AWS) tags, size, virtual size, image version.

[00141] Container Data: container start time, container type, container name, container ID, network mode, privileged, PID mode, IP addresses, listening ports, volume map, process ID.

[00142] 6. File Data

[00143] File path, file data hash, symbolic links, file creation data, file change data, file metadata, file mode.

[00144] As mentioned above, an agent, such as agent 112, can be deployed in a container (e.g., a Docker container), and can also be used to collect information about containers. Collection about a container can be performed by an agent irrespective of whether the agent is itself deployed in a container or not (as the agent can be deployed in a container running in a privileged mode that allows for monitoring).

[00145] Agents can discover containers (e.g., for monitoring) by listening for container create events (e.g., provided by Docker), and can also perform periodic ordered discovery scans to determine whether containers are running on a node. When a container is discovered, the agent can obtain attributes of the container, e.g., using standard Docker API calls (e.g., to obtain IP addresses associated with the container, whether there's a server running inside, what port it is listening on, associated PIDs, etc.). Information such as the parent process that started the container can also be collected, as can information about the image (which comes from the Docker repository).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00150

[00146] In various embodiments, agents may use namespaces to determine whether a process is associated with a container. Namespaces are a feature of the Linux kernel that can be used to isolate resources of a collection of processes. Examples of namespaces include process ID (PID) namespaces, network namespaces, and user namespaces. Given a process, the agent can perform a fast lookup to determine whether the process is part of the namespace the container claims to be its namespace.

[00147] As mentioned, agents can be configured to report certain types of information (e.g., attribute information) once, when the agent first becomes aware of a process. In various embodiments, such static information is not reported again (or is reported once a day, every twelve hours, etc.), unless it changes (e.g., a process changes its parent, changes its owner, or a SHA-1 of the binary associated with the process changes).

[00148] In contrast to static/attribute information, certain types of data change constantly (e.g., network-related data). In various embodiments, agents are configured to report a list of current connections every minute (or other appropriate time interval). In that connection list will be connections that started in that minute interval, connections that ended in that minute interval, and connections that were ongoing throughout the minute interval (e.g., a one minute slice of a one hour connection).

[00149] In various embodiments, agents are configured to collect/compute statistical information about connections (e.g., at the one minute level of granularity and or at any other time interval). Examples of such information include, for the time interval, the number of bytes transferred, and in which direction. Another example of information collected by an agent about a connection is the length of time between packets. For connections that span multiple time intervals (e.g., a seven minute connection), statistics may be calculated for each minute of the connection. Such statistical information (for all connections) can be reported (e.g., to a data aggregator) once a minute.

[00150] In various embodiments, agents are also configured to maintain histogram data for a given network connection, and provide the histogram data (e.g., in the Apache Avro data exchange format) under the Connection event type data. Examples of such histograms include: 1. a packet length histogram (packet_len_hist), which characterizes network packet distribution; 2. a session length histogram (session_len_hist), which characterizes a network session length; 3. a session time histogram (session_time_hist), which characterizes a network session time; and 4. a session switch time histogram (session_switch_time_hist), which characterizes network session switch time (i.e., incoming->outgoing and vice versa). For example, histogram data may include one or more of the following fields: 1. count, which provides a count of the elements in the sampling; 2. sum, which provides a sum of elements in the sampling; 3. max, which provides the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00151

highest value element in the sampling; 4. std_dev, which provides the standard deviation of elements in the sampling; and 5. buckets, which provides a discrete sample bucket distribution of sampling data (if applicable).

[00151] For some protocols (e.g., HTTP), typically, a connection is opened, a string is sent, a string is received, and the connection is closed. For other protocols (e.g., NFS), both sides of the connection engage in a constant chatter. Histograms allow data platform 12 to model application behavior (e.g., using machine learning techniques), for establishing baselines, and for detecting deviations. As one example, suppose that a given HTTP server typically sends/receives 1,000 bytes (in each direction) whenever a connection is made with it. If a connection generates 500 bytes of traffic, or 2,000 bytes of traffic, such connections would be considered within the typical usage pattern of the server. Suppose, however, that a connection is made that results in 10G of traffic. Such a connection is anomalous and can be flagged accordingly.

[00152] Returning to Fig. 1D, as previously mentioned, data aggregator 114 may be configured to provide information (e.g., collected from nodes 108 by agents) to data platform 12. Data aggregator 128 may be similarly configured to provide information to data platform 12. As shown in Fig. 1D, both aggregator 114 and aggregator 128 may connect to a load balancer 130, which accepts connections from aggregators (and/or as applicable, agents), as well as other devices, such as computer 126 (e.g., when it communicates with web app 120), and supports fair balancing. In various embodiments, load balancer 130 is a reverse proxy that load balances accepted connections internally to various microservices (described in more detail below), allowing for services provided by data platform 12 to scale up as more agents are added to the environment and/or as more entities subscribe to services provided by data platform 12. Example ways to implement load balancer 130 include, but are not limited to, using HaProxy, using nginx, and using elastic load balancing (ELB) services made available by Amazon.

[00153] Agent service 132 is a microservice that is responsible for accepting data collected from agents (e.g., provided by aggregator 114). In various embodiments, agent service 132 uses a standard secure protocol, such as HTTPS to communicate with aggregators (and as applicable agents), and receives data in an appropriate format such as Apache Avro. When agent service 132 receives an incoming connection, it can perform a variety of checks, such as to see whether the data is being provided by a current customer, and whether the data is being provided in an appropriate format. If the data is not appropriately formatted (and/or is not provided by a current customer), it may be rejected.

[00154] If the data is appropriately formatted, agent service 132 may facilitate copying the received data to a streaming data stable storage using a streaming service (e.g., Amazon Kinesis and/or any other suitable streaming service. Once the ingesting into the streaming service is

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00152

complete, service 132 may send an acknowledgement to the data provider (e.g., data aggregator 114). If the agent does not receive such an acknowledgement, it is configured to retry sending the data to data platform 12. One way to implement agent service 132 is as a REST API server framework (e.g., Java DropWizard), configured to communicate with Kinesis (e.g., using a Kinesis library).

[00155] In various embodiments, data platform 12 uses one or more streams (e.g., Kinesis streams) for all incoming customer data (e.g., including data provided by data aggregator 114 and data aggregator 128), and the data is sharded based on the node (also referred to herein as a "machine") that originated the data (e.g., node 116 vs. node 122), with each node having a globally unique identifier within data platform 12. Multiple instances of agent service 132 can write to multiple shards.

[00156] Kinesis is a streaming service with a limited period (e.g., 1-7 days). To persist data longer than a day, the data may be copied to long term storage 42 (e.g., S3). Data loader 136 is a microservice that is responsible for picking up data from a data stream (e.g., a Kinesis stream) and persisting it in long term storage 42. In one example embodiment, files collected by data loader 136 from the Kinesis stream are placed into one or more buckets, and segmented using a combination of a customer identifier and time slice. Given a particular time segment, and a given customer identifier, the corresponding file (stored in long term storage) contains five minutes (or another appropriate time slice) of data collected at that specific customer from all of the customer's nodes. Data loader 136 can be implemented in any appropriate programming language, such as Java or C, and can be configured to use a Kinesis library to interface with Kinesis. In various embodiments, data loader 136 uses the Amazon Simple Queue Service (SQS) (e.g., to alert DB loader 140 that there is work for it to do).

[00157] DB loader 140 is a microservice that is responsible for loading data into an appropriate data store 30, such as SnowflakeDB or Amazon Redshift, using individual per-customer databases. In particular, DB loader 140 is configured to periodically load data into a set of raw tables from files created by data loader 136 as per above. DB loader 140 manages throughput, errors, etc., to make sure that data is loaded consistently and continuously. Further, DB loader 140 can read incoming data and load into data store 30 data that is not already present in tables of data store 30 (also referred to herein as a database). DB loader 140 can be implemented in any appropriate programming language, such as Java or C, and an SQL framework such as jOOQ (e.g., to manage SQLs for insertion of data), and SQL/JDBC libraries. In some examples, DB loader 140 may use Amazon S3 and Amazon Simple Queue Service (SQS) to manage files being transferred to and from data store 30.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00153

[00158] Customer data included in data store 30 can be augmented with data from additional data sources, such as AWS CloudTrail and/or other types of external tracking services. To this end, data platform may include a tracking service analyzer 144, which is another microservice. Tracking service analyzer 144 may pull data from an external tracking service (e.g., Amazon CloudTrail) for each applicable customer account, as soon as the data is available. Tracking service analyzer 144 may normalize the tracking data as applicable, so that it can be inserted into data store 30 for later querying/analysis. Tracking service analyzer 144 can be written in any appropriate programming language, such as Java or C. Tracking service analyzer 144 also makes use of SQL/JDBC libraries to interact with data store 30 to insert/query data.

[00159] As described herein, data platform 12 can model activities that occur within datacenters, such as datacenters 104 and 106. The model may be stable over time, and differences, even subtle ones (e.g., between a current state of the datacenter and the model) can be surfaced. The ability to surface such anomalies can be particularly beneficial in datacenter environments where rogue employees and/or external attackers may operate slowly (e.g., over a period of months), hoping that the elastic nature of typical resource use (e.g., virtualized servers) will help conceal their nefarious activities.

[00160] Using techniques described herein, data platform 12 can automatically discover entities (which may implement compute assets 16) deployed in a given datacenter. Examples of entities include workloads, applications, processes, machines, virtual machines, containers, files, IP addresses, domain names, and users. The entities may be grouped together logically (into analysis groups) based on behaviors, and temporal behavior baselines can be established. In particular, using techniques described herein, periodic graphs can be constructed (also referred to herein as polygraphs), in which the nodes are applicable logical entities, and the edges represent behavioral relationships between the logical entities in the graph. Baselines can be created for every node and edge.

[00161] Communication (e.g., between applications/nodes) is one example of a behavior. A model of communications between processes is an example of a behavioral model. As another example, the launching of applications is another example of a behavior that can be modeled. The baselines may be periodically updated (e.g., hourly) for every entity. Additionally or alternatively, the baselines may be continuously updated in substantially real-time as data is collected by agents. Deviations from the expected normal behavior can then be detected and automatically reported (e.g., as anomalies or threats detected). Such deviations may be due to a desired change, a misconfiguration, or malicious activity. As applicable, data platform 12 can score the detected deviations (e.g., based on severity and threat posed). Additional examples of analysis groups include models of machine communications, models of privilege changes, and

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00154

models of insider behaviors (monitoring the interactive behavior of human users as they operate within the datacenter).

[00162] Two example types of information collected by agents are network level information and process level information. As previously mentioned, agents may collect information about every connection involving their respective nodes. And, for each connection, information about both the server and the client may be collected (e.g., using the connection-to-process identification techniques described above). DNS queries and responses may also be collected. The DNS query information can be used in logical entity graphing (e.g., collapsing many different IP addresses to a single service – e.g., s3.amazon.com). Examples of process level information collected by agents include attributes (user ID, effective user ID, and command line). Information such as what user/application is responsible for launching a given process and the binary being executed (and its SHA-256 values) may also be provided by agents.

[00163] The dataset collected by agents across a datacenter can be very large, and many resources (e.g., virtual machines, IP addresses, etc.) are recycled very quickly. For example, an IP address and port number used at a first point in time by a first process on a first virtual machine may very rapidly be used (e.g., an hour later) by a different process/virtual machine.

[00164] A dataset (and elements within it) can be considered at both a physical level, and a logical level, as illustrated in Fig. 2B. In particular, Fig. 2B illustrates an example 5-tuple of data 210 collected by an agent, represented physically (216) and logically (217). The 5-tuple includes a source address 211, a source port 212, a destination address 213, a destination port 214, and a protocol 215. In some cases, port numbers (e.g., 212, 214) may be indicative of the nature of a connection (e.g., with certain port usage standardized). However, in many cases, and in particular in datacenters, port usage is ephemeral. For example, a Docker container can listen on an ephemeral port, which is unrelated to the service it will run. When another Docker container starts (for the same service), the port may well be different. Similarly, particularly in a virtualized environment, IP addresses may be recycled frequently (and are thus also potentially ephemeral) or could be NATed, which makes identification difficult.

[00165] A physical representation of the 5-tuple is depicted in region 216. A process 218 (executing on machine 219) has opened a connection to machine 220. In particular, process 218 is in communication with process 221. Information such as the number of packets exchanged between the two machines over the respective ports can be recorded.

[00166] As previously mentioned, in a datacenter environment, portions of the 5-tuple may change – potentially frequently – but still be associated with the same behavior. Namely, one application (e.g., Apache) may frequently be in communication with another application (e.g., Oracle), using ephemeral datacenter resources. Further, either/both of Apache and Oracle may be

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00155

multi-homed. This can lead to potentially thousands of 5-tuples (or more) that all correspond to Apache communicating with Oracle within a datacenter. For example, Apache could be executed on a single machine, and could also be executed across fifty machines, which are variously spun up and down (with different IP addresses each time). An alternate representation of the 5-tuple of data 210 is depicted in region 217, and is logical. The logical representation of the 5-tuple aggregates the 5-tuple (along with other connections between Apache and Oracle having other 5-tuples) as logically representing the same connection. By aggregating data from raw physical connection information into logical connection information, using techniques described herein, a size reduction of six orders of magnitude in the data set can be achieved.

[00167] Fig. 2C depicts a portion of a logical polygraph. Suppose a datacenter has seven instances of the application update_engine 225, executing as seven different processes on seven different machines, having seven different IP addresses, and using seven different ports. The instances of update_engine variously communicate with update.core-os.net 226, which may have a single IP address or many IP addresses itself, over the one hour time period represented in the polygraph. In the example shown in Fig. 2C, update_engine is a client, connecting to the server update.core-os.net, as indicated by arrow 228.

[00168] Behaviors of the seven processes are clustered together, into a single summary. As indicated in region 227, statistical information about the connections is also maintained (e.g., number of connections, histogram information, etc.). A polygraph such as is depicted in Fig. 2C can be used to establish a baseline of behavior (e.g., at the one-hour level), allowing for the future detection of deviations from that baseline. As one example, suppose that statistically an update_engine instance transmits data at 11 bytes per second. If an instance were instead to transmit data at 1000 bytes per second, such behavior would represent a deviation from the baseline and could be flagged accordingly. Similarly, changes that are within the baseline (e.g., an eighth instance of update_engine appears, but otherwise behaves as the other instances; or one of the seven instances disappears) are not flagged as anomalous. Further, datacenter events, such as failover, autobalancing, and A-B refresh are unlikely to trigger false alarms in a polygraph, as at the logical level, the behaviors remain the same.

[00169] In various embodiments, polygraph data is maintained for every application in a datacenter, and such polygraph data can be combined to make a single datacenter view across all such applications. Fig. 2D illustrates a portion of a polygraph for a service that evidences more complex behaviors than are depicted in Fig. 2C. In particular, Fig. 2D illustrates the behaviors of S3 as a service (as used by a particular customer datacenter). Clients within the datacenter variously connect to the S3 service using one of five fully qualified domains (listed in region 230). Contact with any of the domains is aggregated as contact with S3 (as indicated in region

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00156

231). Depicted in region 232 are various containers which (as clients) connect with S3. Other containers (which do not connect with S3) are not included. As with the polygraph portion depicted in Fig. 2C, statistical information about the connections is known and summarized, such as the number of bytes transferred, histogram information, etc.

[00170] Fig. 2E illustrates a communication polygraph for a datacenter. In particular, the polygraph indicates a one hour summary of approximately 500 virtual machines, which collectively run one million processes, and make 100 million connections in that hour. As illustrated in Fig. 2E, a polygraph represents a drastic reduction in size (e.g., from tracking information on 100 million connections in an hour, to a few hundred nodes and a few hundred edges). Further, as a datacenter scales up (e.g., from using 10 virtual machines to 100 virtual machines as the datacenter uses more workers to support existing applications), the polygraph for the datacenter will tend to stay the same size (with the 100 virtual machines clustering into the same nodes that the 10 virtual machines previously clustered into). As new applications are added into the datacenter, the polygraph may automatically scale to include behaviors involving those applications.

[00171] In the particular polygraph shown in Fig. 2E, nodes generally correspond to workers, and edges correspond to communications the workers engage in (with connection activity being the behavior modeled in polygraph 235). Another example polygraph could model other behavior, such as application launching. The communications graphed in Fig. 2E include traffic entering the datacenter, traffic exiting the datacenter, and traffic that stays wholly within the datacenter (e.g., traffic between workers). One example of a node included in polygraph 235 is the sshd application, depicted as node 236. As indicated in Fig. 2E, 421 instances of sshd were executing during the one hour time period of data represented in polygraph 235. As indicated in region 237, nodes within the datacenter communicated with a total of 1349 IP addresses outside of the datacenter (and not otherwise accounted for, e.g., as belonging to a service such as Amazon AWS 238 or Slack 239).

[00172] In the following examples, suppose that user B, an administrator of datacenter 106, is interacting with data platform 12 to view visualizations of polygraphs in a web browser (e.g., as served to user B via web app 120). One type of polygraph user B can view is an application-communication polygraph, which indicates, for a given one hour window (or any other suitable time interval), which applications communicated with which other applications. Another type of polygraph user B can view is an application launch polygraph. User B can also view graphs related to user behavior, such as an insider behavior graph which tracks user connections (e.g., to internal and external applications, including chains of such behavior), a privilege change graph

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00157

which tracks how privileges change between processes, and a user login graph, which tracks which (logical) machines a user logs into.

[00173] Fig. 2F illustrates an example of an application-communication polygraph for a datacenter (e.g., datacenter 106) for the one hour period of 9am-10am on June 5. The time slice currently being viewed is indicated in region 240. If user B clicks his mouse in region 241, user B will be shown a representation of the application-communication polygraph as generated for the following hour (10am-11am on June 5).

[00174] Fig. 2G depicts what is shown in user B's browser after he has clicked on region 241, and has further clicked on region 242. The selection in region 242 turns on and off the ability to compare two time intervals to one another. User B can select from a variety of options when comparing the 9am-10am and 10am-11am time intervals. By clicking region 248, user B will be shown the union of both graphs (i.e., any connections that were present in either time interval). By clicking region 249, user B will be shown the intersection of both graphs (i.e., only those connections that were present in both time intervals).

[00175] As shown in Fig. 2G, user B has elected to click on region 250, which depicts connections that are only present in the 9am-10am polygraph in a first color 251, and depicts connections that are only present in the 10am-11am polygraph in a second color 252. Connections present in both polygraphs are omitted from display. As one example, in the 9am-10am polygraph (corresponding to connections made during the 9am-10am time period at datacenter 106), a connection was made by a server to sshd (253) and also to systemd (254). Both of those connections ended prior to 10am and are thus depicted in the first color. As another example, in the 10am-11am polygraph (corresponding to connections made during the 10am-11am time period at datacenter 106), a connection was made from a known bad external IP to nginx (255). The connection was not present during the 9am-10am time slice and thus is depicted in the second color. As yet another example, two different connections were made to a Slack service between 9am and 11am. However, the first was made by a first client during the 9am-10am time slice (256) and the second was made by a different client during the 10am-11am slice (257), and so the two connections are depicted respectively in the first and second colors and blue.

[00176] Returning to the polygraph depicted in Fig. 2F, suppose user B enters "etcd" into the search box located in region 244. User B will then be presented with the interface illustrated in Fig. 2H. As shown in Fig. 2H, three applications containing the term "etcd" were engaged in communications during the 9am-10am window. One application is etcdctl, a command line client for etcd. As shown in Fig. 2H, a total of three different etcdctl processes were executed during the 9am-10am window, and were clustered together (260). Fig. 2H also depicts two different

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00158

clusters that are both named etcd2. The first cluster includes (for the 9am-10am window) five members (261) and the second cluster includes (for the same window) eight members (262). The reason for these two distinct clusters is that the two groups of applications behave differently (e.g., they exhibit two distinct sets of communication patterns). Specifically, the instances of etcd2 in cluster 261 only communicate with locksmithctl (263) and other etcd2 instances (in both clusters 261 and 262). The instances of etcd2 in cluster 262 communicate with additional entities, such as etcdctl and Docker containers. As desired, user B can click on one of the clusters (e.g., cluster 261) and be presented with summary information about the applications included in the cluster, as is shown in Fig. 2I (e.g., in region 265). User B can also double click on a given cluster (e.g., cluster 261) to see details on each of the individual members of the cluster broken out.

[00177] Suppose user B now clicks on region 245 of the interface shown in Fig. 2F. User B will then be shown an application launch polygraph. Launching an application is another example of a behavior. The launch polygraph models how applications are launched by other applications. Fig. 2J illustrates an example of a portion of a launch polygraph. In particular, user B has typed "find" into region 266, to see how the "find" application is being launched. As shown in Fig. 2J, in the launch polygraph for the 10am-11am time period, find applications (267) are always launched by bash (268), which is in turn always launched by systemd (269). If find is launched by a different application, this would be anomalous behavior.

[00178] Fig. 2K illustrates another example of a portion of an application launch polygraph. In Fig. 2K, user B has searched (270) for "python ma" to see how "python marathon_lb" (271) is launched. As shown in Fig. 2K, in each case (during the one hour time slice of 10am-11am), python marathon_lb is launched as a result of a chain of the same seven applications each time. If python marathon_lb is ever launched in a different manner, this indicates anomalous behavior. The behavior could be indicative of malicious activities, but could also be due to other reasons, such as a misconfiguration, a performance-related issue, and/or a failure, etc.

[00179] Suppose user B now clicks on region 246 of the interface shown in Fig. 2F. User B will then be shown an insider behavior graph. The insider behavior graph tracks information about behaviors such as processes started by a user interactively using protocols such as ssh or telnet, and any processes started by those processes. As one example, suppose an administrator logs into a first virtual machine in datacenter 106 (e.g., using sshd via an external connection he makes from a hotel), using a first set of credentials (e.g., first.last@example.com and an appropriate password). From the first virtual machine, the administrator connects to a second virtual machine (e.g., using the same credentials), then uses the sudo command to change identities to those of another user, and then launches a program. graphs built by data platform 12 can be used to

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00159

associate the administrator with each of his actions, including launching the program using the identity of another user.

[00180] Fig. 2L illustrates an example of a portion of an insider behavior graph. In particular, in Fig. 2L, user B is viewing a graph that corresponds to the time slice of 3pm-4pm on June 1. Fig. 2L illustrates the internal/external applications that users connected to during the one hour time slice. If a user typically communicates with particular applications, that information will become part of a baseline. If the user deviates from his baseline behavior (e.g., using new applications, or changing privilege in anomalous ways), such anomalies can be surfaced.

[00181] Fig. 2M illustrates an example of a portion of a privilege change graph, which identifies how privileges are changed between processes. Typically, when a user launches a process (e.g., "ls"), the process inherits the same privileges that the user has. And, while a process can have fewer privileges than the user (i.e., go down in privilege), it is rare (and generally undesirable) for a user to escalate in privilege. Information included in the privilege change graph can be determined by examining the parent of each running process, and determining whether there is a match in privilege between the parent and the child. If the privileges are different, a privilege change has occurred (whether a change up or a change down). The application ntpd is one rare example of a scenario in which a process escalates (272) to root, and then returns back (273). The sudo command is another example (e.g., used by an administrator to temporarily have a higher privilege). As with the other examples, ntpd's privilege change actions, and the legitimate actions of various administrators (e.g., using sudo) will be incorporated into a baseline model by data platform 12. When deviations occur, such as where a new application that is not ntpd escalates privilege, or where an individual that has not previously/does not routinely use sudo does so, such behaviors can be identified as anomalous.

[00182] Fig. 2N illustrates an example of a portion of a user login graph, which identifies which users log into which logical nodes. Physical nodes (whether bare metal or virtualized) are clustered into a logical machine cluster, for example, using yet another graph, a machine-server graph, an example of which is shown in Fig. 2O. For each machine, a determination is made as to what type of machine it is, based on what kind(s) of workflows it runs. As one example, some machines run as master nodes (having a typical set of workflows they run, as master nodes) and can thus be clustered as master nodes. Worker nodes are different from master nodes, for example, because they run Docker containers, and frequently change as containers move around. Worker nodes can similarly be clustered.

[00183] As previously mentioned, the polygraph depicted in Fig. 2E corresponds to activities in a datacenter in which, in a given hour, approximately 500 virtual machines collectively run one million processes, and make 100 million connections in that hour. The polygraph represents a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00160

drastic reduction in size (e.g., from tracking information on 100 million connections in an hour, to a few hundred nodes and a few hundred edges). Using techniques described herein, such a polygraph can be constructed (e.g., using commercially available computing infrastructure) in less than an hour (e.g., within a few minutes). Thus, ongoing hourly snapshots of a datacenter can be created within a two hour moving window (i.e., collecting data for the time period 8am-9am, while also generating a snapshot for the time previous time period 7am-8am). The following describes various example infrastructure that can be used in polygraph construction, and also describes various techniques that can be used to construct polygraphs.

[00184] Returning to Fig. 1D, embodiments of data platform 12 may be built using any suitable infrastructure as a service (IaaS) (e.g., AWS). For example, data platform 12 can use Simple Storage Service (S3) for data storage, Key Management Service (KMS) for managing secrets, Simple Queue Service (SQS) for managing messaging between applications, Simple Email Service (SES) for sending emails, and Route 53 for managing DNS. Other infrastructure tools can also be used. Examples include: orchestration tools (e.g., Kubernetes or Mesos/Marathon), service discovery tools (e.g., Mesos-DNS), service load balancing tools (e.g., marathon-LB), container tools (e.g., Docker or rkt), log/metric tools (e.g., collectd, fluentd, kibana, etc.), big data processing systems (e.g., Spark, Hadoop, AWS Redshift, Snowflake etc.), and distributed key value stores (e.g., Apache Zookeeper or etcd2).

[00185] As previously mentioned, in various embodiments, data platform 12 may make use of a collection of microservices. Each microservice can have multiple instances, and may be configured to recover from failure, scale, and distribute work amongst various such instances, as applicable. For example, microservices are auto-balancing for new instances, and can distribute workload if new instances are started or existing instances are terminated. In various embodiments, microservices may be deployed as self-contained Docker containers. A Mesos-Marathon or Spark framework can be used to deploy the microservices (e.g., with Marathon monitoring and restarting failed instances of microservices as needed). The service etcd2 can be used by microservice instances to discover how many peer instances are running, and used for calculating a hash-based scheme for workload distribution. Microservices may be configured to publish various health/status metrics to either an SQS queue, or etcd2, as applicable. In some examples, Amazon DynamoDB can be used for state management.

[00186] Additional information on various microservices used in embodiments of data platform 12 is provided below.

[00187] Graph generator 146 is a microservice that may be responsible for generating raw behavior graphs on a per customer basis periodically (e.g., once an hour). In particular, graph generator 146 may generate graphs of entities (as the nodes in the graph) and activities between

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00161

entities (as the edges). In various embodiments, graph generator 146 also performs other functions, such as aggregation, enrichment (e.g., geolocation and threat), reverse DNS resolution, TF-IDF based command line analysis for command type extraction, parent process tracking, etc.

[00188] Graph generator 146 may perform joins on data collected by the agents, so that both sides of a behavior are linked. For example, suppose a first process on a first virtual machine (e.g., having a first IP address) communicates with a second process on a second virtual machine (e.g., having a second IP address). Respective agents on the first and second virtual machines may each report information on their view of the communication (e.g., the PID of their respective processes, the amount of data exchanged and in which direction, etc.). When graph generator performs a join on the data provided by both agents, the graph will include a node for each of the processes, and an edge indicating communication between them (as well as other information, such as the directionality of the communication – i.e., which process acted as the server and which as the client in the communication).

[00189] In some cases, connections are process to process (e.g., from a process on one virtual machine within the cloud environment associated with entity A to another process on a virtual machine within the cloud environment associated with entity A). In other cases, a process may be in communication with a node (e.g., outside of entity A) which does not have an agent deployed upon it. As one example, a node within entity A might be in communication with node 172, outside of entity A. In such a scenario, communications with node 172 are modeled (e.g., by graph generator 146) using the IP address of node 172. Similarly, where a node within entity A does not have an agent deployed upon it, the IP address of the node can be used by graph generator in modeling.

[00190] Graphs created by graph generator 146 may be written to data store 30 and cached for further processing. A graph may be a summary of all activity that happened in a particular time interval. As each graph corresponds to a distinct period of time, different rows can be aggregated to find summary information over a larger timestamp. In some examples, picking two different graphs from two different timestamps can be used to compare different periods. If necessary, graph generator can parallelize its workload (e.g., where its backlog cannot otherwise be handled within a particular time period, such as an hour, or if is required to process a graph spanning a long time period).

[00191] Graph generator 146 can be implemented in any appropriate programming language, such as Java or C, and machine learning libraries, such as Spark's MLLib. Example ways that graph generator computations can be implemented include using SQL or Map-R, using Spark or Hadoop.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00162

[00192] SSH tracker 148 is a microservice that may be responsible for following ssh connections and process parent hierarchies to determine trails of user ssh activity. Identified ssh trails are placed by the SSH tracker 148 into data store 30 and cached for further processing.

[00193] SSH tracker 148 can be implemented in any appropriate programming language, such as Java or C, and machine libraries, such as Spark's MLLib. Example ways that SSH tracker computations can be implemented include using SQL or Map-R, using Spark or Hadoop.

[00194] Threat aggregator 150 is a microservice that may be responsible for obtaining third party threat information from various applicable sources, and making it available to other micro-services. Examples of such information include reverse DNS information, GeoIP information, lists of known bad domains/IP addresses, lists of known bad files etc. As applicable, the threat information is normalized before insertion into data store 30. Threat aggregator 150 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 (e.g., for insertions and queries).

[00195] Scheduler 152 is a microservice that may act as a scheduler and that may run arbitrary jobs organized as a directed graph. In some examples, scheduler 152 ensures that all jobs for all customers are able to run during at a given time interval (e.g., every hour). Scheduler 152 may handle errors and retrying for failed jobs, track dependencies, manage appropriate resource levels, and/or scale jobs as needed. Scheduler 152 can be implemented in any appropriate programming language, such as Java or C. A variety of components can also be used, such as open source scheduler frameworks (e.g., Airflow), or AWS services (e.g., the AWS Data pipeline) which can be used for managing schedules.

[00196] Graph Behavior Modeler (GBM) 154 is a microservice that may compute polygraphs. In particular, GBM 154 can be used to find clusters of nodes in a graph that should be considered similar based on some set of their properties and relationships to other nodes. As described herein, the clusters and their relationships can be used to provide visibility into a datacenter environment without requiring user specified labels. GBM 154 may track such clusters over time persistently, allowing for changes to be detected and alerts to be generated.

[00197] GBM 154 may take as input a raw graph (e.g., as generated by graph generator 146). Nodes are actors of a behavior, and edges are the behavior relationship itself. For example, in the case of communication, example actors include processes, which communicate with other processes. The GBM 154 clusters the raw graph based on behaviors of actors and produces a summary (the polygraph). The polygraph summarizes behavior at a datacenter level. The GBM also produces "observations" that represent changes detected in the datacenter. Such observations may be based on differences in cumulative behavior (e.g., the baseline) of the datacenter with its current behavior. The GBM 154 can be implemented in any appropriate programming language,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00163

such as Java, C, or Golang, using appropriate libraries (as applicable) to handle distributed graph computations (handling large amounts of data analysis in a short amount of time). Apache Spark is another example tool that can be used to compute polygraphs. The GBM can also take feedback from users and adjust the model according to that feedback. For example, if a given user is interested in relearning behavior for a particular entity, the GBM can be instructed to "forget" the implicated part of the polygraph.

[00198] GBM runner 156 is a microservice that may be responsible for interfacing with GBM 154 and providing GBM 154 with raw graphs (e.g., using a query language, such as SQL, to push any computations it can to data store 30). GBM runner 156 may also insert polygraph output from GBM 154 to data store 30. GBM runner 156 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 to insert and query data.

[00199] Alert generator 158 is a microservice that may be responsible for generating alerts. Alert generator 158 may examine observations (e.g., produced by GBM 154) in aggregate, deduplicate them, and score them. Alerts may be generated for observations with a score exceeding a threshold. Alert generator 158 may also compute (or retrieves, as applicable) data that a customer (e.g., user A or user B) might need when reviewing the alert. Examples of events that can be detected by data platform 12 (and alerted on by alert generator 158) include, but are not limited to the following:

[00200] new user: This event may be created the first time a user (e.g., of node 116) is first observed by an agent within a datacenter.

[00201] user launched new binary: This event may be generated when an interactive user launches an application for the first time.

[00202] new privilege escalation: This event may be generated when user privileges are escalated and a new application is run.

[00203] new application or container: This event may be generated when an application or container is seen for the first time.

[00204] new external connection: This event may be generated when a connection to an external IP/domain is made from a new application.

[00205] new external host or IP: This event may be generated when a new external host or IP is involved in a connection with a datacenter.

[00206] new internal connection: This event may be generated when a connection between internal-only applications is seen for the first time.

[00207] new external client: This event may be generated when a new external connection is seen for an application which typically does not have external connections.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00164

[00208] new parent: This event may be generated when an application is launched by a different parent.

[00209] connection to known bad IP/domain: Data platform 12 maintains (or can otherwise access) one or more reputation feeds. If an environment makes a connection to a known bad IP or domain, an event will be generated.

[00210] login from a known bad IP/domain: An event may be generated when a successful connection to a datacenter from a known bad IP is observed by data platform 12.

[00211] Alert generator 158 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 to insert and query data. In various embodiments, alert generator 158 also uses one or more machine learning libraries, such as Spark's MLLib (e.g., to compute scoring of various observations). Alert generator 158 can also take feedback from users about which kinds of events are of interest and which to suppress.

[00212] QsJobServer 160 is a microservice that may look at all the data produced by data platform 12 for an hour, and compile a materialized view (MV) out of the data to make queries faster. The MV helps make sure that the queries customers most frequently run, and data that they search for, can be easily queried and answered. QsJobServer 160 may also precompute and cache a variety of different metrics so that they can quickly be provided as answers at query time. QsJobServer 160 can be implemented using any appropriate programming language, such as Java or C, using SQL/JDBC libraries. In some examples, QsJobServer 160 is able to compute an MV efficiently at scale, where there could be a large number of joins. An SQL engine, such as Oracle, can be used to efficiently execute the SQL, as applicable.

[00213] Alert notifier 162 is a microservice that may take alerts produced by alert generator 158 and send them to customers' integrated Security Information and Event Management (SIEM) products (e.g., Splunk, Slack, etc.). Alert notifier 162 can be implemented using any appropriate programming language, such as Java or C. Alert notifier 162 can be configured to use an email service (e.g., AWS SES or pagerduty) to send emails. Alert notifier 162 may also provide templating support (e.g., Velocity or Moustache) to manage templates and structured notifications to SIEM products.

[00214] Reporting module 164 is a microservice that may be responsible for creating reports out of customer data (e.g., daily summaries of events, etc.) and providing those reports to customers (e.g., via email). Reporting module 164 can be implemented using any appropriate programming language, such as Java or C. Reporting module 164 can be configured to use an email service (e.g., AWS SES or pagerduty) to send emails. Reporting module 164 may also provide templating support (e.g., Velocity or Moustache) to manage templates (e.g., for constructing HTML-based email).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00165

[00215] Web app 120 is a microservice that provides a user interface to data collected and processed on data platform 12. Web app 120 may provide login, authentication, query, data visualization, etc. features. Web app 120 may, in some embodiments, include both client and server elements. Example ways the server elements can be implemented are using Java DropWizard or Node.Js to serve business logic, and a combination of JSON/HTTP to manage the service. Example ways the client elements can be implemented are using frameworks such as React, Angular, or Backbone. JSON, jQuery, and JavaScript libraries (e.g., underscore) can also be used.

[00216] Query service 166 is a microservice that may manage all database access for web app 120. Query service 166 abstracts out data obtained from data store 30 and provides a JSON-based REST API service to web app 120. Query service 166 may generate SQL queries for the REST APIs that it receives at run time. Query service 166 can be implemented using any appropriate programming language, such as Java or C and SQL/JDBC libraries, or an SQL framework such as jOOQ. Query service 166 can internally make use of a variety of types of databases, including a relational database engine 168 (e.g., AWS Aurora) and/or data store 30 to manage data for clients. Examples of tables that query service 166 manages are OLTP tables and data warehousing tables.

[00217] Cache 170 may be implemented by Redis and/or any other service that provides a key-value store. Data platform 12 can use cache 170 to keep information for frontend services about users. Examples of such information include valid tokens for a customer, valid cookies of customers, the last time a customer tried to login, etc.

[00218] Fig. 3A illustrates an example of a process for detecting anomalies in a network environment. In various embodiments, process 300 is performed by data platform 12. The process begins at 301 when data associated with activities occurring in a network environment (such as entity A's datacenter) is received. One example of such data that can be received at 301 is agent-collected data described above (e.g., in conjunction with process 200).

[00219] At 302, a logical graph model is generated, using at least a portion of the monitored activities. A variety of approaches can be used to generate such logical graph models, and a variety of logical graphs can be generated (whether using the same, or different approaches). The following is one example of how data received at 301 can be used to generate and maintain a model.

[00220] During bootstrap, data platform 12 creates an aggregate graph of physical connections (also referred to herein as an aggregated physical graph) by matching connections that occurred in the first hour into communication pairs. Clustering is then performed on the communication pairs. Examples of such clustering, described in more detail below, include performing Matching

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00166

Neighbor clustering and similarity (e.g., SimRank) clustering. Additional processing can also be performed (and is described in more detail below), such as by splitting clusters based on application type, and annotating nodes with DNS query information. The resulting graph (also referred to herein as a base graph or common graph) can be used to generate a variety of models, where a subset of node and edge types (described in more detail below) and their properties are considered in a given model. One example of a model is a UID to UID model (also referred to herein as a Uid2Uid model) which clusters together processes that share a username and show similar privilege change behavior. Another example of a model is a CType model, which clusters together processes that share command line similarity. Yet another example of a model is a PType model, which clusters together processes that share behaviors over time.

[00221] Each hour (or any other predetermined time interval) after bootstrap, a new snapshot is taken (i.e., data collected about a datacenter in the last hour is processed) and information from the new snapshot is merged with existing data to create and (as additional data is collected/processed) maintain a cumulative graph. The cumulative graph (also referred to herein as a cumulative PType graph and a polygraph) is a running model of how processes behave over time. Nodes in the cumulative graph are PType nodes, and provide information such as a list of all active processes and PIDs in the last hour, the number of historic total processes, the average number of active processes per hour, the application type of the process (e.g., the CType of the PType), and historic CType information/frequency. Edges in the cumulative graph can represent connectivity and provide information such as connectivity frequency. The edges can be weighted (e.g., based on number of connections, number of bytes exchanged, etc.). Edges in the cumulative graph (and snapshots) can also represent transitions.

[00222] One approach to merging a snapshot of the activity of the last hour into a cumulative graph is as follows. An aggregate graph of physical connections is made for the connections included in the snapshot (as was previously done for the original snapshot used during bootstrap). And, clustering/splitting is similarly performed on the snapshot's aggregate graph. Next, PType clusters in the snapshot's graph are compared against PType clusters in the cumulative graph to identify commonality.

[00223] One approach to determining commonality is, for any two nodes that are members of a given CmdType (described in more detail below), comparing internal neighbors and calculating a set membership Jaccard distance. The pairs of nodes are then ordered by decreasing similarity (i.e., with the most similar sets first). For nodes with a threshold amount of commonality (e.g., at least 66% members in common), any new nodes (i.e., appearing in the snapshot's graph but not the cumulative graph) are assigned the same PType identifier as is assigned to the corresponding node in the cumulative graph. For each node that is not classified (i.e., has not been assigned a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00167

PType identifier), a network signature is generated (i.e., indicative of the kinds of network connections the node makes, who the node communicates with, etc.). The following processing is then performed until convergence. If a match of the network signature is found in the cumulative graph, the unclassified node is assigned the PType identifier of the corresponding node in the cumulative graph. Any nodes which remain unclassified after convergence are new PTypes and are assigned new identifiers and added to the cumulative graph as new. As applicable, the detection of a new PType can be used to generate an alert. If the new PType has a new CmdType, a severity of the alert can be increased. If any surviving nodes (i.e., present in both the cumulative graph and the snapshot graph) change PTypes, such change is noted as a transition, and an alert can be generated. Further, if a surviving node changes PType and also changes CmdType, a severity of the alert can be increased.

[00224] Changes to the cumulative graph (e.g., a new PType or a new edge between two PTypes) can be used (e.g., at 303) to detect anomalies (described in more detail below). Two example kinds of anomalies that can be detected by data platform 12 include security anomalies (e.g., a user or process behaving in an unexpected manner) and devops/root cause anomalies (e.g., network congestion, application failure, etc.). Detected anomalies can be recorded and surfaced (e.g., to administrators, auditors, etc.), such as through alerts which are generated at 304 based on anomaly detection.

[00225] Additional detail regarding processing performed, by various components depicted in Fig. 1D (whether performed individually or in combination), in conjunction with model/polygraph construction (e.g., as performed at 302) are provided below.

[00226] As explained above, an aggregated physical graph can be generated on a per customer basis periodically (e.g., once an hour) from raw physical graph information, by matching connections (e.g., between two processes on two virtual machines). In various embodiments, a deterministic fixed approach is used to cluster nodes in the aggregated physical graph (e.g., representing processes and their communications). As one example, Matching Neighbors Clustering (MNC) can be performed on the aggregated physical graph to determine which entities exhibit identical behavior and cluster such entities together.

[00227] Fig. 3B depicts a set of example processes (p1, p2, p3, and p4) communicating with other processes (p10 and p11). Fig. 3B is a graphical representation of a small portion of an aggregated physical graph showing (for a given time period, such as an hour) which processes in a datacenter communicate with which other processes. Using MNC, processes p1, p2, and p3 will be clustered together (305), as they exhibit identical behavior (they communicate with p10 and only p10). Process p4, which communicates with both p10 and p11, will be clustered separately.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00168

[00228] In MNC, only those processes exhibiting identical (communication) behavior will be clustered. In various embodiments, an alternate clustering approach can also/instead be used, which uses a similarity measure (e.g., constrained by a threshold value, such as a 60% similarity) to cluster items. In some embodiments, the output of MNC is used as input to SimRank, in other embodiments, MNC is omitted.

[00229] Fig. 3C depicts a set of example processes (p4, p5, p6) communicating with other processes (p7, p8, p9). As illustrated, most of nodes p4, p5, and p6 communicate with most of nodes p7, p8, and p9 (as indicated in Fig. 3C with solid connection lines). As one example, process p4 communicates with process p7 (310), process p8 (311), and process p9 (312). An exception is process p6, which communicates with processes p7 and p8, but does not communicate with process p9 (as indicated by dashed line 313). If MNC were applied to the nodes depicted in Fig. 3C, nodes p4 and p5 would be clustered (and node p6 would not be included in their cluster).

[00230] One approach to similarity clustering is to use SimRank. In an embodiment of the SimRank approach, for a given node v in a directed graph, $I(v)$ and $O(v)$ denote the respective set of in-neighbors and out-neighbors of v. Individual in-neighbors are denoted as $I_i(v)$, for $1 \leq i \leq |I(v)|$, and individual out-neighbors are denoted as $O_i(v)$, for $1 \leq i \leq |O(v)|$. The similarity between two objects a and b can be denoted by $s(a,b) \in [1,0]$. A recursive equation (hereinafter "the SimRank equation") can be written for $s(a,b)$, where, if a=b, then $s(a,b)$ is defined as 1, otherwise, $s(a,b)=\frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$ where C is a constant between 0 and 1. One example value for the decay factor C is 0.8 (and a fixed number of iterations such as five). Another example value for the decay factor C is 0.6 (and/or a different number of iterations). In the event that a or b has no in-neighbors, similarity is set to $s(a,b)=0$, so the summation is defined to be 0 when $I(a)=\emptyset$ or $I(b)=\emptyset$.

[00231] The SimRank equations for a graph G can be solved by iteration to a fixed point. Suppose n is the number of nodes in G. For each iteration k, n^2 entries $s_k(*,*)$ are kept, where $s_k(a,b)$ gives the score between a and b on iteration k. Successive computations of $s_{k+1}(*,*)$ are made based on $s_k(*,*)$. Starting with $s_0(*,*)$, where each $s_0(a,b)$ is a lower bound on the actual SimRank score $s(a,b)$: $s_0(a,b) = \begin{cases} 1, \text{if } a = b, \\ 0, \text{if } a \neq b. \end{cases}$

[00232] The SimRank equation can be used to compute $s_{k+1}(a,b)$ from $s_k(*,*)$ with $s_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s_k(I_i(a), I_j(b))$ for $a \neq b$, and $s_{k+1}(a,b)=1$ for $a=b$. On each iteration k+1, the similarity of (a,b) is updated using the similarity scores of the neighbors of (a,b) from the previous iteration k according to the SimRank equation. The values $s_k(*,*)$ are nondecreasing as k increases.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00169

[00233] Returning to Fig. 3C, while MNC would cluster nodes p4 and p5 together (and not include node p6 in their cluster), application of SimRank would cluster nodes p4-p6 into one cluster (314) and also cluster nodes p7-p9 into another cluster (315).

[00234] Fig. 3D depicts a set of processes, and in particular server processes s1 and s2, and client processes c1, c2, c3, c4, c5, and c6. Suppose only nodes s1, s2, c1, and c2 are present in the graph depicted in Fig. 3D (and the other nodes depicted are omitted from consideration). Using MNC, nodes s1 and s2 would be clustered together, as would nodes c1 and c2. Performing SimRank clustering as described above would also result in those two clusters (s1 and s2, and c1 and c2). As previously mentioned, in MNC, identical behavior is required. Thus, if node c3 were now also present in the graph, MNC would not include c3 in a cluster with c2 and c1 because node c3 only communicates with node s2 and not node s1. In contrast, a SimRank clustering of a graph that includes nodes s1, s2, c1, c2, and c3 would result (based, e.g., on an applicable selected decay value and number of iterations) in a first cluster comprising nodes s1 and s2, and a second cluster of c1, c2, and c3. As an increasing number of nodes which communicate with server process s2, and do not also communicate with server process s1, are included in the graph (e.g., as c4, c5, and c6 are added), under SimRank, nodes s1 and s2 will become decreasingly similar (i.e., their intersection is reduced).

[00235] In various embodiments, SimRank is modified (from what is described above) to accommodate differences between the asymmetry of client and server connections. As one example, SimRank can be modified to use different thresholds for client communications (e.g., an 80% match among nodes c1-c6) and for server communications (e.g., a 60% match among nodes s1 and s2). Such modification can also help achieve convergence in situations such as where a server process dies on one node and restarts on another node.

[00236] The application of MNC/SimRank to an aggregated physical graph results in a smaller graph, in which processes which are determined to be sufficiently similar are clustered together. Typically, clusters generated as output of MNC will be underinclusive. For example, for the nodes depicted in Fig. 3C, process p6 will not be included in a cluster with processes p4 and p5, despite substantial similarity in their communication behaviors. The application of SimRank (e.g., to the output of MNC) helps mitigate the underinclusiveness of MNC, but can result in overly inclusive clusters. As one example, suppose (returning to the nodes depicted in Fig. 3B) that as a result of applying SimRank to the depicted nodes, nodes p1-p4 are all included in a single cluster. Both MNC and SimRank operate agnostically of which application a given process belongs to. Suppose processes p1-p3 each correspond to a first application (e.g., an update engine), and process p4 corresponds to a second application (e.g., sshd). Further suppose process p10 corresponds to contact with AWS. Clustering all four of the processes together (e.g., as a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00170

result of SimRank) could be problematic, particularly in a security context (e.g., where granular information useful in detecting threats would be lost).

[00237] As previously mentioned, data platform 12 may maintain a mapping between processes and the applications to which they belong. In various embodiments, the output of SimRank (e.g., SimRank clusters) is split based on the applications to which cluster members belong (such a split is also referred to herein as a "CmdType split"). If all cluster members share a common application, the cluster remains. If different cluster members originate from different applications, the cluster members are split along application-type (CmdType) lines. Using the nodes depicted in Fig. 3D as an example, suppose that nodes c1, c2, c3, and c5 all share "update engine" as the type of application to which they belong (sharing a CmdType). Suppose that node c4 belongs to "ssh," and suppose that node c6 belongs to "bash." As a result of SimRank, all six nodes (c1-c6) might be clustered into a single cluster. After a CmdType split is performed on the cluster, however, the single cluster will be broken into three clusters (c1, c2, c3, c5; c4; and c6). Specifically, the resulting clusters comprise processes associated with the same type of application, which exhibit similar behaviors (e.g., communication behaviors). Each of the three clusters resulting from the CmdType split represents, respectively, a node (also referred to herein as a PType) of a particular CmdType. Each PType is given a persistent identifier and stored persistently as a cumulative graph.

[00238] A variety of approaches can be used to determine a CmdType for a given process. As one example, for some applications (e.g., sshd), a one-to-one mapping exists between the CmdType and the application/binary name. Thus, processes corresponding to the execution of sshd will be classified using a CmdType of sshd. In various embodiments, a list of common application/binary names (e.g., sshd, apache, etc.) is maintained by data platform 12 and manually curated as applicable. Other types of applications (e.g., Java, Python, and Ruby) are multi-homed, meaning that several very different applications may all execute using the binary name, "java." For these types of applications, information such as command line / execution path information can be used in determining a CmdType. In particular, the subapplication can be used as the CmdType of the application, and/or term frequency analysis (e.g., TF/IDF) can be used on command line information to group, for example, any marathon related applications together (e.g., as a python.marathon CmdType) and separately from other Python applications (e.g., as a python.airflow CmdType).

[00239] In various embodiments, machine learning techniques are used to determine a CmdType. The CmdType model is constrained such that the execution path for each CmdType is unique. One example approach to making a CmdType model is a random forest based approach. An initial CmdType model is bootstrapped using process parameters (e.g., available within one

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00171

minute of process startup) obtained using one hour of information for a given customer (e.g., entity A). Examples of such parameters include the command line of the process, the command line of the process's parent(s) (if applicable), the uptime of the process, UID/EUID and any change information, TTY and any change information, listening ports, and children (if any). Another approach is to perform term frequency clustering over command line information to convert command lines into cluster identifiers.

[00240] The random forest model can be used (e.g., in subsequent hours) to predict a CmdType for a process (e.g., based on features of the process). If a match is found, the process can be assigned the matching CmdType. If a match is not found, a comparison between features of the process and its nearest CmdType (e.g., as determined using a Levenstein distance) can be performed. The existing CmdType can be expanded to include the process, or, as applicable, a new CmdType can be created (and other actions taken, such as generating an alert). Another approach to handling processes which do not match an existing CmdType is to designate such processes as unclassified, and once an hour, create a new random forest seeded with process information from a sampling of classified processes (e.g., 10 or 100 processes per CmdType) and the new processes. If a given new process winds up in an existing set, the process is given the corresponding CmdType. If a new cluster is created, a new CmdType can be created.

[00241] Conceptually, a polygraph represents the smallest possible graph of clusters that preserve a set of rules (e.g., in which nodes included in the cluster must share a CmdType and behavior). As a result of performing MNC, SimRank, and cluster splitting (e.g., CmdType splitting) many processes are clustered together based on commonality of behavior (e.g., communication behavior) and commonality of application type. Such clustering represents a significant reduction in graph size (e.g., compared to the original raw physical graph). Nonetheless, further clustering can be performed (e.g., by iterating on the graph data using the GBM to achieve such a polygraph). As more information within the graph is correlated, more nodes can be clustered together, reducing the size of the graph, until convergence is reached and no further clustering is possible.

[00242] Fig. 3E depicts two pairs of clusters. In particular, cluster 320 represents a set of client processes sharing the same CmdType ("a1"), communicating (collectively) with a server process having a CmdType ("a2"). Cluster 322 also represents a set of client processes having a CmdType a1 communicating with a server process having a CmdType a2. The nodes in clusters 320 and 322 (and similarly nodes in 321 and 323) remain separately clustered (as depicted) after MNC/SimRank/CmdType splitting – isolated islands. One reason this could occur is where server process 321 corresponds to processes executing on a first machine (having an IP address

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00172

of 1.1.1.1). The machine fails and a new server process 323 starts, on a second machine (having an IP address of 2.2.2.2) and takes over for process 321.

[00243] Communications between a cluster of nodes (e.g., nodes of cluster 320) and the first IP address can be considered different behavior from communications between the same set of nodes and the second IP address, and thus communications 324 and 325 will not be combined by MNC/SimRank in various embodiments. Nonetheless, it could be desirable for nodes of clusters 320/322 to be combined (into cluster 326), and for nodes of clusters 321/323 to be combined (into cluster 327), as representing (collectively) communications between a1 and a2. One task that can be performed by data platform 12 is to use DNS query information to map IP addresses to logical entities. As will be described in more detail below, GBM 154 can make use of the DNS query information to determine that graph nodes of cluster 320 and graph nodes of cluster 322 both made DNS queries for "appserverabc.example.com," which first resolved to 1.1.1.1 and then to 2.2.2.2, and to combine nodes 320/322 and 321/323 together into a single pair of nodes (326 communicating with 327).

[00244] In various embodiments, GBM 154 operates in a batch manner in which it receives as input the nodes and edges of a graph for a particular time period along with its previous state, and generates as output clustered nodes, cluster membership edges, cluster-to-cluster edges, events, and its next state.

[00245] GBM 154 may not try to consider all types of entities and their relationships that may be available in a conceptual common graph all at once. Instead, GBM uses a concept of models where a subset of node and edge types and their properties are considered in a given model. Such an approach is helpful for scalability, and also to help preserve detailed information (of particular importance in a security context) – as clustering entities in a more complex and larger graph could result in less useful results. In particular, such an approach allows for different types of relationships between entities to be preserved/more easily analyzed.

[00246] While GBM 154 can be used with different models corresponding to different subgraphs, core abstractions remain the same across types of models.

[00247] For example, each node type in a GBM model is considered to belong to a class. The class can be thought of as a way for the GBM to split nodes based on the criteria it uses for the model. The class for a node is represented as a string whose value is derived from the node's key and properties depending on the GBM Model. Note that different GBM models may create different class values for the same node. For each node type in a given GBM model, GBM 154 can generate clusters of nodes for that type. A GBM generated cluster for a given member node type cannot span more than one class for that node type. GBM 154 generates edges between clusters that have the same types as the edges between source and destination cluster node types.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00173

[00248] Additionally or alternatively, the processes described herein as being used for a particular model can be used (can be the same) across models, and different models can also be configured with different settings.

[00249] Additionally or alternatively, the node types and the edge types may correspond to existing types in the common graph node and edge tables but this is not necessary. Even when there is a correspondence, the properties provided to GBM 154 are not limited to the properties that are stored in the corresponding graph table entries. They can be enriched with additional information before being passed to GBM 154.

[00250] Logically, the input for a GBM model can be characterized in a manner that is similar to other graphs. Edge triplets can be expressed, for example, as an array of source node type, edge type, and destination node type. And, each node type is associated with node properties, and each edge type is associated with edge properties. Other edge triplets can also be used (and/or edge triplets can be extended) in accordance with various embodiments.

[00251] Note that the physical input to the GBM model need not (and does not, in various embodiments) conform to the logical input. For example, the edges in the PtypeConn model correspond to edges between Matching Neighbors (MN) clusters, where each process node has an MN cluster identifier property. In the User ID to User ID model (also referred to herein as the Uid2Uid model), edges are not explicitly provided separately from nodes (as the euid array in the node properties serves the same purpose). In both cases, however, the physical information provides the applicable information necessary for the logical input.

[00252] The state input for a particular GBM model can be stored in a file, a database, or other appropriate storage. The state file (from a previous run) is provided, along with graph data, except for when the first run for a given model is performed, or the model is reset. In some cases, no data may be available for a particular model in a given time period, and GBM may not be run for that time period. As data becomes available at a future time, GBM can run using the latest state file as input.

[00253] GBM 154 outputs cluster nodes, cluster membership edges, and inter-cluster relationship edges that are stored (in some embodiments) in the graph node tables: node_c, node_cm, and node_icr, respectively. The type names of nodes and edges may conform to the following rules:

[00254] * A given node type can be used in multiple different GBM models. The type names of the cluster nodes generated by two such models for that node type will be different. For instance, process type nodes will appear in both PtypeConn and Uid2Uid models, but their cluster nodes will have different type names.

[00255] * The membership edge type name is "MemberOf."

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00174

[00256] * The edge type names for cluster-to-cluster edges will be the same as the edge type names in the underlying node-to-node edges in the input.

[00257] The following are example events GBM 154 can generate: new class, new cluster, new edge from class to class, split class (the notion that GBM 154 considers all nodes of a given type and class to be in the same cluster initially and if GBM 154 splits them into multiple clusters, it is splitting a class), new edge from cluster and class, new edge between cluster and cluster, and/or new edge from class to cluster.

[00258] One underlying node or edge in the logical input can cause multiple types of events to be generated. Conversely, one event can correspond to multiple nodes or edges in the input. Not every model generates every event type.

[00259] Additional information regarding examples of data structures/models that can be used in conjunction with models used by data platform 12 is now provided.

[00260] In some examples, a PTypeConn Model clusters nodes of the same class that have similar connectivity relationships. For example, if two processes had similar incoming neighbors of the same class and outgoing neighbors of the same class, they could be clustered.

[00261] The node input to the PTypeConn model for a given time period includes non-interactive (i.e., not associated with tty) process nodes that had connections in the time period and the base graph nodes of other types (IP Service Endpoint (IPSep) comprising an IP address and a port), DNS Service Endpoint (DNSSep) and IPAddress) that have been involved in those connections. The base relationship is the connectivity relationship for the following type triplets:

[00262] * Process, ConnectedTo, Process

[00263] * Process, ConnectedTo, IP Service Endpoint (IPSep)

[00264] * Process, ConnectedTo, DNS Service Endpoint (DNSSep)

[00265] * IPAddress, ConnectedTo, ProcessProcess, DNS, ConnectedTo, Process

[00266] The edge inputs to this model are the ConnectedTo edges from the MN cluster, instead of individual node-to-node ConnectedTo edges from the base graph. The membership edges created by this model refer to the base graph node type provided in the input.

[00267] Class Values:

[00268] The class values of nodes are determined as follows depending on the node type (e.g., Process nodes, IPSep nodes, DNSSep nodes, and IP Address nodes).

[00269] Process nodes:

[00270] if exe_path contains java then "java <cmdline_term_1> ..."

[00271] else if exe_path contains python then "python <cmdline_term_1> ..."

[00272] else "last_part_of_exe_path"

[00273] IPSep nodes:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00175

[00274] if IP_internal then "IntIPS"

[00275] else if severity = 0 then "<IP_addr>:<protocol>:<port>"

[00276] else "<IP_addr>:<port>_BadIP"

[00277] DNSSep nodes:

[00278] if IP_internal = 1 then "<hostname>"

[00279] else if severity = 0 then "<hostname>:<protocol>:port"

[00280] else "<hostname>:<port>_BadIP"

[00281] IPAddress nodes (will appear only on client side):

[00282] if IP_internal = 1 then "IPIntC"

[00283] else if severity = 0 then "ExtIPC"

[00284] else "ExtBadIPC"

[00285] Events:

[00286] A new class event in this model for a process node is equivalent to seeing a new CType being involved in a connection for the first time. Note that this does not mean the CType was not seen before. It is possible that it was previously seen but did not make a connection at that time.

[00287] A new class event in this model for an IPSep node with IP_internal = 0 is equivalent to seeing a connection to a new external IP address for the first time.

[00288] A new class event in this model for a DNSSep node is equivalent to seeing a connection to a new domain for the first time.

[00289] A new class event in this model for an IPAddress node with IP_internal = 0 and severity = 0 is equivalent to seeing a connection from any external IP address for the first time.

[00290] A new class event in this model for an IPAddress node with IP_internal = 0 and severity > 0 is equivalent to seeing a connection from any bad external IP address for the first time.

[00291] A new class to class to edge from a class for a process node to a class for a process node is equivalent to seeing a communication from the source CType making a connection to the destination CType for the first time.

[00292] A new class to class to edge from a class for a process node to a class for a DNSSep node is equivalent to seeing a communication from the source CType making a connection to the destination domain name for the first time.

[00293] An IntPConn Model may be similar to the PtypeConn Model, except that connection edges between parent/child processes and connections between processes where both sides are not interactive are filtered out.

[00294] A Uid2Uid Model may cluster processes with the same username that show similar privilege change behavior. For instance, if two processes with the same username had similar

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00176

effective user values, launched processes with similar usernames, and were launched by processes with similar usernames, then they could be clustered.

[00295] An edge between a source cluster and destination cluster generated by this model means that all of the processes in the source cluster had a privilege change relationship to at least one process in the destination cluster.

[00296] The node input to this model for a given time period includes process nodes that are running in that period. The value of a class of process nodes is "<username>".

[00297] The base relationship that is used for clustering is privilege change, either by the process changing its effective user ID, or by launching a child process which runs with a different user.

[00298] The physical input for this model includes process nodes (only), with the caveat that the complete ancestor hierarchy of process nodes active (i.e., running) for a given time period is provided as input even if an ancestor is not active in that time period. Note that effective user IDs of a process are represented as an array in the process node properties, and launch relationships are available from ppid_hash fields in the properties as well.

[00299] A new class event in this model is equivalent to seeing a user for the first time.

[00300] A new class to class edge event is equivalent to seeing the source user making a privilege change to the destination user for the first time.

[00301] A Ct2Ct Model may cluster processes with the same CType that show similar launch behavior. For instance, if two processes with the same CType have launched processes with similar CTypes, then they could be clustered.

[00302] The node input to this model for a given time period includes process nodes that are running in that period. The value class of process nodes is CType (similar to how it is created for the PtypeConn Model).

[00303] The base relationship that is used for clustering is a parent process with a given CType launching a child process with another given destination CType.

[00304] The physical input for this model includes process nodes (only) with the caveat that the complete ancestor hierarchy active process nodes (i.e., that are running) for a given time period is provided as input even if an ancestor is not active in that time period. Note that launch relationships are available from ppid_hash fields in the process node properties.

[00305] An edge between a source cluster and destination cluster generated by this model means that all of the processes in the source cluster launched at least one process in the destination cluster.

[00306] A new class event in this model is equivalent to seeing a CType for the first time. Note that the same type of event will be generated by the PtypeConn Model as well.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00177

[00307] A new class to class edge event is equivalent to seeing the source CType launching the destination CType for the first time.

[00308] An MTypeConn Model may cluster nodes of the same class that have similar connectivity relationships. For example, if two machines had similar incoming neighbors of the same class and outgoing neighbors of the same class, they could be clustered.

[00309] A new class event in this model will be generated for external IP addresses or (as applicable) domain names seen for the first time. Note that a new class to class to edge Machine, class to class for an IPSep or DNSName node will also be generated at the same time.

[00310] The membership edges generated by this model will refer to Machine, IPAddress, DNSName, and IPSep nodes in the base graph. Though the nodes provided to this model are IPAddress nodes instead of IPSep nodes, the membership edges it generates will refer to IPSep type nodes. Alternatively, the base graph can generate edges between Machine and IPSep node types. Note that the Machine to IPAddress edges have tcp_dst_ports/udp_dst_ports properties that can be used for this purpose.

[00311] The node input to this model for a given time period includes machine nodes that had connections in the time period and the base graph nodes of other types (IPAddress and DNSName) that were involved in those connections.

[00312] The base relationship is the connectivity relationship for the following type triplets:

[00313] * Machine, ConnectedTo, Machine

[00314] * Machine, ConnectedTo, IPAddress

[00315] * Machine, ConnectedTo, DNSName

[00316] * IPAddress, ConnectedTo, Machine, DNS, ConnectedTo, Machine

[00317] The edge inputs to this model are the corresponding ConnectedTo edges in the base graph.

[00318] Class Values:

[00319] Machine:

[00320] The class value for all Machine nodes is "Machine."

[00321] The machine_terms property in the Machine nodes is used, in various embodiments, for labeling machines that are clustered together. If a majority of the machines clustered together share a term in the machine_terms, that term can be used for labeling the cluster.

[00322] IPSep:

[00323] The class value for IPSep nodes is determined as follows:

[00324] if IP_internal then "IntIPS"

[00325] else

[00326] if severity = 0 then "<ip_addr>:<protocol>:<port>"

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00178

[00327] else "<IP_addr_BadIP>"

[00328] IPAddress:

[00329] The class value for IpAddress nodes is determined as follows:

[00330] if IP_internal then "IntIPC"

[00331] else

[00332] if severity = 0 then "ExtIPC"

[00333] else "ExtBadIPC"

[00334] DNSName:

[00335] The class value for DNSName nodes is determined as follows:

[00336] if severity = 0 then "<hostname>"

[00337] else then "<hostname>_BadIP"

[00338] An example structure for a New Class Event is now described.

[00339] The key field for this event type looks as follows (using the PtypeConn model as an example):

[00340] {

[00341] "node": {

[00342] "class": {

[00343] "cid": "httpd"

[00344] },

[00345] "key": {

[00346] "cid": "29654"

[00347] },

[00348] "type": "PtypeConn"

[00349] }

[00350] }

[00351] It contains the class value and also the ID of the cluster where that class value is observed. Multiple clusters can be observed with the same value in a given time period. It contains the class value and also the ID of the cluster where that class value is observed. Multiple clusters can be observed with the same value in a given time period. Accordingly, in some embodiments, GBM 154 generates multiple events of this type for the same class value.

[00352] The properties field looks as follows:

[00353] {

[00354] "set_size": 5

[00355] }

[00356] The set_size indicates the size of the cluster referenced in the keys field.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00179

[00357] Conditions:

[00358] For a given model and time period, multiple NewClass events can be generated if there is more than one cluster in that class. NewNode events will not be generated separately in this case.

[00359] Example New Class to Class Edge Event structure:

[00360] The key field for this event type looks as follows (using the PtypeConn model as an example):

[00361] "edge": {

[00362] "dst_node": {

[00363] "class": {

[00364] "cid": "java war"

[00365] },

[00366] "key": {

[00367] "cid": "27635"

[00368] },

[00369] "type": "PtypeConn"

[00370] },

[00371] "src_node": {

[00372] "class": {

[00373] "cid": "IntIPC"

[00374] },

[00375] "key": {

[00376] "cid": "20881"

[00377] },

[00378] "type": "PtypeConn"

[00379] },

[00380] "type": "ConnectedTo"

[00381] }

[00382] }

[00383] The key field contains source and destination class values and also source and destination cluster identifiers (i.e., the src/dst_node:key.cid represents the src/dst cluster identifier).

[00384] In a given time period for a given model, an event of this type could involve multiple edges between different cluster pairs that have the same source and destination class values.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00180

GBM 154 can generate multiple events in this case with different source and destination cluster identifiers.

[00385] The props fields look as follows for this event type:

[00386] {

[00387] "dst_set_size": 2,

[00388] "src_set_size": 1

[00389] }

[00390] The source and destination sizes represent the sizes of the clusters given in the keys field.

[00391] Conditions:

[00392] For a given model and time period, multiple NewClassToClass events can be generated if there are more than one pair of clusters in that class pair. NewNodeToNode events are not generated separately in this case.

[00393] Combining Events at the Class Level: for a given model and time period, the following example types of events can represent multiple changes in the underlying GBM cluster level graph in terms of multiple new clusters or multiple new edges between clusters:

[00394] * NewClass

[00395] * NewEdgeClassToClass

[00396] * NewEdgeNodeToClass

[00397] * NewEdgeClassToNode

[00398] Multiple NewClass events with the same model and class can be output if there are multiple clusters in that new class.

[00399] Multiple NewEdgeClassToClass events with the same model and class pair can be output if there are multiple new cluster edges within that class pair.

[00400] Multiple NewEdgeNodeToClass events with the same model and destination class can be output if there are multiple new edges from the source cluster to the destination clusters in that destination class (the first time seeing this class as a destination cluster class for the source cluster).

[00401] Multiple NewEdgeClassToNode events with the same model and source class can be output if there are multiple new edges from source clusters to the destination clusters in that source class (the first time seeing this class as a source cluster class for the destination cluster).

[00402] These events may be combined at the class level and treated as a single event when it is desirable to view changes at the class level, e.g., when one wants to know when there is a new CType.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00181

[00403] In some examples, different models may have partial overlap in the types of nodes they use from the base graph. Therefore, they can generate NewClass type events for the same class. NewClass events can also be combined across models when it is desirable to view changes at the class level.

[00404] Using techniques herein, actions can be associated with processes and (e.g., by associating processes with users) actions can thus also be associated with extended user sessions. Such information can be used to track user behavior correctly, even where a malicious user attempts to hide his trail by changing user identities (e.g., through lateral movement). Extended user session tracking can also be useful in operational use cases without malicious intent, e.g., where users make original logins with distinct usernames (e.g., "charlie" or "dave") but then perform actions under a common username (e.g., "admin" or "support"). One such example is where multiple users with administrator privileges exist, and they need to gain superuser privilege to perform a particular type of maintenance. It may be desirable to know which operations are performed (as the superuser) by which original user when debugging issues. In the following examples describing extended user session tracking, reference is generally made to using the secure shell (ssh) protocol as implemented by openssh (on the server side) as the mechanism for logins. However, extended user session tracking is not limited to the ssh protocol or a particular limitation and the techniques described herein can be extended to other login mechanisms.

[00405] On any given machine, there will be a process that listens for and accepts ssh connections on a given port. This process can run the openssh server program running in daemon mode or it could be running another program (e.g., initd on a Linux system). In either case, a new process running openssh will be created for every new ssh login session and this process can be used to identify an ssh session on that machine. This process is called the "privileged" process in openssh.

[00406] After authentication of the ssh session, when an ssh client requests a shell or any other program to be run under that ssh session, a new process that runs that program will be created under (i.e., as a child of) the associated privileged process. If an ssh client requests port forwarding to be performed, the connections will be associated with the privileged process.

[00407] In modern operating systems such as Linux and Windows, each process has a parent process (except for the very first process) and when a new process is created the parent process is known. By tracking the parent-child hierarchy of processes, one can determine if a particular process is a descendant of a privileged openssh process and thus if it is associated with an ssh login session.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00182

[00408] For user session tracking across machines (or on a single machine with multiple logins) in a distributed environment, it is established when two login sessions have a parent-child relationship. After that, the "original" login session, if any, for any given login session can be determined by following the parent relationship recursively.

[00409] Fig. 3F is a representation of a user logging into a first machine and then into a second machine from the first machine, as well as information associated with such actions. In the example of Fig. 3F, a user, Charlie, logs into Machine A (331) from a first IP address (332). As part of the login process, he provides a username (333). Once connected to Machine A, an openssh privileged process (334) is created to handle the connection for the user, and a terminal session is created and a bash process (335) is created as a child. Charlie launches an ssh client (336) from the shell, and uses it to connect (337) to Machine B (338). As with the connection he makes to Machine A, Charlie's connection to Machine B will have an associated incoming IP address (339), in this case, the IP address of Machine A. And, as part of the login process with Machine B, Charlie will provide a username (340) which need not be the same as username 333. An openssh privileged process (341) is created to handle the connection, and a terminal session and child bash process (342) will be created. From the command line of Machine B, Charlie launches a curl command (343), which opens an HTTP connection (2028) to an external Machine C (345).

[00410] Fig. 3G is an alternate representation of actions occurring in Fig. 3F, where events occurring on Machine A are indicated along line 350, and events occurring on Machine B are indicated along line 351. As shown in Fig. 3G, an incoming ssh connection is received at Machine A (352). Charlie logs in (as user "x") and an ssh privileged process is created to handle Charlie's connection (353). A terminal session is created and a bash process is created (354) as a child of process 353. Charlie wants to ssh to Machine B, and so executes an ssh client on Machine A (355), providing credentials (as user "y") at 356. Charlie logs into Machine B, and an ssh privileged process is created to handle Charlie's connection (357). A terminal session is created and a bash process is created (358) as a child of process 357. Charlie then executes curl (359) to download content from an external domain (via connection 360).

[00411] The external domain could be a malicious domain, or it could be benign. Suppose the external domain is malicious (and, e.g., Charlie has malicious intent). It would be advantageous (e.g., for security reasons) to be able to trace the contact with the external domain back to Machine A, and then back to Charlie's IP address. Using techniques described herein (e.g., by correlating process information collected by various agents), such tracking of Charlie's activities back to his original login (330) can be accomplished. In particular, an extended user session can

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00183

be tracked that associates Charlie's ssh processes together with a single original login and thus original user.

[00412] As described herein, software agents (such as agent 112) may run on machines (such as a machine that implements one of nodes 116) and detect new connections, processes, and/or logins. As also previously explained, such agents send associated records to data platform 12 which includes one or more datastores (e.g., data store 30) for persistently storing such data. Such data can be modeled using logical tables, also persisted in datastores (e.g., in a relational database that provides an SQL interface), allowing for querying of the data. Other datastores such as graph oriented databases and/or hybrid schemes can also be used.

[00413] The following identifiers are commonly used in the tables:

[00414] * MID

[00415] * PID_hash

[00416] An ssh login session can be identified uniquely by an (MID, PID_hash) tuple. The MID is a machine identifier that is unique to each machine, whether physical or virtual, across time and space. Operating systems use numbers called process identifiers (PIDs) to identify processes running at a given time. Over time processes may die and new processes may be started on a machine or the machine itself may restart. The PID is not necessarily unique across time in that the same PID value can be reused for different processes at different times. In order to track process descendants across time, one should therefore account for time as well. In order to be able to identify a process on a machine uniquely across time, another number called a PID_hash is generated for the process. In various embodiments, the PID_hash is generated using a collision-resistant hash function that takes the PID, start time, and (in various embodiments, as applicable) other properties of a process.

[00417] Input data collected by agents comprises the input data model and is represented by the following logical tables:

[00418] * connections

[00419] * processes

[00420] * logins

[00421] A connections table may maintain records of TCP/IP connections observed on each machine. Example columns included in a connections table are as follows:

Column Name	Description
MID	Identifier of the machine that the connection was observed on.
start_time	Connection start time.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00184

PID_hash	Identifier of the process that was associated with the connection.
src_IP_addr	Source IP address (the connection was initiated from this IP address).
src_port	Source port.
dst_IP_addr	Destination IP address (the connection was made to this IP address).
dst_port	Destination port.
Prot	Protocol (TCP or UDP).
Dir	Direction of the connection (incoming or outgoing) with respect to this machine.

[00422] The source fields (IP address and port) correspond to the side from which the connection was initiated. On the destination side, the agent associates an ssh connection with the privileged ssh process that is created for that connection.

[00423] For each connection in the system, there will be two records in the table, assuming that the machines on both sides of the connection capture the connection. These records can be matched based on equality of the tuple (src_IP_addr, src_port, dst_IP_addr, dst_port, Prot) and proximity of the start_time fields (e.g., with a one minute upper threshold between the start_time fields).

[00424] A processes table maintains records of processes observed on each machine. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the process was observed on.
PID_hash	Identifier of the process.
start_time	Start time of the process.
exe_path	The executable path of the process.
PPID_hash	Identifier of the parent process.

[00425] A logins table may maintain records of logins to machines. It may have the following columns:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00185

Column Name	Description
MID	Identifier of the machine that the login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with login.
login_time	Time of login.
login_username	Username used in login.

[00426] Output data generated by session tracking is represented with the following logical tables:

[00427] * login-local-descendant

[00428] * login-connection

[00429] * login-lineage

[00430] Using data in these tables, it is possible to determine descendant processes of a given ssh login session across the environment (i.e., spanning machines). Conversely, given a process, it is possible to determine if it is an ssh login descendant as well as the original ssh login session for it if so.

[00431] A login-local-descendant table maintains the local (i.e., on the same machine) descendant processes of each ssh login session. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with login.
login_time	Time of login.
login_username	Username used in login.

[00432] A login-connections table may maintain the connections associated with ssh logins. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the process was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with the login.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00186

login_time	Time of login.
login_username	The username used in the login.
src_IP_addr	Source IP address (connection was initiated from this IP address).
src_port	Source port.
dst_IP_addr	Destination IP address (connection was made to this IP address).
dst_port	Destination port.

[00433] A login-lineage table may maintain the lineage of ssh login sessions. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the ssh login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with the login.
parent_MID	Identifier of the machine that the parent ssh login was observed on.
parent_sshd_PID_hash	Identifier of the sshd privileged process associated with the parent login.
origin_MID	Identifier of the machine that the origin ssh login was observed on.
origin_sshd_PID_hash	Identifier of the sshd privileged process associated with the origin login.

[00434] The parent_MID and parent_sshd_PID_hash columns can be null if there is no parent ssh login. In that case, the (MID, sshd_PID_hash) tuple will be the same as the (origin_MID, origin_sshd_PID_hash) tuple.

[00435] Fig. 3H illustrates an example of a process for performing extended user tracking. In various embodiments, process 361 is performed by data platform 12. The process begins at 362 when data associated with activities occurring in a network environment (such as entity A's datacenter) is received. One example of such data that can be received at 362 is agent-collected data described above (e.g., in conjunction with process 200). At 363, the received network activity is used to identify user login activity. And, at 364, a logical graph that links the user

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00187

login activity to at least one user and at least one process is generated (or updated, as applicable). Additional detail regarding process 361, and in particular, portions 363 and 364 of process 361 are described in more detail below (e.g., in conjunction with discussion of Fig. 3J).

[00436] Fig. 3I depicts a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection. The scenario depicted in Fig. 3I is used to describe an example of processing that can be performed on data collected by agents to generate extended user session tracking information. Fig. 3I is an alternate depiction of the information shown in Figs. 3F and 3G.

[00437] At time t1 (365), a first ssh connection is made to Machine A (366) from an external source (367) by a user having a username of "X." In the following example, suppose the external source has an IP address of 1.1.1.10 and uses source port 10000 to connect to Machine A (which has an IP address of 2.2.2.20 and a destination port 22). External source 367 is considered an external source because its IP address is outside of the environment being monitored (e.g., is a node outside of entity A's datacenter, connecting to a node inside of entity A's datacenter).

[00438] A first ssh login session LS1 is created on machine A for user X. The privileged openssh process for this login is A1 (368). Under the login session LS1, the user creates a bash shell process with PID_hash A2 (369).

[00439] At time t2 (370), inside the bash shell process A2, the user runs an ssh program under a new process A3 (371) to log in to machine B (372) with a different username ("Y"). In particular, an ssh connection is made from source IP address 2.2.2.20 and source port 10001 (Machine A's source information) to destination IP address 2.2.2.21 and destination port 22 (Machine B's destination information).

[00440] A second ssh login session LS2 is created on machine B for user Y. The privileged openssh process for this login is B1 (373). Under the login session LS2, the user creates a bash shell process with PID_hash B2 (374).

[00441] At time t3 (376), inside the bash shell process B2, the user runs a curl command under a new process B3 (377) to download a file from an external destination (378). In particular, an HTTPS connection is made from source IP address 2.2.2.21 and source port 10002 (Machine B's source information) to external destination IP address 3.3.3.30 and destination port 443 (the external destination's information).

[00442] Using techniques described herein, it is possible to determine the original user who initiated the connection to external destination 378, which in this example is a user having the username X on machine A (where the extended user session can be determined to start with ssh login session LS1).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00188

[00443] Based on local descendant tracking, the following determinations can be on machine A and B without yet having performed additional processing (described in more detail below):

[00444] * A3 is a descendant of A1 and thus associated with LS1.

[00445] * The connection to the external domain from machine B is initiated by B3.

[00446] * B3 is a descendant of B1 and is thus associated with LS2.

[00447] * Connection to the external domain is thus associated with LS2.

[00448] An association between A3 and LS2 can be established based on the fact that LS2 was created based on an ssh connection initiated from A3. Accordingly, it can be determined that LS2 is a child of LS1.

[00449] To determine the user responsible for making the connection to the external destination (e.g., if it were a known bad destination), first, the process that made the connection would be traced, i.e., from B3 to LS2. Then LS2 would be traced to LS1 (i.e., LS1 is the origin login session for LS2). Thus the user for this connection is the user for LS1, i.e., X. As represented in Fig. 3I, one can visualize the tracing by following the links (in the reverse direction of arrows) from external destination 378 to A1 (368).

[00450] In the example scenario, it is assumed that both ssh connections occur in the same analysis period. However, the approaches described herein will also work for connections and processes that are created in different time periods.

[00451] Fig. 3J illustrates an example of a process for performing extended user tracking. In various embodiments, process 380 is performed periodically (e.g., once an hour in a batch fashion) by ssh tracker 148 to generate new output data. In general, batch processing allows for efficient analysis of large volumes of data. However, the approach can be adapted, as applicable, to process input data on a record-by-record fashion while maintaining the same logical data processing flow. As applicable the results of a given portion of process 380 are stored for use in a subsequent portion.

[00452] The process begins at 381 when new ssh connection records are identified. In particular, new ssh connections started during the current time period are identified by querying the connections table. The query uses filters on the start_time and dst_port columns. The values of the range filter on the start_time column are based on the current time period. The dst_port column is checked against ssh listening port(s). By default, the ssh listening port number is 22. However, as this could vary across environments, the port(s) that openssh servers are listening to in the environment can be determined by data collection agents dynamically and used as the filter value for the dst_port as applicable. In the scenario depicted in Fig. 3I, the query result will generate the records shown in Fig. 3K. Note that for the connection between machine A and B, the two machines are likely to report start_time values that are not exactly the same but close

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00189

enough to be considered matching (e.g., within one minute or another appropriate amount of time). In the above table, they are shown to be the same for simplicity.

[00453] At 382, ssh connection records reported from source and destination sides of the same connection are matched. The ssh connection records (e.g., returned from the query at 381) are matched based on the following criteria:

[00454] * The five tuples (src_IP, dst_IP, IP_prot, src_port, dst_port) of the connection records must match.

[00455] * The delta between the start times of the connections must be within a limit that would account for the worst case clock difference expected between two machines in the environment and typical connection setup latency.

[00456] * If there are multiple matches possible, then the match with the smallest time delta is chosen.

[00457] Note that record 390 from machine A for the incoming connection from the external source cannot be matched with another record as there is an agent only on the destination side for this connection. Example output of portion 382 of process 380 is shown in Fig. 3L. The values in the dst_PID_hash column (391) are that of the sshd privileged process associated with ssh logins.

[00458] At 383, new logins during the current time period are identified by querying the logins table. The query uses a range filter on the login_time column with values based on the current time period. In the example depicted in Fig. 3I, the query result will generate the records depicted in Fig. 3M.

[00459] At 384, matched ssh connection records created at 382 and new login records created at 383 are joined to create new records that will eventually be stored in the login-connection table. The join condition is that dst_MID of the matched connection record is equal to the MID of the login record and the dst_PID_hash of the matched connection record is equal to the sshd_PID_hash of the login record. In the example depicted in Fig. 3I, the processing performed at 384 will generate the records depicted in Fig. 3N.

[00460] At 385, login-local-descendant records in the lookback time period are identified. It is possible that a process that is created in a previous time period makes an ssh connection in the current analysis batch period. Although not depicted in the example illustrated in Fig. 3I, consider a case where bash process A2 does not create ssh process A3 right away but instead that the ssh connection A3 later makes to machine B is processed in a subsequent time period than the one where A2 was processed. While processing this subsequent time period in which processes A3 and B1 are seen, knowledge of A2 would be useful in establishing that B1 is associated with A3 (via ssh connection) which is associated with A2 (via process parentage) which in turn would be useful in establishing that the parent of the second ssh login is the first ssh login. The time

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00190

period for which look back is performed can be limited to reduce the amount of historical data that is considered. However, this is not a requirement (and the amount of look back can be determined, e.g., based on available processing resources). The login local descendants in the lookback time period can be identified by querying the login-local-descendant table. The query uses a range filter on the login_time column where the range is from start_time_of_current_period - lookback_time to start_time_of_current_period. (No records as a result of performing 385 on the scenario depicted in Fig. 3I are obtained, as only a single time period is applicable in the example scenario.)

[00461] At 386, new processes that are started in the current time period are identified by querying the processes table. The query uses a range filter on the start_time column with values based on the current time period. In the example depicted in Fig. 3I, the processing performed at 386 will generate the records depicted in Fig. 3O.

[00462] At 387, new login-local-descendant records are identified. The purpose is to determine whether any of the new processes in the current time period are descendants of an ssh login process and if so to create records that will be stored in the login-local-descendant table for them. In order to do so, the parent-child relationships between the processes are recursively followed. Either a top down or bottom up approach can be used. In a top down approach, the ssh local descendants in the lookback period identified at 385, along with new ssh login processes in the current period identified at 384 are considered as possible ancestors for the new processes in the current period identified at 386.

[00463] Conceptually, the recursive approach can be considered to include multiple sub-steps where new processes that are identified to be ssh local descendants in the current sub-step are considered as ancestors for the next step. In the example scenario depicted in Fig. 3I, the following descendancy relationships will be established in two sub-steps:

[00464] Sub-step 1:

[00465] Process A2 is a local descendant of LS1 (i.e., MID = A, sshd_PID_hash = A1) because it is a child of process A1 which is the login process for LS1.

[00466] Process B2 is a local descendant of LS2 (i.e., MID = B, sshd_PID_hash = B1) because it is a child of process B1 which is the login process for LS2.

[00467] Sub-step 2:

[00468] Process A3 is a local descendant of LS1 because it is a child of process A2 which is associated to LS1 in sub-step 1.

[00469] Process B3 is a local descendant of LS2 because it is a child of process B1 which is associated to LS2 in sub-step 1.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00191

[00470] Implementation portion 387 can use a datastore that supports recursive query capabilities, or, queries can be constructed to process multiple conceptual sub-steps at once. In the example depicted in Fig. 3I, the processing performed at 387 will generate the records depicted in Fig. 3P. Note that the ssh privileged processes associated with the logins are also included as they are part of the login session.

[00471] At 388, the lineage of new ssh logins created in the current time period is determined by associating their ssh connections to source processes that may be descendants of other ssh logins (which may have been created in the current period or previous time periods). In order to do so, first an attempt is made to join the new ssh login connections in the current period (identified at 384) with the combination of the login local descendants in the lookback period (identified at 385) and the login local descendants in the current time period (identified at 386). This will create adjacency relationships between child and parent logins. In the example depicted in Fig. 3I, the second ssh login connection will be associated with process A3 and an adjacency relationship between the two login sessions will be created (as illustrated in Fig. 3Q).

[00472] Next, the adjacency relationships are used to find the original login sessions. While not shown in the sample scenario, there could be multiple ssh logins in a chain in the current time period, in which case a recursive approach (as in 387) could be used. At the conclusion of portion 388, the login lineage records depicted in Fig. 3R will be generated.

[00473] Finally, at 389, output data is generated. In particular, the new login-connection, login-local-descendant, and login-lineage records generated at 384, 387, and 388 are inserted into their respective output tables (e.g., in a transaction manner).

[00474] An alternate approach to matching TCP connections between machines running an agent is for the client to generate a connection GUID and send it in the connection request (e.g., the SYN packet) it sends and for the server to extract the GUID from the request. If two connection records from two machines have the same GUID, they are for the same connection. Both the client and server will store the GUID (if it exists) in the connection records they maintain and report. On the client side, the agent can configure the network stack (e.g. using IP tables functionality on Linux) to intercept an outgoing TCP SYN packet and modify it to add the generated GUID as a TCP option. On the server side, the agent already extracts TCP SYN packets and thus can look for this option and extract the GUID if it exists.

[00475] Example graph-based user tracking and threat detection embodiments associated with data platform 12 will now be described. Administrators and other users of network environments (e.g., entity A's datacenter 104) often change roles to perform tasks. As one example, suppose that at the start of a workday, an administrator (hereinafter "Joe Smith") logs in to a console, using an individualized account (e.g., username=joe.smith). Joe performs various tasks as

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00192

himself (e.g., answering emails, generating status reports, writing code, etc.). For other tasks (e.g., performing updates), Joe may require different/additional permission than his individual account has (e.g., root privileges). One way Joe can gain access to such permissions is by using sudo, which will allow Joe to run a single command with root privileges. Another way Joe can gain access to such permissions is by su or otherwise logging into a shell as root. After gaining root privileges, another thing that Joe can do is switch identities. As one example, to perform administrative tasks, Joe may use "su help" or "su database-admin" to become (respectively) the help user or the database-admin user on a system. He may also connect from one machine to another, potentially changing identities along the way (e.g., logging in as joe.smith at a first console, and connecting to a database server as database-admin). When he's completed various administrative tasks, Joe can relinquish his root privileges by closing out of any additional shells created, reverting back to a shell created for user joe.smith.

[00476] While there are many legitimate reasons for Joe to change his identity throughout the day, such changes may also correspond to nefarious activity. Joe himself may be nefarious, or Joe's account (joe.smith) may have been compromised by a third party (whether an "outsider" outside of entity A's network, or an "insider"). Using techniques described herein, the behavior of users of the environment can be tracked (including across multiple accounts and/or multiple machines) and modeled (e.g., using various graphs described herein). Such models can be used to generate alerts (e.g., to anomalous user behavior). Such models can also be used forensically, e.g., helping an investigator visualize various aspects of a network and activities that have occurred, and to attribute particular types of actions (e.g., network connections or file accesses) to specific users.

[00477] In a typical day in a datacenter, a user (e.g., Joe Smith) will log in, run various processes, and (optionally) log out. The user will typically log in from the same set of IP addresses, from IP addresses within the same geographical area (e.g., city or country), or from historically known IP addresses/geographical areas (i.e., ones the user has previously/occasionally used). A deviation from the user's typical (or historical) behavior indicates a change in login behavior. However, it does not necessarily mean that a breach has occurred. Once logged into a datacenter, a user may take a variety of actions. As a first example, a user might execute a binary/script. Such binary/script might communicate with other nodes in the datacenter, or outside of the datacenter, and transfer data to the user (e.g., executing "curl" to obtain data from a service external to the datacenter). As a second example, the user can similarly transfer data (e.g., out of the datacenter), such as by using POST. As a third example, a user might change privilege (one or more times), at which point the user can send/receive data as per

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00193

above. As a fourth example, a user might connect to a different machine within the datacenter (one or more times), at which point the user can send/receive data as per the above.

[00478] In various embodiments, the above information associated with user behavior is broken into four tiers. The tiers represent example types of information that data platform 12 can use in modeling user behavior:

[00479] 1. The user's entry point (e.g., domains, IP addresses, and/or geolocation information such as country/city) from which a user logs in.

[00480] 2. The login user and machine class.

[00481] 3. Binaries, executables, processes, etc. a user launches.

[00482] 4. Internal servers with which the user (or any of the user's processes, child processes, etc.) communicates, and external contacts (e.g., domains, IP addresses, and/or geolocation information such as country/city) with which the user communicates (i.e., transfers data).

[00483] In the event of a security breach, being able to concretely answer questions about such information can be very important. And, collectively, such information is useful in providing an end-to-end path (e.g., for performing investigations).

[00484] In the following example, suppose a user ("UserA") logs into a machine ("Machine01") from a first IP address ("IP01"). Machine01 is inside a datacenter. UserA then launches a script ("runnable.sh") on Machine01. From Machine01, UserA next logs into a second machine ("Machine02") via ssh, also as UserA, also within the datacenter. On Machine02, UserA again launches a script ("new_runnable.sh"). On Machine02, UserA then changes privilege, becoming root on Machine02. From Machine02, UserA (now as root) logs into a third machine ("Machine03") in the datacenter via ssh, as root on Machine03. As root on Machine03, the user executes a script ("collect_data.sh") on Machine03. The script internally communicates (as root) to a MySQL-based service internal to the datacenter, and downloads data from the MySQL-based service. Finally, as root on Machine03, the user externally communicates with a server outside the datacenter ("External01"), using a POST command. To summarize what has occurred, in this example, the source/entry point is IP01. Data is transferred to an external server External01. The machine performing the transfer to External01 is Machine03. The user transferring the data is "root" (on Machine03), while the actual user (hiding behind root) is UserA.

[00485] In the above scenario, the "original user" (ultimately responsible for transmitting data to External01) is UserA, who logged in from IP01. Each of the processes ultimately started by UserA, whether started at the command line (tty) such as "runnable.sh" or started after an ssh connection such as "new_runnable.sh," and whether as UserA, or as a subsequent identity, are all examples of child processes which can be arranged into a process hierarchy.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00194

[00486] As previously mentioned, machines can be clustered together logically into machine clusters. One approach to clustering is to classify machines based on information such as the types of services they provide/binaries they have installed upon them/processes they execute. Machines sharing a given machine class (as they share common binaries/services/etc.) will behave similarly to one another. Each machine in a datacenter can be assigned to a machine cluster, and each machine cluster can be assigned an identifier (also referred to herein as a machine class). One or more tags can also be assigned to a given machine class (e.g., database_servers_west or prod_web_frontend). One approach to assigning a tag to a machine class is to apply term frequency analysis (e.g., TF/IDF) to the applications run by a given machine class, selecting as tags those most unique to the class. Data platform 12 can use behavioral baselines taken for a class of machines to identify deviations from the baseline (e.g., by a particular machine in the class).

[00487] Fig. 3S illustrates an example of a process for detecting anomalies. In various embodiments, process 392 is performed by data platform 12. As explained above, a given session will have an original user. And, each action taken by the original user can be tied back to the original user, despite privilege changes and/or lateral movement throughout a datacenter. Process 392 begins at 393 when log data associated with a user session (and thus an original user) is received. At 394, a logical graph is generated, using at least a portion of the collected data. When an anomaly is detected (395), it can be recorded, and as applicable, an alert is generated (396). The following are examples of graphs that can be generated (e.g., at 394), with corresponding examples of anomalies that can be detected (e.g., at 395) and alerted upon (e.g., at 396).

[00488] Fig. 4A illustrates a representation of an embodiment of an insider behavior graph. In the example of Fig. 4A, each node in the graph can be: (1) a cluster of users; (2) a cluster of launched processes; (3) a cluster of processes/servers running on a machine class; (4) a cluster of external IP addresses (of incoming clients); or (5) a cluster of external servers based on DNS/IP/etc. As depicted in Fig. 4A, graph data is vertically tiered into four tiers. Tier 0 (400) corresponds to entry point information (e.g., domains, IP addresses, and/or geolocation information) associated with a client entering the datacenter from an external entry point. Entry points are clustered together based on such information. Tier 1 (401) corresponds to a user on a machine class, with a given user on a given machine class represented as a node. Tier 2 (402) corresponds to launched processes, child processes, and/or interactive processes. Processes for a given user and having similar connectivity (e.g., sharing the processes they launch and the machines with which they communicate) are grouped into nodes. Finally, Tier 3 (403) corresponds to the services/servers/domains/IP addresses with which processes communicate. A relationship

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00195

between the tiers can be stated as follows: Tier 0 nodes log in to tier 1 nodes. Tier 1 nodes launch tier 2 nodes. Tier 2 nodes connect to tier 3 nodes.

[00489] The inclusion of an original user in both Tier 1 and Tier 2 allows for horizontal tiering. Such horizontal tiering ensures that there is no overlap between any two users in Tier 1 and Tier 2. Such lack of overlap provides for faster searching of an end-to-end path (e.g., one starting with a Tier 0 node and terminating at a Tier 3 node). Horizontal tiering also helps in establishing baseline insider behavior. For example, by building an hourly insider behavior graph, new edges/changes in edges between nodes in Tier 1 and Tier 2 can be identified. Any such changes correspond to a change associated with the original user. And, any such changes can be surfaced as anomalous and alerts can be generated.

[00490] As explained above, Tier 1 corresponds to a user (e.g., user "U") logging into a machine having a particular machine class (e.g., machine class "M"). Tier 2 is a cluster of processes having command line similarity (e.g., CType "C"), having an original user "U," and running as a particular effective user (e.g., user "U1"). The value of U1 may be the same as U (e.g., joe.smith in both cases), or the value of U1 may be different (e.g., U=joe.smith and U1=root). Thus, while an edge may be present from a Tier 1 node to a Tier 2 node, the effective user in the Tier 2 node may or may not match the original user (while the original user in the Tier 2 node will match the original user in the Tier 1 node).

[00491] A change from a user U into a user U1 can take place in a variety of ways. Examples include where U becomes U1 on the same machine (e.g., via su), and also where U sshes to other machine(s). In both situations, U can perform multiple changes, and can combine approaches. For example, U can become U1 on a first machine, ssh to a second machine (as U1), become U2 on the second machine, and ssh to a third machine (whether as user U2 or user U3). In various embodiments, the complexity of how user U ultimately becomes U3 (or U5, etc.) is hidden from a viewer of an insider behavior graph, and only an original user (e.g., U) and the effective user of a given node (e.g., U5) are depicted. As applicable (e.g., if desired by a viewer of the insider behavior graph), additional detail about the path (e.g., an end-to-end path of edges from user U to user U5) can be surfaced (e.g., via user interactions with nodes).

[00492] Fig. 4B illustrates an example of a portion of an insider behavior graph (e.g., as rendered in a web browser). In the example shown, node 405 (the external IP address, 52.32.40.231) is an example of a Tier 0 node, and represents an entry point into a datacenter. As indicated by directional arrows 406 and 407, two users, "aruneli_prod" and "harish_prod," both made use of the source IP 52.32.40.231 when logging in between 5pm and 6pm on Sunday July 30 (408). Nodes 409 and 410 are examples of Tier 1 nodes, having aruneli_prod and harish_prod as associated respective original users. As previously mentioned, Tier 1 nodes correspond to a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00196

combination of a user and a machine class. In the example depicted in Fig. 4B, the machine class associated with nodes 409 and 410 is hidden from view to simplify visualization, but can be surfaced to a viewer of interface 404 (e.g., when the user clicks on node 409 or 410).

[00493] Nodes 414-423 are examples of Tier 2 nodes – processes that are launched by users in Tier 1 and their child, grandchild, etc. processes. Note that also depicted in Fig. 4B is a Tier 1 node 411 that corresponds to a user, "root," that logged in to a machine cluster from within the datacenter (i.e., has an entry point within the datacenter). Nodes 425-1 and 425-2 are examples of Tier 3 nodes – internal/external IP addresses, servers, etc., with which Tier 2 nodes communicate.

[00494] In the example shown in Fig. 4B, a viewer of interface 404 has clicked on node 423. As indicated in region 426, the user running the marathon container is "root." However, by following the directional arrows in the graph backwards from node 423 (i.e. from right to left), the viewer can determine that the original user, responsible for node 423, is "aruneli_prod," who logged into the datacenter from IP 52.32.40.231.

[00495] The following are examples of changes that can be tracked using an insider behavior graph model:

[00496] * A user logs in from a new IP address.

[00497] * A user logs in from a geolocation not previously used by that user.

[00498] * A user logs into a new machine class.

[00499] * A user launches a process not previously used by that user.

[00500] * A user connects to an internal server to which the user has not previously connected.

[00501] * An original user communicates with an external server (or external server known to be malicious) with which that user has not previously communicated.

[00502] * A user communicates with an external server which has a geolocation not previously used by that user.

[00503] Such changes can be surfaced as alerts, e.g., to help an administrator determine when/what anomalous behavior occurs within a datacenter. Further, the behavior graph model can be used (e.g., during forensic analysis) to answer questions helpful during an investigation. Examples of such questions include:

[00504] * Was there any new login activity (Tier 0) in the timeframe being investigated? As one example, has a user logged in from an IP address with unknown geolocation information? Similarly, has a user started communicating externally with a new Tier 3 node (e.g., one with unknown geolocation information).

[00505] * Has there been any suspicious login activity (Tier 0) in the timeframe being investigated? As one example, has a user logged in from an IP address that corresponds to a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00197

known bad IP address as maintained by Threat aggregator 150? Similarly, has there been any suspicious Tier 3 activity?

[00506] * Were any anomalous connections made within the datacenter during the timeframe being investigated? As one example, suppose a given user ("Frank") typically enters a datacenter from a particular IP address (or range of IP addresses), and then connects to a first machine type (e.g., bastion), and then to a second machine type (e.g., database_prod). If Frank has directly connected to database_prod (instead of first going through bastion) during the timeframe, this can be surfaced using the insider graph.

[00507] * Who is (the original user) responsible for running a particular process?

[00508] An example of an insider behavior graph being used in an investigation is depicted in Figs. 4C and 4D. Fig. 4C depicts a baseline of behavior for a user, "Bill." As shown in Fig. 4C, Bill typically logs into a datacenter from the IP address, 71.198.44.40 (427). He typically makes use of ssh (428), and sudo (429), makes use of a set of typical applications (430) and connects (as root) with the external service, api.lacework.net (431).

[00509] Suppose Bill's credentials are compromised by a nefarious outsider ("Eve"). Fig. 4D depicts an embodiment of how the graph depicted in Fig. 4C would appear once Eve begins exfiltrating data from the datacenter. Eve logs into the datacenter (using Bill's credentials) from 52.5.66.8 (432). As Bill, Eve escalates her privilege to root (e.g., via su), and then becomes a different user, Alex (e.g., via su alex). As Alex, Eve executes a script, "sneak.sh" (433), which launches another script, "post.sh" (434), which contacts external server 435 which has an IP address of 52.5.66.7, and transmits data to it. Edges 436-439 each represent changes in Bill's behavior. As previously mentioned, such changes can be detected as anomalies and associated alerts can be generated. As a first example, Bill logging in from an IP address he has not previously logged in from (436) can generate an alert. As a second example, while Bill does typically make use of sudo (429), he has not previously executed sneak.sh (433) or post.sh (434) and the execution of those scripts can generate alerts as well. As a third example, Bill has not previously communicated with server 435, and an alert can be generated when he does so (439). Considered individually, each of edges 436-439 may indicate nefarious behavior, or may be benign. As an example of a benign edge, suppose Bill begins working from a home office two days a week. The first time he logs in from his home office (i.e., from an IP address that is not 71.198.44.40), an alert can be generated that he has logged in from a new location. Over time, however, as Bill continues to log in from his home office but otherwise engages in typical activities, Bill's graph will evolve to include logins from both 71.198.44.40 and his home office as baseline behavior. Similarly, if Bill begins using a new tool in his job, an alert can be generated the first time he executes the tool, but over time will become part of his baseline.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00198

[00510] In some cases, a single edge can indicate a serious threat. For example, if server 432 (or 435) is included in a known bad IP listing, edge 436 (or 439) indicates compromise. An alert that includes an appropriate severity level (e.g., "threat level high") can be generated. In other cases, a combination of edges could indicate a threat (where a single edge might otherwise result in a lesser warning). In the example shown in Fig. 4D, the presence of multiple new edges is indicative of a serious threat. Of note, even though "sneak.sh" and "post.sh" were executed by Alex, because data platform 12 also keeps track of an original user, the compromise of user B's account will be discovered.

[00511] Fig. 4E illustrates a representation of an embodiment of a user login graph. In the example of Fig. 4E, tier 0 (440) clusters source IP addresses as belonging to a particular country (including an "unknown" country) or as a known bad IP. Tier 1 (441) clusters user logins, and tier 2 (442) clusters type of machine class into which a user is logging in. The user login graph tracks the typical login behavior of users. By interacting with a representation of the graph, answers to questions such as the following can be obtained:

[00512] * Where is a user logging in from?

[00513] * Have any users logged in from a known bad address?

[00514] * Have any non-developer users accessed development machines?

[00515] * Which machines does a particular user access?

[00516] Examples of alerts that can be generated using the user login graph include:

[00517] * A user logs in from a known bad IP address.

[00518] * A user logs in from a new country for the first time.

[00519] * A new user logs into the datacenter for the first time.

[00520] * A user accesses a machine class that the user has not previously accessed.

[00521] One way to track privilege changes in a datacenter is by monitoring a process hierarchy of processes. To help filter out noisy commands/processes such as "su –u," the hierarchy of processes can be constrained to those associated with network activity. In a *nix system, each process has two identifiers assigned to it, a process identifier (PID) and a parent process identifier (PPID). When such a system starts, the initial process is assigned a PID 0. Each user process has a corresponding parent process.

[00522] Using techniques described herein, a graph can be constructed (also referred to herein as a privilege change graph) which models privilege changes. In particular, a graph can be constructed which identifies where a process P1 launches a process P2, where P1 and P2 each have an associated user U1 and U2, with U1 being an original user, and U2 being an effective user. In the graph, each node is a cluster of processes (sharing a CType) executed by a particular (original) user. As all the processes in the cluster belong to the same user, a label that can be used

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00199

for the cluster is the user's username. An edge in the graph, from a first node to a second node, indicates that a user of the first node changed its privilege to the user of the second node.

[00523] Fig. 4F illustrates an example of a privilege change graph. In the example shown in Fig. 4F, each node (e.g., nodes 444 and 445) represents a user. Privilege changes are indicated by edges, such as edge 446.

[00524] As with other graphs, anomalies in graph 443 can be used to generate alerts. Three examples of such alerts are as follows:

[00525] * New user entering the datacenter. Any time a new user enters the datacenter and runs a process, the graph will show a new node, with a new CType. This indicates a new user has been detected within the datacenter. Fig. 4F is a representation of an example of an interface that depicts such an alert. Specifically, as indicated in region 447, an alert for the time period 1pm-2pm on June 8 was generated. The alert identifies that a new user, Bill (448) executed a process.

[00526] *Privilege change. As explained above, a new edge, from a first node (user A) to a second node (user B) indicates that user A has changed privilege to user B.

[00527] * Privilege escalation. Privilege escalation is a particular case of privilege change, in which the first user becomes root.

[00528] An example of an anomalous privilege change and an example of an anomalous privilege escalation are each depicted in graph 450 of Fig. 4G. In particular, as indicated in region 451, two alerts for the time period 2pm-3pm on June 8 were generated (corresponding to the detection of the two anomalous events). In region 452, root has changed privilege to the user "daemon," which root has not previously done. This anomaly is indicated to the user by highlighting the daemon node (e.g., outlining it in a particular color, e.g., red). As indicated by edge 453, Bill has escalated his privilege to the user root (which can similarly be highlighted in region 454). This action by Bill represents a privilege escalation.

[00529] An Extensible query interface for dynamic data compositions and filter applications will now be described.

[00530] As described herein, datacenters are highly dynamic environments. And, different customers of data platform 12 (e.g., entity A vs. entity B) may have different/disparate needs/requirements of data platform 12, e.g., due to having different types of assets, different applications, etc. Further, as time progresses, new software tools will be developed, new types of anomalous behavior will be possible (and should be detectable), etc. In various embodiments, data platform 12 makes use of predefined relational schema (including by having different predefined relational schema for different customers). However, the complexity and cost of maintaining/updating such predefined relational schema can rapidly become problematic – particularly where the schema includes a mix of relational, nested, and hierarchical (graph)

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00200

datasets. In other embodiments, the data models and filtering applications used by data platform 12 are extensible. As will be described in more detail below, in various embodiments, data platform 12 supports dynamic query generation by automatic discovery of join relations via static or dynamic filtering key specifications among composable data sets. This allows a user of data platform 12 to be agnostic to modifications made to existing data sets as well as creation of new data sets. The extensible query interface also provides a declarative and configurable specification for optimizing internal data generation and derivations.

[00531] As will also be described in more detail below, data platform 12 is configured to dynamically translate user interactions (e.g., received via web app 120) into SQL queries (and without the user needing to know how to write queries). Such queries can then be performed (e.g., by query service 166) against any compatible backend (e.g., data store 30).

[00532] Fig. 4H illustrates an example of a user interacting with a portion of an interface. When a user visits data platform 12 (e.g., via web app 120 using a browser), data is extracted from data store 30 as needed (e.g., by query service 166), to provide the user with information, such as the visualizations depicted variously herein). As the user continues to interact with such visualizations (e.g., clicking on graph nodes, entering text into search boxes, navigating between tabs (e.g., tab 455 vs. 465)), such interactions act as triggers that cause query service 166 to continue to obtain information from data store 30 as needed (and as described in more detail below).

[00533] In the example shown in Fig. 4H, user A is viewing a dashboard that provides various information about entity A users (455), during the time period March 2 at midnight – March 25 at 7pm (which she selected by interacting with region 456). Various statistical information is presented to user A in region 457. Region 458 presents a timeline of events that occurred during the selected time period. User A has opted to list only the critical, high, and medium events during the time period by clicking on the associated boxes (459-461). A total of 55 low severity, and 155 info-only events also occurred during the time period. Each time user A interacts with an element in Fig. 4H (e.g., clicks on box 461, clicks on link 464-1, or clicks on tab 465), her actions are translated/formalized into filters on the data set and used to dynamically generate SQL queries. The SQL queries are generated transparently to user A (and also to a designer of the user interface shown in Fig. 4H).

[00534] User A notes in the timeline (462) that a user, Harish, connected to a known bad server (examplebad.com) using wget, an event that has a critical severity level. User A can click on region 463 to expand details about the event inline (which will display, for example, the text "External connection made to known bad host examplebad.com at port 80 from application 'wget' running on host dev1.lacework.internal as user harish") directly below line 462. User A

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00201

can also click on link 464-1, which will take her to a dossier for the event (depicted in Fig. 4I). As will be described in more detail below, a dossier is a template for a collection of visualizations.

[00535] As shown in interface 466, the event of Harish using wget to contact examplebad.com on March 16 was assigned an event ID of 9291 by data platform 12 (467). For convenience to user A, the event is also added to her dashboard in region 476 as a bookmark (468). A summary of the event is depicted in region 469. By interacting with boxes shown in region 470, user A can see a timeline of related events. In this case, user A has indicated that she would like to see other events involving the wget application (by clicking box 471). Events of critical and medium security involving wget occurred during the one hour window selected in region 472.

[00536] Region 473 automatically provides user A with answers to questions that may be helpful to have answers to while investigating event 9291. If user A clicks on any of the links in the event description (474), she will be taken to a corresponding dossier for the link. As one example, suppose user A clicks on link 475. She will then be presented with interface 477 shown in Fig. 4J.

[00537] Interface 477 is an embodiment of a dossier for a domain. In this example, the domain is "examplebad.com," as shown in region 478. Suppose user A would like to track down more information about interactions entity A resources have made with examplebad.com between January 1 and March 20. She selects the appropriate time period in region 479 and information in the other portions of interface 477 automatically update to provide various information corresponding to the selected time frame. As one example, user A can see that contact was made with examplebad.com a total of 17 times during the time period (480), as well as a list of each contact (481). Various statistical information is also included in the dossier for the time period (482). If she scrolls down in interface 477, user A will be able to view various polygraphs associated with examplebad.com, such as an application-communication polygraph (483).

[00538] Data stored in data store 30 can be internally organized as an activity graph. In the activity graph, nodes are also referred to as Entities. Activities generated by Entities are modeled as directional edges between nodes. Thus, each edge is an activity between two Entities. One example of an Activity is a "login" Activity, in which a user Entity logs into a machine Entity (with a directed edge from the user to the machine). A second example of an Activity is a "launch" Activity, in which a parent process launches a child process (with a directed edge from the parent to the child). A third example of an Activity is a "DNS query" Activity, in which either a process or a machine performs a query (with a directed edge from the requestor to the answer, e.g., an edge from a process to www.example.com). A fourth example of an Activity is a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00202

network "connected to" Activity, in which processes, IP addresses, and listen ports can connect to each other (with a directed edge from the initiator to the server).

[00539] As will be described in more detail below, query service 166 provides either relational views or graph views on top of data stored in data store 30. Typically, a user will want to see data filtered using the activity graph. For example, if an entity was not involved in an activity in a given time period, that entity should be filtered out of query results. Thus, a request to show "all machines" in a given time frame will be interpreted as "show distinct machines that were active" during the time frame.

[00540] Query service 166 relies on three main data model elements: fields, entities, and filters. As used herein, a field is a collection of values with the same type (logical and physical). A field can be represented in a variety of ways, including: 1. a column of relations (table/view), 2. a return field from another entity, 3. an SQL aggregation (e.g., COUNT, SUM, etc.), 4. an SQL expression with the references of other fields specified, and 5. a nested field of a JSON object. As viewed by query service 166, an entity is a collection of fields that describe a data set. The data set can be composed in a variety of ways, including: 1. a relational table, 2. a parameterized SQL statement, 3. DynamicSQL created by a Java function, and 4. join/project/aggregate/subclass of other entities. Some fields are common for all entities. One example of such a field is a "first observed" timestamp (when first use of the entity was detected). A second example of such a field is the entity classification type (e.g., one of: 1. Machine (on which an agent is installed), 2. Process, 3. Binary, 4. UID, 5. IP, 6. DNS Information, 7. ListenPort, and 8. PType). A third example of such a field is a "last observed" timestamp.

[00541] A filter is an operator that: 1. takes an entity and field values as inputs, 2. a valid SQL expression with specific reference(s) of entity fields, or 3. is a conjunct/disjunct of filters. As will be described in more detail below, filters can be used to filter data in various ways, and limit data returned by query service 166 without changing the associated data set.

[00542] As mentioned above, a dossier is a template for a collection of visualizations. Each visualization (e.g., the box including chart 484) has a corresponding card, which identifies particular target information needed (e.g., from data store 30) to generate the visualization. In various embodiments, data platform 12 maintains a global set of dossiers/cards. Users of data platform 12 such as user A can build their own dashboard interfaces using preexisting dossiers/cards as components, and/or they can make use of a default dashboard (which incorporates various of such dossiers/cards).

[00543] A JSON file can be used to store multiple cards (e.g., as part of a query service catalog). A particular card is represented by a single JSON object with a unique name as a field name.

[00544] Each card may be described by the following named fields:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00203

[00545] TYPE: the type of the card. Example values include:

[00546] * Entity (the default type)

[00547] * SQL

[00548] * Filters

[00549] * DynamicSQL

[00550] * graphFilter

[00551] * graph

[00552] * Function

[00553] * Template

[00554] PARAMETERS: a JSON array object that contains an array of parameter objects with the following fields:

[00555] * name (the name of the parameter)

[00556] * required (a Boolean flag indicating whether the parameter is required or not)

[00557] * default (a default value of the parameter)

[00558] * props (a generic JSON object for properties of the parameter. Possible values are: "utype" (a user defined type), and "scope" (an optional property to configure a namespace of the parameter))

[00559] * value (a value for the parameter - non-null to override the default value defined in nested source entities)

[00560] SOURCES: a JSON array object explicitly specifying references of input entities. Each source reference has the following attributes:

[00561] * name (the card/entity name or fully-qualified Table name)

[00562] * type (required for base Table entity)

[00563] * alias (an alias to access this source entity in other fields (e.g., returns, filters, groups, etc))

[00564] RETURNS: a required JSON array object of a return field object. A return field object can be described by the following attributes:

[00565] * field (a valid field name from a source entity)

[00566] * expr (a valid SQL scalar expression. References to input fields of source entities are specified in the format of #{Entity.Field}. Parameters can also be used in the expression in the format of ${ParameterName})

[00567] * type (the type of field, which is required for return fields specified by expr. It is also required for all return fields of an Entity with an SQL type)

[00568] * alias (the unique alias for return field)

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00204

[00569] * aggr (possible aggregations are: COUNT, COUNT_DISTINCT, DISTINCT, MAX, MIN, AVG, SUM, FIRST_VALUE, LAST_VALUE)

[00570] * case (JSON array object represents conditional expressions "when" and "expr")

[00571] * fieldsFrom, and, except (specification for projections from a source entity with excluded fields)

[00572] * props (general JSON object for properties of the return field. Possible properties include: "filterGroup," "title," "format," and "utype")

[00573] PROPS: generic JSON objects for other entity properties

[00574] SQL: a JSON array of string literals for SQL statements. Each string literal can contain parameterized expressions ${ParameterName} and/or composable entity by #{EntityName}

[00575] GRAPH: required for graph entity. Has the following required fields:

[00576] * source (including "type," "props," and "keys")

[00577] * target (including "type," "props," and "keys")

[00578] * edge (including "type" and "props")

[00579] JOINS: a JSON array of join operators. Possible fields for a join operator include:

[00580] * type (possible join types include: "loj" - Left Outer Join, "join" - Inner Join, "in" - Semi Join, "implicit" - Implicit Join)

[00581] * left (a left hand side field of join)

[00582] * right (a right hand side field of join)

[00583] * keys (key columns for multi-way joins)

[00584] * order (a join order of multi-way joins)

[00585] FKEYS: a JSON array of FilterKey(s). The fields for a FilterKey are:

[00586] * type (type of FilterKey)

[00587] * fieldRefs (reference(s) to return fields of an entity defined in the sources field)

[00588] * alias (an alias of the FilterKey, used in implicit join specification)

[00589] FILTERS: a JSON array of filters (conjunct). Possible fields for a filter include:

[00590] * type (types of filters, including: "eq" - equivalent to SQL =, "ne" - equivalent to SQL <>, "ge" - equivalent to SQL >=, "gt" - equivalent to SQL >, "le" - equivalent to SQL <=, "lt" - equivalent to SQL <, "like" - equivalent to SQL LIKE, "not_like" - equivalent to SQL NOT LIKE, "rlike" - equivalent to SQL RLIKE (Snowflake specific), "not_rlike" - equivalent to SQL NOT RLIKE (Snowflake specific), "in" - equivalent to SQL IN, "not_in" - equivalent to SQL NOT IN)

[00591] * expr (generic SQL expression)

[00592] * field (field name)

[00593] * value (single value)

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00205

[00594] * values (for both IN and NOT IN)

[00595] ORDERS: a JSON array of ORDER BY for returning fields. Possible attributes for the ORDER BY clause include:

[00596] * field (field ordinal index (1 based) or field alias)

[00597] * order (asc/desc, default is ascending order)

[00598] GROUPS: a JSON array of GROUP BY for returning fields. Field attributes are:

[00599] * field (ordinal index (1 based) or alias from the return fields)

[00600] LIMIT: a limit for the number of records to be returned

[00601] OFFSET: an offset of starting position of returned data. Used in combination with limit for pagination.

[00602] Suppose customers of data platform 12 (e.g., entity A and entity B) request new data transformations or a new aggregation of data from an existing data set (as well as a corresponding visualization for the newly defined data set). As mentioned above, the data models and filtering applications used by data platform 12 are extensible. Thus, two example scenarios of extensibility are (1) extending the filter data set, and (2) extending a FilterKey in the filter data set.

[00603] Data platform 12 includes a query service catalog that enumerates cards available to users of data platform 12. New cards can be included for use in data platform 12 by being added to the query service catalog (e.g., by an operator of data platform 12). For reusability and maintainability, a single external-facing card (e.g., available for use in a dossier) can be composed of multiple (nested) internal cards. Each newly added card (whether external or internal) will also have associated FilterKey(s) defined. A user interface (UI) developer can then develop a visualization for the new data set in one or more dossier templates. The same external card can be used in multiple dossier templates, and a given external card can be used multiple times in the same dossier (e.g., after customization). Examples of external card customization include customization via parameters, ordering, and/or various mappings of external data fields (columns).

[00604] As mentioned above, a second extensibility scenario is one in which a FilterKey in the filter data set is extended (i.e., existing template functions are used to define a new data set). As also mentioned above, data sets used by data platform 12 are composable/reusable/extensible, irrespective of whether the data sets are relational or graph data sets. One example data set is the User Tracking polygraph, which is generated as a graph data set (comprising nodes and edges). Like other polygraphs, User Tracking is an external data set that can be visualized both as a graph (via the nodes and edges) and can also be used as a filter data set for other cards, via the cluster identifier (CID) field.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00206

[00605] As mentioned above, as users such as user A navigate through/interact with interfaces provided by data platform 12 (e.g., as shown in Fig. 4H), such interactions trigger query service 166 to generate and perform queries against data store 30. Dynamic composition of filter datasets can be implemented using FilterKeys and FilterKey Types. A FilterKey can be defined as a list of columns and/or fields in a nested structure (e.g., JSON). Instances of the same FilterKey Type can be formed as an Implicit Join Group. The same instance of a FilterKey can participate in different Implicit Join Groups. A list of relationships among all possible Implicit Join Groups is represented as a Join graph for the entire search space to create a final data filter set by traversing edges and producing Join Path(s).

[00606] Each card (e.g., as stored in the query service catalog and used in a dossier) can be introspected by a /card/describe/CardID REST request.

[00607] At runtime (e.g., whenever it receives a request from web app 120), query service 166 parses the list of implicit joins and creates a Join graph to manifest relationships of FilterKeys among Entities. A Join graph (an example of which is depicted in Fig. 4K) comprises a list of Join Link(s). A Join Link represents each implicit join group by the same FilterKey type. A Join Link maintains a reverse map (Entity-to-FilterKey) of FilterKeys and their Entities. As previously mentioned, Entities can have more than one FilterKey defined. The reverse map guarantees one FilterKey per Entity can be used for each JoinLink. Each JoinLink also maintains a list of entities for the priority order of joins. Each JoinLink is also responsible for creating and adding directional edge(s) to graphs. An edge represents a possible join between two Entities.

[00608] At runtime, each Implicit Join uses the Join graph to find all possible join paths. The search of possible join paths starts with the outer FilterKey of an implicit join. One approach is to use a shortest path approach, with breadth first traversal and subject to the following criteria:

[00609] * Use the priority order list of Join Links for all entities in the same implicit join group.

[00610] * Stop when a node (Entity) is reached which has local filter(s).

[00611] * Include all join paths at the same level (depth).

[00612] * Exclude join paths based on the predefined rules (path of edges).

[00613] Fig. 4L illustrates an example of a process for dynamically generating and executing a query. In various embodiments, process 485 is performed by data platform 12. The process begins at 486 when a request is received to filter information associated with activities within a network environment. One example of such a request occurs in response to user A clicking on tab 465. Another example of such a request occurs in response to user A clicking on link 464-1. Yet another example of such a request occurs in response to user A clicking on link 464-2 and selecting (e.g., from a dropdown) an option to filter (e.g., include, exclude) based on specific criteria that she provides (e.g., an IP address, a username, a range of criteria, etc.).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00207

[00614] At 487, a query is generated based on an implicit join. One example of processing that can be performed at 487 is as follows. As explained above, one way dynamic composition of filter datasets can be implemented is by using FilterKeys and FilterKey Types. And, instances of the same FilterKey Type can be formed as an Implicit Join Group. A Join graph for the entire search space can be constructed from a list of all relationships among all possible Join Groups. And, a final data filter set can be created by traversing edges and producing one or more Join Paths. Finally, the shortest path in the join paths is used to generate an SQL query string.

[00615] One approach to generating an SQL query string is to use a query building library (authored in an appropriate language such as Java). For example, a common interface "sqlGen" may be used in conjunction with process 485 is as follows. First, a card/entity is composed by a list of input cards/entities, where each input card recursively is composed by its own list of input cards. This nested structure can be visualized as a tree of query blocks(SELECT) in standard SQL constructs. SQL generation can be performed as the traversal of the tree from root to leaf entities (top-down), calling the sqlGen of each entity. Each entity can be treated as a subclass of the Java class(Entity). An implicit join filter (EntityFilter) is implemented as a subclass of Entity, similar to the right hand side of a SQL semi-join operator. Unlike the static SQL semi-join construct, it is conditionally and recursively generated even if it is specified in the input sources of the JSON specification. Another recursive interface can also be used in conjunction with process 485, preSQLGen, which is primarily the entry point for EntityFilter to run a search and generate nested implicit join filters. During preSQLGen recursive invocations, the applicability of implicit join filters is examined and pushed down to its input subquery list. Another top-down traversal, pullUpCachable, can be used to pull up common sub-query blocks, including those dynamically generated by preSQLGen, such that SELECT statements of those cacheable blocks are generated only once at top-level WITH clauses. A recursive interface, sqlWith, is used to generate nested subqueries inside WITH clauses. The recursive calls of a sqlWith function can generate nested WITH clauses as well. An sqlFrom function can be used to generate SQL FROM clauses by referencing those subquery blocks in the WITH clauses. It also produces INNER/OUTER join operators based on the joins in the specification. Another recursive interface, sqlWhere, can be used to generate conjuncts and disjuncts of local predicates and semi-join predicates based on implicit join transformations. Further, sqlProject, sqlGroupBy, sqlOrderBy, and sqlLimitOffset can respectively be used to translate the corresponding directives in JSON spec to SQL SELECT list, GROUP BY, ORDER BY, and LIMIT/OFFSET clauses.

[00616] Returning to process 485, at 488, the query (generated at 487) is used to respond to the request. As one example of the processing performed at 488, the generated query is used to query

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00208

data store 30 and provide (e.g., to web app 120) fact data formatted in accordance with a schema (e.g., as associated with a card associated with the request received at 486).

[00617] Although the examples described herein largely relate to embodiments where data is collected from agents and ultimately stored in a data store such as those provided by Snowflake, in other embodiments data that is collected from agents and other sources may be stored in different ways. For example, data that is collected from agents and other sources may be stored in a data warehouse, data lake, data mart, and/or any other data store.

[00618] A data warehouse may be embodied as an analytic database (e.g., a relational database) that is created from two or more data sources. Such a data warehouse may be leveraged to store historical data, often on the scale of petabytes. Data warehouses may have compute and memory resources for running complicated queries and generating reports. Data warehouses may be the data sources for business intelligence ('BI') systems, machine learning applications, and/or other applications. By leveraging a data warehouse, data that has been copied into the data warehouse may be indexed for good analytic query performance, without affecting the write performance of a database (e.g., an Online Transaction Processing ('OLTP') database). Data warehouses also enable the joining data from multiple sources for analysis. For example, a sales OLTP application probably has no need to know about the weather at various sales locations, but sales predictions could take advantage of that data. By adding historical weather data to a data warehouse, it would be possible to factor it into models of historical sales data.

[00619] Data lakes, which store files of data in their native format, may be considered as "schema on read" resources. As such, any application that reads data from the lake may impose its own types and relationships on the data. Data warehouses, on the other hand, are "schema on write," meaning that data types, indexes, and relationships are imposed on the data as it is stored in the EDW. "Schema on read" resources may be beneficial for data that may be used in several contexts and poses little risk of losing data. "Schema on write" resources may be beneficial for data that has a specific purpose, and good for data that must relate properly to data from other sources. Such data stores may include data that is encrypted using homomorphic encryption, data encrypted using privacy-preserving encryption, smart contracts, non-fungible tokens, decentralized finance, and other techniques.

[00620] Data marts may contain data oriented towards a specific business line whereas data warehouses contain enterprise-wide data. Data marts may be dependent on a data warehouse, independent of the data warehouse (e.g., drawn from an operational database or external source), or a hybrid of the two. In embodiments described herein, different types of data stores (including combinations thereof) may be leveraged. Such data stores may be proprietary or may be embodied as vendor provided products or services such as, for example, Google BigQuery,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00209

Druid, Amazon Redshift, IBM Db2, Dremio, Databricks Lakehouse Platform, Cloudera, Azure Synapse Analytics, and others.

[00621] The deployments (e.g., a customer's cloud deployment) that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may be provisioned, deployed, and/or managed using infrastructure as code ('IaC'). IaC involves the managing and/or provisioning of infrastructure through code instead of through manual processes. With IaC, configuration files may be created that include infrastructure specifications. IaC can be beneficial as configurations may be edited and distributed, while also ensuring that environments are provisioned in a consistent manner. IaC approaches may be enabled in a variety of ways including, for example, using IaC software tools such as Terraform by HashiCorp. Through the usage of such tools, users may define and provide data center infrastructure using JavaScript Object Notation ('JSON'), YAML, proprietary formats, or some other format. In some embodiments, the configuration files may be used to emulate a cloud deployment for the purposes of analyzing the emulated cloud deployment using the systems described herein. Likewise, the configuration files themselves may be used as inputs to the systems described herein, such that the configuration files may be inspected to identify vulnerabilities, misconfigurations, violations of regulatory requirements, or other issues. In fact, configuration files for multiple cloud deployments may even be used by the systems described herein to identify best practices, to identify configuration files that deviate from typical configuration files, to identify configuration files with similarities to deployments that have been determined to be deficient in some way, or the configuration files may be leveraged in some other ways to detect vulnerabilities, misconfigurations, violations of regulatory requirements, or other issues prior to deploying an infrastructure that is described in the configuration files. In some embodiments the techniques described herein may be use in multi-cloud, multi-tenant, cross-cloud, cross-tenant, cross-user, industry cloud, digital platform, and other scenarios depending on specific need or situation.

[00622] In some embodiments, the deployments that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may be monitored to determine the extent to which a particular component has experienced "drift" relative to its associated IaC configuration. Discrepancies between how cloud resources were defined in an IaC configuration file and how they are currently configured in runtime may be identified and remediation workflows may be initiated to generate an alert, reconfigure the deployment, or take some other action. Such discrepancies may occur for a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00210

variety of reasons. Such discrepancies may occur, for example, due to maintenance operations being performed, due to incident response tasks being carried out, or for some other reason. Readers will appreciate that while IaC helps avoid initial misconfigurations of a deployment by codifying and enforcing resource creation, resource configuration, security policies, and so on, the systems described herein may prevent unwanted drift from occurring during runtime and after a deployment has been created in accordance with an IaC configuration.

[00623] In some embodiments, the deployments (e.g., a customer's cloud deployment) that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may also be provisioned, deployed, and/or managed using security as code ('SaC'). SaC extends IaC concepts by defining cybersecurity policies and/or standards programmatically, so that the policies and/or standards can be referenced automatically in the configuration scripts used to provision cloud deployments. Stated differently, SaC can automate policy implementation and cloud deployments may even be compared with the policies to prevent "drift." For example, if a policy is created where all personally identifiable information ('PII') or personal health information ('PHI') must be encrypted when it is stored, that policy is translated into a process that is automatically launched whenever a developer submits code, and code that violates the policy may be automatically rejected.

[00624] In some embodiments, SaC may be implemented by initially classifying workloads (e.g., by sensitivity, by criticality, by deployment model, by segment). Policies that can be instantiated as code may subsequently be designed. For example, compute-related policies may be designed, access-related policies may be designed, application-related policies may be designed, network-related policies may be designed, data-related policies may be designed, and so on. Security as code may then be instantiated through architecture and automation, as successful implementation of SaC can benefit from making key architectural-design decisions and executing the right automation capabilities. Next, operating model protections may be built and supported. For example, an operating model may "shift left" to maximize self-service and achieve full-life-cycle security automation (e.g., by standardizing common development toolchains, CI/CD pipelines, and the like). In such an example, security policies and access controls may be part of the pipeline, automatic code review and bug/defect detection may be performed, automated build processes may be performed, vulnerability scanning may be performed, checks against a risk-control framework may be made, and other tasks may be performed all before deploying an infrastructure or components thereof.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00211

[00625] The systems described herein may be useful in analyzing, monitoring, evaluating, or otherwise observing a GitOps environment. In a GitOps environment, Git may be viewed as the one and only source of truth. As such, GitOps may require that the desired state of infrastructure (e.g., a customer's cloud deployment) be stored in version control such that the entire audit trail of changes to such infrastructure can be viewed or audited. In a GitOps environment, all changes to infrastructure are embodied as fully traceable commits that are associated with committer information, commit IDs, time stamps, and/or other information. In such an embodiment, both an application and the infrastructure (e.g., a customer's cloud deployment) that supports the execution of the application are therefore versioned artifacts and can be audited using the gold standards of software development and delivery. Readers will appreciate that while the systems described herein are described as analyzing, monitoring, evaluating, or otherwise observing a GitOps environment, in other embodiments other source control mechanisms may be utilized for creating infrastructure, making changes to infrastructure, and so on. In these embodiments, the systems described herein may similarly be used for analyzing, monitoring, evaluating, or otherwise observing such environments.

[00626] As described in other portions of the present disclosure, the systems described herein may be used to analyze, monitor, evaluate, or otherwise observe a customer's cloud deployment. While securing traditional datacenters requires managing and securing an IP-based perimeter with networks and firewalls, hardware security modules ('HSMs'), security information and event management ('SIEM') technologies, and other physical access restrictions, such solutions are not particularly useful when applied to cloud deployments. As such, the systems described herein may be configured to interact with and even monitor other solutions that are appropriate for cloud deployments such as, for example, "zero trust" solutions.

[00627] A zero trust security model (a.k.a., zero trust architecture) describes an approach to the design and implementation of IT systems. A primary concept behind zero trust is that devices should not be trusted by default, even if they are connected to a managed corporate network such as the corporate LAN and even if they were previously verified. Zero trust security models help prevent successful breaches by eliminating the concept of trust from an organization's network architecture. Zero trust security models can include multiple forms of authentication and authorization (e.g., machine authentication and authorization, human/user authentication and authorization) and can also be used to control multiple types of accesses or interactions (e.g., machine-to-machine access, human-to-machine access).

[00628] In some embodiments, the systems described herein may be configured to interact with zero trust solutions in a variety of ways. For example, agents that collect input data for the systems described herein (or other components of such systems) may be configured to access

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00212

various machines, applications, data sources, or other entity through a zero trust solution, especially where local instances of the systems described herein are deployed at edge locations. Likewise, given that zero trust solutions may be part of a customer's cloud deployment, the zero trust solution itself may be monitored to identify vulnerabilities, anomalies, and so on. For example, network traffic to and from the zero trust solution may be analyzed, the zero trust solution may be monitored to detect unusual interactions, log files generated by the zero trust solution may be gathered and analyzed, and so on.

[00629] In some embodiments, the systems described herein may leverage various tools and mechanisms in the process of performing its primary tasks (e.g., monitoring a cloud deployment). For example, Linux eBPF is mechanism for writing code to be executed in the Linux kernel space. Through the usage of eBPF, user mode processes can hook into specific trace points in the kernel and access data structures and other information. For example, eBPF may be used to gather information that enables the systems described herein to attribute the utilization of networking resources or network traffic to specific processes. This may be useful in analyzing the behavior of a particular process, which may be important for observability/SIEM.

[00630] The systems described may be configured to collect security event logs (or any other type of log or similar record of activity) and telemetry in real time for threat detection, for analyzing compliance requirements, or for other purposes. In such embodiments, the systems described herein may analyze telemetry in real time (or near real time), as well as historical telemetry, to detect attacks or other activities of interest. The attacks or activities of interest may be analyzed to determine their potential severity and impact on an organization. In fact, the attacks or activities of interest may be reported, and relevant events, logs, or other information may be stored for subsequent examination.

[00631] In one embodiment, systems described herein may be configured to collect security event logs (or any other type of log or similar record of activity) and telemetry in real time to provide customers with a SIEM or SIEM-like solution. SIEM technology aggregates event data produced by security devices, network infrastructure, systems, applications, or other source. Centralizing all of the data that may be generated by a cloud deployment may be challenging for a traditional SIEM, however, as each component in a cloud deployment may generate log data or other forms of machine data, such that the collective amount of data that can be used to monitor the cloud deployment can grow to be quite large. A traditional SIEM architecture, where data is centralized and aggregated, can quickly result in large amounts of data that may be expensive to store, process, retain, and so on. As such, SIEM technologies may frequently be implemented such that silos are created to separate the data.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00213

[00632] In some embodiments of the present disclosure, data that is ingested by the systems described herein may be stored in a cloud-based data warehouse such as those provided by Snowflake and others. Given that companies like Snowflake offer data analytics and other services to operate on data that is stored in their data warehouses, in some embodiments one or more of the components of the systems described herein may be deployed in or near Snowflake as part of a secure data lake architecture (a.k.a., a security data lake architecture, a security data lake/warehouse). In such an embodiment, components of the systems described herein may be deployed in or near Snowflake to collect data, transform data, analyze data for the purposes of detecting threats or vulnerabilities, initiate remediation workflows, generate alerts, or perform any of the other functions that can be performed by the systems described herein. In such embodiments, data may be received from a variety of sources (e.g., EDR or EDR-like tools that handle endpoint data, cloud access security broker ('CASB') or CASB-like tools that handle data describing interactions with cloud applications, Identity and Access Management ('IAM') or IAM-like tools, and many others), normalized for storage in a data warehouse, and such normalized data may be used by the systems described herein. In fact, the systems described herein may actually implement the data sources (e.g., an EDR tool, a CASB tool, an IAM tool) described above.

[00633] In some embodiments one data source that is ingested by the systems described herein is log data, although other forms of data such as network telemetry data (flows and packets) and/or many other forms of data may also be utilized. In some embodiments, event data can be combined with contextual information about users, assets, threats, vulnerabilities, and so on, for the purposes of scoring, prioritization and expediting investigations. In some embodiments, input data may be normalized, so that events, data, contextual information, or other information from disparate sources can be analyzed more efficiently for specific purposes (e.g., network security event monitoring, user activity monitoring, compliance reporting). The embodiments described here offer real-time analysis of events for security monitoring, advanced analysis of user and entity behaviors, querying and long-range analytics for historical analysis, other support for incident investigation and management, reporting (for compliance requirements, for example), and other functionality.

[00634] In some embodiments, the systems described herein may be part of an application performance monitoring ('APM') solution. APM software and tools enable the observation of application behavior, observation of its infrastructure dependencies, observation of users and business key performance indicators ('KPIs') throughout the application's life cycle, and more. The applications being observed may be developed internally, as packaged applications, as

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00214

software as a service ('SaaS'), or embodied in some other ways. In such embodiments, the systems described herein may provide one or more of the following capabilities:

[00635] The ability to operate as an analytics platform that ingests, analyzes, and builds context from traces, metrics, logs, and other sources.

[00636] Automated discovery and mapping of an application and its infrastructure components.

[00637] Observation of an application's complete transactional behavior, including interactions over a data communications network.

[00638] Monitoring of applications running on mobile (native and browser) and desktop devices.

[00639] Identification of probable root causes of an application's performance problems and their impact on business outcomes.

[00640] Integration capabilities with automation and service management tools.

[00641] Analysis of business KPIs and user journeys (for example, login to check-out).

[00642] Domain-agnostic analytics capabilities for integrating data from third-party sources.

[00643] Endpoint monitoring to understand the user experience and its impact on business outcomes.

[00644] Support for virtual desktop infrastructure ('VDI') monitoring.

[00645] In embodiments where the systems described herein are used for APM, some components of the system may be modified, other components may be added, some components may be removed, and other components may remain the same. In such an example, similar mechanisms as described elsewhere in this disclosure may be used to collect information from the applications, network resources used by the application, and so on. The graph based modelling techniques may also be leveraged to perform some of the functions mentioned above, or other functions as needed.

[00646] In some embodiments, the systems described herein may be part of a solution for developing and/or managing artificial intelligence ('AI') or machine learning ('ML') applications. For example, the systems described herein may be part of an AutoML tool that automate the tasks associated with developing and deploying ML models. In such an example, the systems described herein may perform various functions as part of an AutoML tool such as, for example, monitoring the performance of a series of processes, microservices, and so on that are used to collectively form the AutoML tool. In other embodiments, the systems described herein may perform other functions as part of an AutoML tool or may be used to monitor, analyze, or otherwise observe an environment that the AutoML tool is deployed within.

[00647] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include other forms of AI/ML tools. For example, the systems described herein may manage, analyze, or otherwise observe deployments that include

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00215

AI services. AI services are, like other resources in an as-a-service model, ready-made models and AI applications that are consumable as services and made available through APIs. In such an example, rather than using their own data to build and train models for common activities, organizations may access pre-trained models that accomplish specific tasks. Whether an organization needs natural language processing ('NLP'), automatic speech recognition ('ASR'), image recognition, or some other capability, AI services simply plug-and-play into an application through an API. Likewise, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include other forms of AI/ML tools such as Amazon Sagemaker (or other cloud machine-learning platform that enables developers to create, train, and deploy ML models) and related services such as Data Wrangler (a service to accelerate data prep for ML) and Pipelines (a CI/CD service for ML).

[00648] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include various data services. For example, data services may include secure data sharing services, data marketplace services, private data exchanges services, and others. Secure data sharing services can allow access to live data from its original location, where those who are granted access to the data simply reference the data in a controlled and secure manner, without latency or contention from concurrent users. Because changes to data are made to a single version, data remains up-to-date for all consumers, which ensures data models are always using the latest version of such data. Data marketplace services operate as a single location to access live, ready-to-query data (or data that is otherwise ready for some other use). A data marketplace can even include a "feature stores," which can allow data scientists to repurpose existing work. For example, once a data scientist has converted raw data into a metric (e.g., costs of goods sold), this universal metric can be found quickly and used by other data scientists for quick analysis against that data.

[00649] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include distributed training engines or similar mechanisms such as, for example, such as tools built on Dask. Dask is an open source library for parallel computing that is written in Python. Dask is designed to enable data scientists to improve model accuracy faster, as Dask enables data scientists can do everything in Python end-to-end, which means that they no longer need to convert their code to execute in environments like Apache Spark. The result is reduced complexity and increased efficiency. The systems described herein may also be used to manage, analyze, or otherwise observe deployments that include technologies such as RAPIDS (an open source Python framework which is built on top of Dask). RAPIDS optimizes compute time and speed by providing data pipelines and executing data science code entirely on graphics processing units (GPUs) rather than CPUs. Multi-cluster,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00216

shared data architecture, DataFrames, Java user-defined functions (UDF) are supported to enable trained models to run within a data warehouse.

[00650] In some embodiments, the systems described herein may be leveraged for the specific use case of detecting and/or remediating ransomware attacks and/or other malicious action taken with respect to data, systems, and/or other resources associated with one or more entities. Ransomware is a type of malware from cryptovirology that threatens to publish the victim's data or perpetually block access to such data unless a ransom is paid. In such embodiments, ransomware attacks may be carried out in a manner such that patterns (e.g., specific process-to-process communications, specific data access patterns, unusual amounts of encryption/re-encryption activities) emerge, where the systems described herein may monitor for such patterns. Alternatively, ransomware attacks may involve behavior that deviates from normal behavior of a cloud deployment that is not experiencing a ransomware attack, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a ransomware attack.

[00651] In some embodiments, particular policies may be put in place the systems described herein may be configured to enforce such policies as part of an effort to thwart ransomware attacks. For example, particular network sharing protocols (e.g., Common Internet File System ('CIFS'), Network File System ('NFS')) may be avoided when implementing storage for backup data, policies that protect backup systems may be implemented and enforced to ensure that usable backups are always available, multifactor authentication for particular accounts may be utilized and accounts may be configured with the minimum privilege required to function, isolated recovery environments may be created and isolation may be monitored and enforced to ensure the integrity of the recovery environment, and so on. As described in the present disclosure, the systems described herein may be configured to explicitly enforce such policies or may be configured to detect unusual activity that represents a violation of such policies, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a violation of a particular policy.

[00652] Readers will appreciate that ransomware attacks are often deployed as part of a larger attack that may involve, for example:

[00653] Penetration of the network through means such as, for example, stolen credentials and remote access malware.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00217

[00654] Stealing of credentials for critical system accounts, including subverting critical administrative accounts that control systems such as backup, Active Directory ('AD'), DNS, storage admin consoles, and/or other key systems.

[00655] Attacks on a backup administration console to turn off or modify backup jobs, change retention policies, or even provide a roadmap to where sensitive application data is stored.

[00656] Data theft attacks.

[00657] As a result of the many aspects that are part of a ransomware attack, embodiments of the present disclosure may be configured as follows:

[00658] The systems may include one or more components that detect malicious activity based on the behavior of a process.

[00659] The systems may include one or more components that store indicator of compromise ('IOC') or indicator of attack ('IOA') data for retrospective analysis.

[00660] The systems may include one or more components that detect and block fileless malware attacks.

[00661] The systems may include one or more components that remove malware automatically when detected.

[00662] The systems may include a cloud-based, SaaS-style, multitenant infrastructure.

[00663] The systems may include one or more components that identify changes made by malware and provide the recommended remediation steps or a rollback capability.

[00664] The systems may include one or more components that detect various application vulnerabilities and memory exploit techniques.

[00665] The systems may include one or more components that continue to collect suspicious event data even when a managed endpoint is outside of an organization's network.

[00666] The systems may include one or more components that perform static, on-demand malware detection scans of folders, drives, devices, or other entities.

[00667] The systems may include data loss prevention (DLP) functionality.

[00668] In some embodiments, the systems described herein may manage, analyze, or otherwise observe deployments that include deception technologies. Deception technologies allow for the use of decoys that may be generated based on scans of true network areas and data. Such decoys may be deployed as mock networks running on the same infrastructure as the real networks, but when an intruder attempts to enter the real network, they are directed to the false network and security is immediately notified. Such technologies may be useful for detecting and stopping various types of cyber threats such as, for example, Advanced Persistent Threats ('APTs'), malware, ransomware, credential dumping, lateral movement and malicious insiders. To continue

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00218

to outsmart increasingly sophisticated attackers, these solutions may continuously deploy, support, refresh and respond to deception alerts.

[00669] In some embodiments, the systems described herein may manage, analyze, or otherwise observe deployments that include various authentication technologies, such as multi-factor authentication and role-based authentication. In fact, the authentication technologies may be included in the set of resources that are managed, analyzed, or otherwise observed as interactions with the authentication technologies may monitored. Likewise, log files or other information retained by the authentication technologies may be gathered by one or more agents and used as input to the systems described herein.

[00670] In some embodiments, the systems described herein may be leveraged for the specific use case of detecting supply chain attacks. More specifically, the systems described herein may be used to monitor a deployment that includes software components, virtualized hardware components, and other components of an organization's supply chain such that interactions with an outside partner or provider with access to an organization's systems and data can be monitored. In such embodiments, supply chain attacks may be carried out in a manner such that patterns (e.g., specific interactions between internal and external systems) emerge, where the systems described herein may monitor for such patterns. Alternatively, supply chain attacks may involve behavior that deviates from normal behavior of a cloud deployment that is not experiencing a supply chain attack, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a supply chain attack.

[00671] In some embodiments, the systems described herein may be leveraged for other specific use cases such as, for example, detecting the presence of (or preventing infiltration from) cryptocurrency miners (e.g., bitcoin miners), token miners, hashing activity, non-fungible token activity, other viruses, other malware, and so on. As described in the present disclosure, the systems described herein may monitor for such threats using known patterns or by detecting unusual activity, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a particular type of threat, intrusion, vulnerability, and so on.

[00672] The systems described herein may also be leveraged for endpoint protection, such the systems described herein form all of or part of an endpoint protection platform. In such an embodiment, agents, sensors, or similar mechanisms may be deployed on or near managed endpoints such as computers, servers, virtualized hardware, internet of things ('IotT') devices, mobile devices, phones, tablets, watches, other personal digital devices, storage devices, thumb

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00219

drives, secure data storage cards, or some other entity. In such an example, the endpoint protection platform may provide functionality such as:

[00673] Prevention and protection against security threats including malware that uses file-based and fileless exploits.

[00674] The ability to apply control (allow/block) to access of software, scripts, processes, microservices, and so on.

[00675] The ability to detect and prevent threats using behavioral analysis of device activity, application activity, user activity, and/or other data.

[00676] The ability for facilities to investigate incidents further and/or obtain guidance for remediation when exploits evade protection controls

[00677] The ability to collect and report on inventory, configuration and policy management of the endpoints.

[00678] The ability to manage and report on operating system security control status for the monitored endpoints.

[00679] The ability to scan systems for vulnerabilities and report/manage the installation of security patches.

[00680] The ability to report on internet, network and/or application activity to derive additional indications of potentially malicious activity.

[00681] Example embodiments are described in which policy enforcement, threat detection, or some other function is carried out by the systems described herein by detecting unusual activity, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a particular type of threat, intrusion, vulnerability, and so on. Although these examples are largely described in terms of identifying unusual activity, in these examples the systems described herein may be configured to learn what constitutes 'normal activity' – where 'normal activity' is activity observed, modeled, or otherwise identified in the absence of a particular type of threat, intrusion, vulnerability, and so on. As such, detecting 'unusual activity' may alternatively be viewed as detecting a deviation from 'normal activity' such that 'unusual activity' does not need to be identified and sought out. Instead, deviations from 'normal activity' may be assumed to be 'unusual activity'.

[00682] Readers will appreciate that while specific examples of the functionality that the systems described herein can provide are included in the present disclosure, such examples are not to be interpreted as limitations as to the functionality that the systems described herein can provide. Other functionality may be provided by the systems described herein, all of which are within the scope of the present disclosure. For the purposes of illustration and not as a limitation, additional

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00220

examples can include governance, risk, and compliance ('GRC'), threat detection and incident response, identity and access management, network and infrastructure security, data protection and privacy, identity and access management ('IAM'), and many others.

[00683] In order to provide the functionality described above, the systems described herein or the deployments that are monitored by such systems may implement a variety of techniques. For example, the systems described herein or the deployments that are monitored by such systems may tag data and logs to provide meaning or context, persistent monitoring techniques may be used to monitor a deployment at all times and in real time, custom alerts may be generated based on rules, tags, and/or known baselines from one or more polygraphs, and so on.

[00684] Although examples are described above where data may be collected from one or more agents, in some embodiments other methods and mechanisms for obtaining data may be utilized. For example, some embodiments may utilize agentless deployments where no agent (or similar mechanism) is deployed on one or more customer devices, deployed within a customer's cloud deployment, or deployed at another location that is external to the data platform. In such embodiments, the data platform may acquire data through one or more APIs such as the APIs that are available through various cloud services. For example, one or more APIs that enable a user to access data captured by Amazon CloudTrail may be utilized by the data platform to obtain data from a customer's cloud deployment without the use of an agent that is deployed on the customer's resources. In some embodiments, agents may be deployed as part of a data acquisition service or tool that does not utilize a customer's resources or environment. In some embodiments, agents (deployed on a customer's resources or elsewhere) and mechanisms in the data platform that can be used to obtain data from through one or more APIs such as the APIs that are available through various cloud services may be utilized. In some embodiments, one or more cloud services themselves may be configured to push data to some entity (deployed anywhere), which may or may not be an agent. In some embodiments, other data acquisition techniques may be utilized, including combinations and variations of the techniques described above, each of which is within the scope of the present disclosure.

[00685] Readers will appreciate that while specific examples of the cloud deployments that may be monitored, analyzed, or otherwise observed by the systems described herein have been provided, such examples are not to be interpreted as limitations as to the types of deployments that may be monitored, analyzed, or otherwise observed by the systems described herein. Other deployments may be monitored, analyzed, or otherwise observed by the systems described herein, all of which are within the scope of the present disclosure. For the purposes of illustration and not as a limitation, additional examples can include multi-cloud deployments, on-premises environments, hybrid cloud environments, sovereign cloud environments, heterogeneous

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00221

environments, DevOps environments, DevSecOps environments, GitOps environments, quantum computing environments, data fabrics, composable applications, composable networks, decentralized applications, and many others.

[00686] Readers will appreciate that while specific examples of the types of data that may be collected, transformed, stored, and/or analyzed by the systems described herein have been provided, such examples are not to be interpreted as limitations as to the types of data that may be collected, transformed, stored, and/or analyzed by the systems described herein. Other types of data can include, for example, data collected from different tools (e.g., DevOps tools, DevSecOps, GitOps tools), different forms of network data (e.g., routing data, network translation data, message payload data, Wi-Fi data, Bluetooth data, personal area networking data, payment device data, near field communication data, metadata describing interactions carried out over a network, and many others), data describing processes executing in a container, lambda, EC2 instance, virtual machine, or other execution environment), information describing the execution environment itself, and many other types of data.

[00687] In some embodiments, one or more of the components described above may be deployed using a secure access service edge ('SASE') model or similar model. In a SASE model, the services, functionality, or components described above may be deployed at edge devices (or relatively close to such edge device) such as a user's laptop, tablet, smartphone, or other device. In such a way, network security controls may be delivered on such edge devices. SASE capabilities may be delivered as a service based upon the identity of the entity, real-time context, enterprise security/compliance policies and continuous assessment of risk/trust throughout the sessions, where the identity of entities can be associated with people, groups of people, devices, applications, services, IoT systems or edge computing locations, and so on. In these embodiments, one or more of the components described above may be deployed at or near the edge devices, and the edge devices may even include local applications that are configured to utilize one or more of the components described above where the components are not deployed on the edge devices themselves.

[00688] For further explanation, Fig. 5 sets forth a flowchart illustrating an example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The cloud deployments 508, 514 may be similar to the cloud deployments described above, where a particular cloud deployment can include a variety of components 510, 512 such as one or more applications, one or more data sources, networking resources, processing resources, and other resources. Such components 510, 512 may, in some embodiments, be deployed in the cloud deployments 508, 514 using one or more as-a-service models where software, infrastructure,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00222

platforms, databases, and other components as delivered as services. Configuring cloud deployments 508, 514 based on learnings obtained by monitoring other cloud deployments may be carried out using the systems described above. As such, one or more of the steps depicted in Fig. 5 may be performed by the systems described above.

[00689] The example method depicted in Figure 5 includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508. Determining 502 normal behavior for one or more components 510 in a first cloud deployment 508 may be carried out, for example, as described in greater detail above (at times described as identifying 'normal activity') by the systems described above (also referred to herein as a 'data platform').

[00690] The example method depicted in Figure 5 also includes determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514. Determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514 may also be carried out, for example, as described in greater detail above (at times described as identifying 'normal activity') by the systems described above.

[00691] In some embodiments, a customer-specific data platform may be used to analyze, monitor, or otherwise observe a particular customer's cloud deployment (or some other deployment). Within such a cloud deployment, various clusters may exist. For example, a collection of microservices may form a cluster by virtue of those microservices communicating only (or mostly) with each other. Likewise, one or more cloud computing instances (e.g., one or more EC2 instances) and a database may form a cluster by virtue of the EC2 instances accessing the database as the only source of data utilized by the EC2 instances. Using the techniques and mechanisms described above, such clusters may be identified. Although clusters may be identified and characteristics associated with the cluster may be learned, limited insights may be gained if only a particular customer's cloud deployment is analyzed, monitored, or otherwise observed. In accordance with embodiments of the present disclosure, cross-customer analysis may be leveraged to gain deeper insights than would be gained if only a single customer's cloud deployment is analyzed, monitored, or otherwise observed.

[00692] In some embodiments, cross-customer analysis may be carried out by gathering information related to cloud deployments (or some other deployments) for multiple customers and comparing such information. Using the example described above, information describing clusters identified in a first customer's cloud deployment may be compared to information describing clusters identified in a second customer's cloud deployment for the purposes of identifying similar or identical clusters in each customer's cloud deployment.

[00693] Consider an example in which each customer's deployment included a web server that was deployed in one or more EC2 instances. In such an example, a particular cluster that

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00223

represents the web server may be identified in each customer's deployment. For example, a first cluster in the first customer's deployment may represent a first web server and a second cluster in the second customer's deployment may represent a second web server. Because the first cluster and the second cluster would have similar characteristics (e.g., each cluster receives data communications using HTTP or HTTPS or any other suitable communication protocol, each cluster communicates with a web browser, each cluster requires similar computing resources, and so on), the first cluster and the second cluster may be identified as being identical clusters by comparing the characteristics of each cluster that each cluster. This process may be repeated across the cloud deployments for many customers such that a collection of 'web server' clusters (in this example) may be identified.

[00694] Readers will appreciate that although the example described above relates to an embodiment where a collection of 'web server' clusters are identified in different customer's cloud deployments, identifying the nature or type (e.g., a web server) of the clusters is not required. In fact, by comparing the characteristics of different clusters to each other, similar or identical clusters may be identified even if the exact nature/type of those clusters is not known. For example, a comparison of the characteristics of multiple clusters may only reveal that the cluster are identical, even if such a comparison does not reveal that clusters are 'web server' clusters. Multiple clusters that have been identified as being identical (or sufficiently similar as measured by a threshold) will be referred to throughout the remainder of this document as a "cluster set," where the clusters that are members of the cluster set may be deployed across multiple customer's cloud deployments.

[00695] In some embodiments, information describing each member of the cluster set may be utilized to identify distributions across the cluster set. Distributions may be identified for traditional resource consumption metrics such as, for example, CPU usage, memory usage, network bandwidth usage, and others. A distribution may reveal, for example, that all members of the cluster set utilize between 10-60 Mb/s of network bandwidth, with the vast majority of members of the cluster set utilize between 45-60 Mb/s of network bandwidth. Readers will appreciate that distributions may also be identified for other quantifiable characteristics of each cluster. Such quantifiable characteristics can include, for example, the failure rate of a cluster or particular components thereof, an identification of communication protocols used by a cluster or particular components thereof, an identification of the types of communications endpoints that a cluster or particular components thereof communicate with (e.g., endpoints that reside on the public internet v. endpoints that are in a private network), characteristics that can be classified by a binary value (e.g., does any component in the cluster perform privileged operations),

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00224

information describing the various privileges that are given to a particular cluster, and many more.

[00696] In some embodiments, the distributions may be used to identify 'normal' behavior for a particular cluster. Consider an example in which the cluster set is identified, where each member of the cluster set represents a payroll system deployed in a particular customer's cloud deployment. In such an example, a distribution may be identified which indicates that all members of the cluster communicate with (and has privileged access to) the same set of cloud services, including: 1) a cloud database service (e.g., Amazon Aurora, Microsoft Azure SQL Database, Amazon Relational Database Service, Google Cloud SQL, Amazon DynamoDB), 2) a vendor provided SaaS offering that provides bill payment services, and 3) a vendor provided SaaS offering that provides accounting services. In such an example, by looking at each member of the cluster set and identifying that each member communicates with the same set of cloud services, a baseline may be established that identifies 'normal' behavior for each member of the cluster set, at least with respect to the specific characteristic (i.e., what cloud services are utilized by members) that the distribution is based on. As such, if monitoring a particular cluster revealed that some member of the cluster set accessed a source code repository cloud service (e.g., GitHub Enterprise on AWS), this sort of access would be outside of the typical distribution for this cluster set and could serve as the basis for raising an alert, denying access to the service, or initiating some other alerting/remediation workflow. Readers will appreciate that many distributions may be created for each cluster set, where each distribution is based on one or more characteristics of the members of the cluster set.

[00697] The example method depicted in Figure 5 also includes recommending 506, based on the normal behavior for one or more components 512 in one or more other cloud deployments 514, a change to the first cloud deployment 508. Recommending 506 a change to the first cloud deployment 508 may be carried out, for example, in response to determining that the normal behavior in one or more other cloud deployments 514 differs from the normal behavior for one or more components 510 in a first cloud deployment 508. In such an example, changes to the first cloud deployment 508 may be recommended that (if implemented) would cause the first cloud deployment 508 to be more similar to the other cloud deployments 514. For example, if the normal behavior in one or more other cloud deployments 514 indicates that all computing resources (e.g., virtual machines, container, serverless computing resources) communicate with each other using a particular secure data communications protocol and the normal behavior for one or more components 510 in a first cloud deployment 508 is for computing resources to communicate using some other data communications protocol, a change may be recommended

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00225

that involves reconfiguring the computing resources to communicate using the particular secure data communications protocol.

[00698] Readers will appreciate that in some embodiments the mere fact that normal behavior in a first cloud deployment 508 deviates from normal behavior in one or more other cloud deployments 514 may be sufficient rationale for recommending 506 a change to the first cloud deployment 508. That is, a departure from normality and standard practices alone may result in recommending 506 a change to the first cloud deployment 508. In other embodiments, recommending 506 a change to the first cloud deployment 508 may only occur where the normal behavior in one or more other cloud deployments 514 is determined to be superior to the normal behavior in the first cloud deployment 508. For example, recommending 506 a change to the first cloud deployment 508 may only be carried where the normal behavior in one or more other cloud deployments 514 is representative of a stronger security posture than the normal behavior in the first cloud deployment 508.

[00699] Consider an example in which a particular threat was detected (in part by detecting a deviation from normal behavior for one or more components 512) in a particular customer's cloud deployment 514, where the threat turned out to be a ransomware attack, which may in some embodiments include an encryption component and/or a data theft or leakage component. In such an example, if an identical (or sufficiently similar, following a general recognized pattern or 'fingerprint') threat is detected in the first customer's cloud deployment 508 (in part by detecting a similar deviation from deviation from normal behavior for one or more components 510), information describing the remedial actions (e.g., disabling encryption, increasing the frequency of backups, locking down a backup system, blocking transmission of data exernally, etc.) that were taken by the particular customer 514 may even recommended 506 as changes to be made to the first cloud deployment 508. Furthermore, if many customers had experienced the same attack and the data platform could determine with sufficient certainty that the first cloud deployment 508 was experiencing the same attack, workflows may be automatically initiated to carry out various remedial actions.

[00700] For further explanation, Fig. 6 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 6 is similar to the example depicted in Fig. 5, as the example depicted in Fig. 6 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00226

deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

[00701] In some embodiments, customers (and their corresponding deployments) may be modeled into logical groups such that cross customer learning could be carried out only across customers in the same logical group, or other customers in the same logical group may be given a greater weighting for the purposes of cross customer learning. For example, a logical group may be tied to a customer's business function (e.g., the customer is a financial company, the customer is a health care company, the customer is a services company, the customer has inventory to manage, the customer sells to retail customers, and so on). Likewise, a logical group may be tied to a customer's cloud deployment (e.g., the deployment uses a particular combination of resources, the deployment uses a particular set of cloud-services, the deployment utilizes availability zones and regions in a particular way, and so on). In such an example, a particular customer may be associated with multiple logical groups and cross customer learning for the particular customer may only involve other customers that are in the same (or sufficiently similar) logical groups. As such, the particular customer's resources (e.g., people, cloud deployment) may only be included in cluster sets with resources associated with other customers that are in the same (or sufficiently similar) logical groups.

[00702] To that end, the example method depicted in Fig. 6 also includes identifying 604, from the other cloud deployments, cloud deployments that are similar to the first cloud deployment 508. Identifying 604 cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by taking an inventory of the various components in each cloud deployment 508, 514 and identifying cloud deployments that have similar components, portions of the cloud deployment that have similar deployments, and so on. Alternatively, identifying 604 cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by identifying cloud deployments that are associated with customers in the same industry as the customer associated with the first cloud deployment 508, by identifying cloud deployments that are associated with a similar intended functionality as the first cloud deployment 508 (e.g., both cloud deployment are intended to provide an online store for an online retailer), and so on. In the examples in the preceding sentence, it may be assumed that two cloud deployments should be relatively similar even without inspecting the cloud deployments themselves. In such examples, similarity may be judge based on the application of rules, policies, heuristics, or similar mechanism. Alternatively, similarity may be detected via machine learning techniques as where information describing the activity of various components in many cloud deployments are fed as input to one or more machine learning models which subsequently identifies clusters across various cloud deployments that are similar.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00227

[00703] In the example depicted in Fig. 6, recommending 506 the change to the first cloud deployment 508 may be based on the normal behavior for the cloud deployments that are similar to the first cloud deployment. Recommending 606 the change to the first cloud deployment 508 based on the normal behavior for the cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by ignoring normal behavior for dissimilar cloud deployments and only taking into consideration those cloud deployments that have been identified 604 as being sufficiently similar to the first cloud deployment 508.

[00704] The example method depicted in Fig. 6 also includes identifying 602, from the other cloud deployments, one or more highly rated cloud deployments. A cloud deployment may be identified 602 as being 'highly rated' in the sense that fewer than a threshold number of vulnerabilities have been detected in the cloud deployment over some period of time, fewer than a threshold number of outages (or less than a threshold amount of downtime) has been detected in the cloud deployment over some period of time, fewer than a threshold number of regulatory violations (or attempted violations) have been detected in the cloud deployment over some period of time, or identified 602 based on some other standard. In fact, comparisons may be relative (rather than comparing some quantifiable aspect of the other deployments operation) such that, for example, the 15% of cloud deployments with the fewest detected vulnerabilities over some period of time as identified 602 as being 'highly rated'. In other embodiments, identifying 602 one or more highly rated cloud deployments may be carried out through the use of machine learning techniques where information describing the activity of various components in many cloud deployments are fed as input to one or more machine learning models which subsequently identifies deployments that have the best performance against a combination of one or more metrics (e.g., vulnerabilities, availability, cost). In such a way, the best-of-breed cloud deployments may be identified 602.

[00705] In the example depicted in Fig. 6, recommending 506 the change to the first cloud deployment 508 may be based on the normal behavior for the highly rated cloud deployments. Recommending 608 the change to the first cloud deployment 508 based on the normal behavior for the highly rated cloud deployments may be carried out, for example, by ignoring normal behavior for cloud deployments that are not highly rated and only taking into consideration those cloud deployments are highly rated.

[00706] Readers will appreciate that although the example in Fig. 6 is depicted where identifying 604 cloud deployments that are similar to the first cloud deployment 508 and identifying 602 one or more highly rated cloud deployments are alternatives to each other, in other embodiments both steps may be performed. In such embodiments, recommending 506 the change to the first cloud

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00228

deployment 508 may be based on the normal behavior other cloud deployments that are both highly rated and similar to the first cloud deployment 508.

[00707] For further explanation, Fig. 7 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 7 is similar to the examples depicted in Fig. 5 and Fig. 6, as the example depicted in Fig. 7 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

[00708] The example method depicted in Fig. 7 also includes ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514. Ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514 may be carried out, for example, by identifying how well the cloud deployment 508 compares to other cloud deployments 514 with respect to one or more metrics. For example, cloud deployments may be ranked based on a weighted combination of multiple metrics (e.g., reliability, cost, regulatory compliance) such that the best-of-breed deployments may be identified. In such an example, the ranking of the first cloud deployment 508 relative to the other cloud deployments 514 may be presented to an administrator or other user associated with the first cloud deployment 508 for analysis by the administrator or other user.

[00709] The example method depicted in Fig. 7 also includes comparing 704 the trajectory of the first cloud deployment 508 to the trajectory of the other cloud deployments 514. The trajectory of a particular cloud deployment 508, 514 may be determined, for example, by evaluating the particular cloud deployment's 508, 514 performance over time as measured by one or more metrics. For example, the performance of each cloud deployment may be periodically scored based on a weighted combination of multiple metrics (e.g., reliability, cost, regulatory compliance). In such an example, the trajectory for a particular cloud deployment may be determined by determining the extent to which the score associated with a particular cloud deployment is changing over time. For example, if the first cloud deployment's 508 scoring is improving over time while the other cloud deployments' 514 scores are remaining the same over time, the trajectory of the first cloud deployment 508 may be determined to be better than the trajectory of the other cloud deployments 514 (in embodiments where a higher score is considered to be better).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00229

[00710] Readers will appreciate that ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514 and comparing 704 the trajectory of the first cloud deployment 508 to the trajectory of the other cloud deployments 514 are just two examples of the sort of analytics that can be performed by comparing the performance of multiple cloud deployments 508, 514. Readers will appreciate that other analytics may also be put in place due to the availability of information describing the performance of multiple cloud deployments 508, 514.

[00711] For further explanation, Fig. 8 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 8 is similar to the examples depicted in Figs. 5-7, as the example depicted in Fig. 8 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

[00712] The example method depicted in Fig. 8 also includes identifying 802 other cloud deployments 514 to exclude from consideration when recommending changes to the first cloud deployment 508. Readers will appreciate that other cloud deployments 514 may be excluded from consideration when recommending changes to the first cloud deployment 508 for a variety of reasons. For example, another cloud deployment 514 may be so dissimilar to the first cloud deployment 508 that there may be very little to learn that is relevant to the first cloud deployment 508, another cloud deployment 514 may be deficient for a variety of reasons such that there may be very little high quality takeaways from evaluating the other cloud deployment 514 that are relevant to the first cloud deployment 508, another cloud deployment 514 may be a relatively recent deployment such that any takeaways from evaluating the other cloud deployment 514 may not be reliable and proven over time, or for some other reason. In such an example, identifying 802 other cloud deployments 514 to exclude from consideration when recommending changes to the first cloud deployment 508 may be carried out by filtering the set of other cloud deployments 514 according to some criteria, or carried out in some other way. In the example depicted in Fig. 8, recommending 506 the change to the first cloud deployment 508 may therefore include recommending 806 changes based on the other cloud deployments 514 that are not excluded from consideration.

[00713] The example method depicted in Fig. 8 also includes identifying 804 a response to a recommended change in one or more other cloud deployments 514. Identifying 804 a response to a recommended change in one or more other cloud deployments 514 may be carried out, for

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00230

example, by tracking the status of a recommended change that was recommended for another cloud deployment 514 to determine whether the recommended change was actually implemented. Consider an example in which a recommendation was made to have another cloud deployment 514 switch from using a first IaaS offering to a second IaaS offering in response to detecting some condition in the cloud deployment 514. In such an example, identifying 804 a response to a recommended change in one or more other cloud deployments 514 may be carried out by determining whether the recommended change was actually implemented and a switch was made to utilize the second IaaS offering as part of the cloud deployment 514. Readers will appreciate that by repeatedly identifying 804 responses to recommended changes in one or more other cloud deployments 514 can be used by the systems described above as an indication as to the systems described above are making useful recommendations. For example, if identifying 804 a response to a recommended change in one or more other cloud deployments 514 reveals that a very small percentage of administrators actually implement the recommended change, this may indicate that the recommend change is not particularly valuable and may be used as feedback in determining whether to make the recommended change if the condition that triggered the recommendation is encountered again. Alternatively, if identifying 804 a response to a recommended change in one or more other cloud deployments 514 reveals that a very large percentage of administrators actually implement the recommended change, this may indicate that the recommend change is valuable and may be used as feedback in determining whether to make the recommended change if the condition that triggered the recommendation is encountered again. In such a way, the response from other customer's may be used to drive recommendations that are made regarding the first cloud deployment 508.

[00714] In the example depicted in Fig. 8, recommending 506 the change to the first cloud deployment 508 recommending 808 the change to the first cloud deployment 508 based on the response to the recommended change in one or more other cloud deployments 514. Recommending 808 the change to the first cloud deployment 508 based on the response to the recommended change in one or more other cloud deployments 514 may be carried out, for example, only if a predetermined percentage of previous recipients of similar recommendations have actually implemented the recommended change, as described above.

[00715] Readers will appreciate that while the examples described in the preceding two paragraphs relate to an example in which the response to a recommended change in one or more other cloud deployments 514 is expressed in terms of whether the recommended change was or was not implemented, in other embodiments the response to a recommended change in one or more other cloud deployments 514 may be measured in other ways. For example, a response to a recommended change in one or more other cloud deployments 514 may represent whether the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00231

recommended change did or did not resolve the condition that caused the recommendation to be generated in the first place. In other embodiments, the response to a recommended change in one or more other cloud deployments 514 may be measured or determined using different criteria.

[00716] Although some of the figures described above depict only a single other cloud deployment 514, a single other cloud deployment 514 is included for ease of illustration but in no way represents a limitation of the embodiments described herein. In fact, most embodiments will include multiple other cloud deployments 514 that may be monitored and learned from.

[00717] Readers will appreciate that while many of the embodiments described above relate to embodiments where cross-customer learnings are used to generate recommendations for changes to the first cloud deployment 508, in other embodiments the cross-customer learning may be used to initiate, based on the normal behavior for the one or more other cloud deployments, a change to the first cloud deployment. Initiating such changes may be carried out, for example, by initiating one or more remediation workflows that implement the changes. As was the case with recommendations, initiating a change to the first cloud environment may be based on the normal behavior for similar cloud deployments, based on the normal behavior for the highly rated cloud deployments, based on the cloud deployments to exclude from consideration when recommending changes to the first cloud deployment, and so on.

[00718] For further explanation, Fig. 9 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 9 may be carried out by the systems described above (also referred to as a 'data platform' above). As such, the systems described above may include computer program instructions executing on computer hardware, virtualized hardware, or some other execution environment (e.g., one or more containers, one or more serverless compute instances), where the computer program instructions carry out the steps described in Fig. 9 when the computer program instructions are executed.

[00719] The example depicted in Fig. 9 includes a plurality of cloud deployments, including a first cloud deployment 910 and one or more additional cloud deployments 912, which in this illustration includes three cloud deployments 914a, 914b, 914n, although more or fewer cloud deployments may be included in the set of additional cloud deployments 912 in other embodiments. Such cloud deployments may be embodied as described above and may include a variety of components such as, for example, software applications, storage resources, computing resources, networking resources, and other resources. In each cloud deployment, each of the resources may be delivered as services provided by a public cloud, private cloud, hybrid cloud, and so on.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00232

[00720] The example method depicted in Fig. 9 includes identifying 902, for at least a portion of a first cloud deployment 910, one or more additional cloud deployments 912 to utilize for cross-customer learning. Cross-customer learning, as the phrase is used here, can generally be described as the process of learning about the cloud deployments of different customers, different organizations, or some other entity for the benefit of tailoring another entity's cloud deployment. For example, the cloud deployments of Customer A, Customer B, Customer C, Customer D, and Customer F may be evaluated and the information gathered about their deployments may be leveraged to help shape the cloud deployment of Customer F. In such an example, lessons learned by other customers may be used to help guide a particular customer's design, deployment, and management of their cloud deployments, which may be particularly useful for organizations that are pivoting to the cloud from on-premises based deployments.

[00721] Identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may be carried out, for example, by identifying cloud deployments for other customers that are in similar industries as the customer associated with the first cloud deployment 910, by identifying highly rated (as described above) cloud deployments, by identifying cloud deployments (or portions thereof) that are similar to the first cloud deployment 910 (or some portion thereof), or in some other way. In some embodiments, identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may be carried out using machine learning techniques as one or more machine learning models may be used to identify cloud deployments that are similar to the first cloud deployment 910, to identify best-of-breed cloud deployments, and so on. In such a way, identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may result in a subset of curated, relevant, and/or exemplary cloud deployments being utilized for cross-customer learning rather than using the entire set of available cloud deployments being utilized for cross-customer learning (although some embodiments could utilize all cloud deployments for cross-customer learning).

[00722] The example method depicted in Fig. 9 also includes receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912. The one or more actions associated with the additional cloud deployments 912 may be embodied as actions taken by components within the additional cloud deployments 912. For example, a particular software service accessing a data store may be an example of an action that is associated with the additional cloud deployments 912, especially where the software service and the data store are components of a particular additional cloud deployment 914a. Likewise, a message, request, or other form of data communications that are exchanged between the additional cloud deployments 912 and actors that are external to the additional cloud deployments 912 may be an example of an action that is associated with the additional cloud deployments 912. For example, a server

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00233

that is external to any of the additional cloud deployments 912 may send requests to access some software service within a particular additional cloud deployment 914a.

[00723] In other embodiments, the one or more actions associated with the additional cloud deployments 912 may include one or more user interactions with the additional cloud deployments 912. Such user interactions can include a description of which resources are accessed and/or utilized by particular users, particular business units (e.g., the finance department of a business, the engineering department of a business), particular personas (e.g., a system administrator, a software developer, a human resources manager), and so on. The one or more actions associated with the additional cloud deployments 912 may also include, for example, one or more interactions involving external applications, external computing devices, or similar external actor and the additional cloud deployments 912. Such user interactions can include, for example, a description of which resources within the additional cloud deployments 912 are accessed and/or utilized by particular external application, what IP address is associated with an external entity that is communicating with the additional cloud deployments 912, a description of the specific requests that an external actor is issuing to the resources within the additional cloud deployments 912, and so on.

[00724] In the example method depicted in Fig. 9, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 may be carried out by receiving (directly or indirectly) such information 916 from the additional cloud deployments 912 themselves. In fact, the additional cloud deployments 912 may be queried for such information. In other embodiments, the information 916 describing one or more actions associated with the additional cloud deployments 912 may be retained by the systems described above as part of monitoring the additional cloud deployments 912. In such an example, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 may be carried out by querying a data repository (e.g., a data warehouse) that contains such information that was gathered while monitoring the additional cloud deployments 912. Readers will appreciate that information 916 may be received 904 describing other types of actions associated with the additional cloud deployments 912.

[00725] The example method depicted in Fig. 9 also includes receiving 906 information 918 describing configurations associated with the additional cloud deployments 912. The information 918 describing configurations associated with the additional cloud deployments 912 may be embodied, for example, as information describing how components within a cloud deployment are organized (including which other components they may communication with), information describing what permissions are granted to various users, information describing the manner in which internal (i.e., occurring entirely within the cloud deployment) data

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00234

communications and/or external (i.e., occurring at least partially external to the cloud deployment) data communications are carried out, and so on. Such information 918 may be received 906 in a similar manner as described above with respect to step 904.

[00726] The example method depicted in Fig. 9 also includes identifying 908, based on the configurations and the one or more actions, one or more configurations to adopt for the first cloud deployment 910. Readers will appreciate that by identifying 908 one or more configurations to adopt for the first cloud deployment 910 in such a way, the configuration of the first cloud deployment 910 may be influenced by the monitoring of the additional cloud deployments 912. In such an example, once the configurations that the first cloud deployment 910 should adopt have been identified 908, the first cloud deployment 910 (or associated entities) may be reconfigured to implement such configurations.

[00727] Identifying 908 one or more configurations to adopt for the first cloud deployment 910 may be carried out, for example, by identifying configurations in the additional cloud deployments 912 that were effective in dealing with various threats, securing vulnerabilities, or otherwise contributed to a healthy cloud deployment. Some configurations associated with the additional cloud deployments 912 may have been put in place in response to some action. For example, a configuration to always backup data stored in a first storage service (e.g., AWS S3) to a second, distinct storage service (e.g., AWS Glacier) may have been put in place in response to some attack that was directed to a particular cloud deployment's 914a S3 buckets. In such an embodiment, if the first cloud deployment 910 also leverages S3 as its object store, it may be desirable for the first cloud deployment 910 to adopt a configuration setting that would result in its S3 buckets being backed up to Glacier so that the first cloud deployment 910 can be ready to survive a similar attack that was experienced by the particular cloud deployment 914a.

[00728] For further explanation, Fig. 10 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 10 is similar to the example depicted in Fig. 9, as the example depicted in Fig. 10 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

[00729] In the example method depicted in Fig. 10, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 can include receiving 1002 information describing a security threat to one or more of the additional cloud deployments 912. The security threat may be embodied, for example, as a ransomware attack that was

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00235

directed to one or more of the additional cloud deployments 912, as a denial of service attack that was directed to one or more of the additional cloud deployments 912, as an SQL injection attack that was directed to one or more of the additional cloud deployments 912, or as some other security threat. The information describing a security threat to one or more of the additional cloud deployments 912 can include, for example, information describing where an attack originated from, information describing the components within the cloud deployment that were targeted by an attack, information describing data access patterns that were associated with an attack, or some other information that would be useful in detecting subsequent security threats.

[00730] In the example method depicted in Fig. 10, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can include receiving 1004 information describing configuration settings used to combat the security threat. The configuration settings used to combat the security threat may be embodied, for example, as configuration settings that caused the additional cloud deployments 912 to blacklist certain IP addresses or network domains, as configuration settings that caused the additional cloud deployments 912 to close vulnerabilities that allowed an attack to succeed, or some other configuration that would be useful in detecting/preventing/mitigating a security threat.

[00731] In the example method depicted in Fig. 10, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 can also include receiving 1006 information describing a detected vulnerability associated with one or more of the additional cloud deployments 912. The detected vulnerability may be embodied, for example, as a vulnerability to some known security threat, as a vulnerability to some data loss event, as a data breach vulnerability, or as some other vulnerability. The information describing a detected vulnerability of one or more of the additional cloud deployments 912 can include, for example, information describing vulnerable components, information describing data communications protocols, endpoints, or other data communications components that expose vulnerabilities, or some other information that would be useful in detecting a vulnerability.

[00732] In the example method depicted in Fig. 10, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can also include receiving 1008 information describing configuration settings used to address the vulnerability. The configuration settings used to address the vulnerability may be embodied, for example, as configuration settings that caused the additional cloud deployments 912 to blacklist certain IP addresses or network domains, as configuration settings that caused certain components within the additional cloud deployments 912 to not connect to external data communications networks, as configuration settings that caused certain components to be accessible only by using enhanced

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00236

authentication protocols, or as some other configuration that would be useful in detecting, preventing, or otherwise mitigating a vulnerability.

[00733] For further explanation, Fig. 11 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 11 is similar to the examples depicted in Fig. 9 and Fig. 10, as the example depicted in Fig. 11 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

[00734] In the example method depicted in Fig. 11, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can include receiving 1102 information describing permissions for one or more users of the additional cloud deployments 912. The one or more users of the additional cloud deployments 912 may be embodied, for example, as individual users (e.g., Bob Smith), as a collection of user (e.g., Finance Group, Marketing Group), and so on. The one or more users of the additional cloud deployments 912 may also be embodied as different personas within an organization, where each persona is associated with a certain role. For example, there may be 'software developer' personas, 'database administrator' personas, 'customer support' personas, and many others. Receiving 1102 information describing permissions for one or more users of the additional cloud deployments 912 may be carried out, for example, by receiving (directly or indirectly) information from one or more of the additional cloud deployments 912 that described what resources a particular user/persona/group can access, what level of privileges they have with respect to a particular resource, and any other information describing the user/persona/group's privileges. In fact, such information may be retained in a data warehouse during monitoring of the additional cloud deployments 912 and received 1102 by querying such a data warehouse.

[00735] The example method depicted in Fig. 11 also includes determining 1104, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment 910 are over-permissioned. A particular user may be over-permissioned in the sense that the user has privileges or permissions beyond those that should be associated with the user. Consider the example of two distinct personas within an organization: 1) a software engineer, and 2) an accounts payable administrator. In such an example, the software engineer might have sufficient privileges to access a code repository, a dev/test environment, a code documentation tool, and so on. The accounts payable administrator, however, might have sufficient privileges to access bill paying software, a financial ledger, and so. The software developer having access to bill paying software or the accounts payable

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00237

administrator having access to a code repository may be examples of the users being over-permissioned, as the user has access to resources that they should not be able to access. Such situations can create vulnerabilities as an over-permissioned user (or a malicious actor using the over-permissioned user's credentials) may intentionally or unintentionally perform some harmful act that they would not be capable of performing if they weren't over-permissioned.

[00736] Determining 1104 that one or more users of the first cloud deployment 910 are over-permissioned may be carried out, for example, by examining the information describing permissions for one or more users of the additional cloud deployments 912 to determine whether similar users in the additional cloud deployments 912 have different/fewer privileges. As such, a user in a first cloud deployment 910 may be determined to be over-permissioned if other cloud deployments 912 are giving similar users fewer permissions than such a user has in the first cloud deployment 910.

[00737] In the example method depicted in Fig. 11, identifying 908 one or more configurations to adopt for the first cloud deployment 910 can include identifying 1106 a reduced privilege level to give to the one or more users. Identifying 1106 a reduced privilege level to give to the one or more users may be carried out, for example, by giving an over-permissioned user of the first cloud deployment 910 privileges that are similar to those held by similar users in the additional cloud deployments 912.

[00738] For further explanation, Fig. 12 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 12 is similar to the examples depicted in Figs. 9-11, as the example depicted in Fig. 12 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

[00739] The example method depicted in Fig. 12 also includes receiving 1202 information 1204 describing one or more deployment processes associated with the additional cloud deployments 912. The information 1204 describing one or more deployment processes associated with the additional cloud deployments 912 can include, for example, information describing how a particular component within an additional cloud deployment was created (e.g., who requested that the component be created, when was the request to create the component issued). In addition, in some embodiments such information 1204 can include information describing the software development processes that were used to develop software that will be deployed in a particular cloud deployment (e.g., was a code repository used, who has access to the repository, who committed the last change to some code before deployment). In such a way, the information

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00238

1204 describing one or more deployment processes associated with the additional cloud deployments 912 can describe the processes that are being implemented in other cloud deployments. Such information 1204 may be used, for example, to determine whether the development and deployment processes associated with the first cloud deployment 910 are abnormal, to use as the basis for making recommended changes to the development and deployment processes associated with the first cloud deployment 910, to initiate remediation workflows to back out changes to the first cloud deployment 910 that deviated from normal activity or violated some policy, and so on.

[00740] The example method depicted in Fig. 12 also includes identifying 1206 abnormally configured components in the first cloud deployment 910. Abnormally configured components in the first cloud deployment 910 may be embodied as components that are configured in a way that deviates from typical configurations observed by monitoring the additional cloud deployments 912. Monitoring the additional cloud deployments 912 may reveal, for example, that a connection to a server that is contained in another cloud deployment 914a should always go through an authentication server that is contained in another cloud deployment 914a. In such an example, if an examination of the first cloud deployment 910 reveals that an authentication server has been bypassed by a connection to a server, this may be an indication of a breach or misconfiguration regardless of customer's network topology. In such an example, the component (i.e., the server) may be identified 1206 as being abnormally configured based on its deviation from typical configurations observed in the additional cloud deployments 912.

[00741] Although the examples described above relate to embodiments where one or more configurations to adopt for the first cloud deployment 910 are identified 908 using the information described above, in other embodiments such information may be used for other purposes. For example, information gained by observing additional cloud deployments 912 may be leveraged to identify best-of-breed deployments, similar deployments, and so on.

[00742] In some embodiments, the distributions that identify 'normal' behavior for a particular cluster set may be used for a variety of purposes. For example, anomaly detection may be performed by identify members of the cluster set that are operating outside of a particular typical distribution. Likewise, best practices may be identified using distributions and members of the cluster set that are not adhering to best practices may be identified if they are operating outside of a particular typical distribution. Vulnerabilities may also be identified using distributions, for example, by identifying members of the cluster set that have over-privileged users or components that may be able access things that they should not be able to access. Readers will appreciate that that deviation from an established baseline of normal behavior, normal activity, normal configuration, or other form of normal operation may be indicative of many other things, all of

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00239

which may be detected through the usage of the distributions described above paired with the monitoring of particular clusters.

[00743] In some embodiments, the distributions that identify 'normal' behavior for a particular cluster set may be used to not only detect threats, vulnerabilities, compromise, and things of that nature, but the distributions that identify 'normal' behavior for a particular cluster set may be used to optimize the operation or configuration of a particular cluster. For example, cloud deployments that utilize fewer resources may be examined and characteristics that are common across such the cloud deployments may be identified as being high efficiency characteristics. Likewise, cloud deployments that are the subject of fewer alerts may be examined and characteristics that are common across such the cloud deployments may be identified as being high efficiency characteristics.

[00744] In some embodiments, historical information may also be retained and utilized to show the trajectory of a particular one or more clusters. For example, historical information may be used to compare the current state of a particular cluster (as measured by one or more quantifiable characteristics associated with the cluster) with a previous state of the cluster such that trends and/or trajectories may be identified. Consider an example in which a particular characteristic associated with a cluster identifies how many users are accessing the cluster. Historical information associated with the cluster may be compared to current (or most recent) information associated with the cluster to determine, for example, that more or less users are accessing the cluster, that the number of users that are accessing the cluster is increasing or decreasing at a certain rate, and so on. In some situations, some changes may be acceptable (e.g., if the cluster represents an interface to an online store, and increase in the number of users accessing the online store may be perfectly acceptable given an expansion in the organization's customer base) whereas in other situations changes may be troubling (e.g., if the cluster represents a source code repository and more users are accessing the repository in spite of a contraction in the number of developers that are employed by an organization) and my require alerts, further investigation, or some other remediation workflow.

[00745] In some embodiments, historical information may also be retained and utilized to show the trajectory of a particular cluster relative to other members of a cluster set. For example, historical information may be used to compare the failure rate of particular cluster (as measured by one or more quantifiable characteristics associated with the cluster) over time with the failure rate over time of other clusters in the cluster set. Such a comparison may reveal, for example, that a particular cluster was previously failing at a rate that was in line with the failure rate of other members of the cluster set, but that the particular cluster has more recently been failing more/less than other members of the cluster set. As such, through the use of such techniques a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00240

determination may be made as to whether a particular cluster is becoming more/less healthy than other members of the cluster set, more/less secure than other members of the cluster set, more/less efficient than other members of the cluster set, more/less reliable than other members of the cluster set, more/less compliant with relevant regulations than other members of the cluster set, slower/faster than other members of the cluster set, and so on. Such an analysis may be part of identifying best-of-breed deployments, best practices, providing remediation actions, providing recommendations, or utilized for a variety of other purposes.

[00746] Although the examples described above relate to embodiments where components within a cloud deployment are analyzed, monitored, or otherwise observed, in other embodiments the techniques described herein may be applied to other entities. For example, the techniques described above may be applied to analyze, monitor, or otherwise observe different personas with an organization, different users with an organization, different user groups with an organization, and so on. Using such techniques, 'normal' behaviors for a particular persona (e.g., a database administrator, a network administrator) can be identified, 'normal' behaviors for a particular user group (e.g., users that are part of an organization's finance department, users that are part of an organization's engineering department) can be identified, and so on.

[00747] In some embodiments, by identifying 'normal' behaviors (i.e., those behaviors that are consistent with standard distributions for particular characteristics associated with an entity) with different personas, different users, different user groups, or other entities, abnormal behavior may also be identified. Consider an example in which a group of users is identified as a cluster by virtue of those users accessing the same set of applications (e.g., all of the users access a set of applications that are finance-related applications). Further assume that similar clusters are identified for other customers, such that a cluster set may be formed. In such an example, a distribution may be identified for the set of applications that members of the cluster typically access. For example, each member of the cluster set may typically access an accounting application, a spreadsheet application, a payroll application, and so on. If one member of the cluster set consists of one or more users that also access an organization's source code repository, such behavior may be determined to be outside of the typical distribution for members of the cluster set. In response to making some determination, alerts may be generated, access to the source code repository may be blocked for the users in the cluster, a remediation workflow may be initiated, or some other action may be taken.

[00748] In some embodiments, the techniques described above may be particularly useful for identifying over-privileged users, user groups, personas, or other entity. Identifying over-privileged users, user groups, personas may be carried out, at least in part, by identifying what set of privileges is 'normal' for a particular entity to have based on evaluating what privileges are

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00241

given to similar or identical members of a cluster set. That is, an evaluation may be made as to what privileges are given to users, user groups, personas, or other entity by one or more other customers. If the set of privileges given to a particular customer's users, user groups, personas, or other entity are not consistent with (e.g., the set of privileges are much greater than) the privileges given to similar or identical users, user groups, personas, or other entity of another customer (i.e., the other members of the cluster set), a determination may be made that the particular customer's users, user groups, personas, or other entities are over-privileged. In some embodiments, an evaluation as to whether a user, user group, persona, or other entity is over-privileged may also include identifying privileges that the user has and comparing that with the privileges that the user actually utilizes. If some privileges are never used, this may be taken as an indication that the user is over-privileged.

[00749] In some embodiments, the cross customer techniques described above may be used to provide additional context to issues identified in a particular customer's cloud deployment, to provide recommendations to a particular customer, or even to drive remediation actions. Consider an example in which a particular threat was detected in a first customer's cloud deployment, where the threat turned out to be a ransomware attack, which may in some embodiments include an encryption component and/or a data theft or leakage component. In such an example, if an identical (or sufficiently similar, following a general recognized pattern or 'fingerprint') threat is detected in a second customer's cloud deployment, additional context may be provided by including information in an alert that is delivered to the second customer that indicates that the threat matches the profile of a ransomware attack that was detected in the first customer's cloud deployment. In fact, information describing the remedial actions (e.g., disabling encryption, increasing the frequency of backups, locking down a backup system, blocking transmission of data externally, etc.) that were taken by the first customer may even be included in the alert to the second customer or otherwise recommended to the second customer. Furthermore, if many customers had experienced the same attack and the data platform could determine with sufficient certainty that the second customer's cloud deployment was experiencing the same attack, workflows may be automatically initiated to carry out various remedial actions.

[00750] In some embodiments, the way other customers investigated or responded to a particular alert may also be used when presenting alerts to a particular customer. Consider an example in which a particular threat was detected in ten different customer's cloud deployments. In such an example, assume that 9 of the 10 customers ignored the alert. In such an example, the conditions that are indicative of the particular threat are detected in a particular customer's cloud deployment, when raising the alert to the particular customer, information may be included in the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00242

alert indicating that most other customers ignored the alert. In such an embodiment, it is the behavior of users of the data platform – not components in a cloud deployment – that is being monitored and evaluated for the benefit of other customers. In other embodiments, information indicating that customers do or do not ignore a particular alert may be used when scoring or ranking the alert. For example, alerts that are ignored by most customers may be ranked as less critical than alerts that are acted upon by most customers. In addition, the particular security stance of a particular customer may be utilized when determining the extent to which their usage of the data platform should be utilized when guiding other customers. If a first customer has relatively poor security practices, for example, the first customer's usage of the data platform may be ignored (or given less weight) for the purposes of guiding other customers. If a second customer has relatively good security practices, however, the second customer's usage of the data platform may be taken into consideration (or given more weight) for the purposes of guiding other customers. Likewise, if users that ignored the alert experienced a security breach while users that did not ignore the alert did not experience a security breach, such an outcome could be taken into consideration when ranking the alert, determining what information to include in the alert, and so on.

[00751] In addition to using other customers interactions with the data platform to score, rank, suppress, or provide context for the alerts provided to other customers, the remediation actions taken by other customers may also be used for guiding a particular customer. Consider an example in which a particular threat was detected in ten different customer's cloud deployments. In such an example, assume that 9 of the 10 customers investigated the threat by accessing a particular customer support page. In this example, if the conditions that are indicative of the threat are detected in a particular customer's cloud deployment, information may be included in an alert indicating that most other customers that received the alert investigated the alert by accessing the particular customer support page. Likewise, if other customers successfully resolved the issue, information may be included in an alert indicating the solution that was implemented by other customers.

[00752] In some embodiments, the way customers investigated or responded to a particular alert may be used with other information to determine whether the customer's responses were actually correct. Consider an example in which an alert is sent to 10 customers identifying a vulnerability that would allow crypto miners to use the customer's resources for solving complex computational problems for the purposes of acquiring cryptocurrency. In such an example, assume that 9 of 10 customers ignored the alert. Further assume that in this example, the 9 customers that ignored the alert subsequently had their cloud resources utilized by hackers for crypto mining, whereas the 1 customer that did not ignore the alert did not suffer such an attack.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00243

In such an example, the data platform described above should not cease issuing alerts when detecting this vulnerability by virtue of 9 of 10 customers ignoring the alert. Instead, by taking the ultimate outcome for each customer into consideration, rather than suppressing the alert, the data platforms should respond by taking actions that would make it more likely that customers would not ignore these alerts. For example, the severity level of the alert may be raised, a user of the data platform may be required to confirm receipt of the alert, contextual information could be included in the alert indicating that recipients that do not respond to the alert end up having their systems hijacked by crypto miners, or some other action may be taken.

[00753] In some embodiments, a customer's interactions may be analyzed to improve how alerts are delivered to a customer. For example, if an evaluation of customer interactions with the data platform indicates that most customer ignore alerts issued between 11:00 PM – 7:00 AM whereas most customers take action in response to alerts issued between 7:01 AM – 10:59 PM, then alerts that are issued between 11:00 PM – 7:00 AM may be reissued between 7:01 AM – 10:59 PM. Likewise, if a particular class of alerts (e.g., those related to vulnerability threats) are largely acted upon whereas another class of alerts (e.g., those related to a lack of compliance with regulatory requirements) are largely ignored, the manner in which alerts are generated may be altered. For example, alerts that are related to a lack of compliance with regulatory requirements may be issued to additional users, such as some user that is designated as having a compliance persona within a customer's organization. In other embodiments, other aspects of the customer's interactions with the data platform may be used to improve the manner in which the data platform interacts with customers (e.g., what type of devices received alerts, were alerts delivered in a primary or secondary window, do the parties that the alerts were sent to have other obligations at the time of alert as determined from an inspection of their calendar, and so on). While the example described above related to embodiments where the interactions of multiple customers with the data platform is analyzed, in other embodiments a single customer's interaction with the date platform may serve as the basis for altering the way that the data platform interacts with the customer.

[00754] In some embodiments, the same or additional cross customer learning techniques may be applied to earlier stages of a software development pipeline and even before an actual cloud deployment is in place for a particular customer. Stated differently, the same or additional cross customer learning techniques may be applied to things other than a deployed system. For example, cross customer learning may be applied to development processes, testing processes, deployment processes, and so on

[00755] Readers will appreciate that developing, testing, and deploying software in cloud environment comes with a few requirements that were not always present in software

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00244

development. In particular, software is always expected to be running but the software is also expected to continue to be updated with fixes, new features, or other improvements. As such, it is not desirable to cease running a software application, install a new version, and begin running the new version. As a result of these changes to the software development paradigm, the software development processes, software testing and validation processes, and deployment processes tend to be fairly independent of each other. In some embodiments, each of these processes may be analyzed, monitored, or otherwise observed by the data platforms described herein. Because each of these processes are analyzed, monitored, or otherwise observed by the data platforms described herein, an opportunity exists to leverage cross customer learning to these processes.

[00756] In some embodiments, the development processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. For example, interactions with a code repository may be monitored, permissions granted to each user of the code repository may be monitored, the number of people that are checking in and checking out code may be monitored, the extent to which code revisions are documented may be monitored, and so on. In such an example, the development processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

[00757] In some embodiments, the testing processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. For example, the number of tests that are run may monitored, the type of tests that are run may be monitored, the number of people that are running tests and evaluating test results may be monitored, the processes through which the results of testing are communicated with developers may be monitored, and so on. In such an example, the testing processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

[00758] In some embodiments, the results of testing may be carried forward and utilized when an application is deployed. Consider an example in which testing reveals that the code performs some action that appears to create a possible vulnerability, but further testing and validation

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00245

reveals that a vulnerability is not created. When this piece of code is subsequently deployed, it presumably will perform the same action (i.e., the action that was flagged during testing) that appears to create a possible vulnerability. Rather than raising an alert, initiating some investigative action, initiating a remedial action, or responding in a similar way, by carrying forward the knowledge gained during testing (i.e., that a vulnerability was not, in fact, created) the possible vulnerability may be ignored based on the conclusions reached during testing.

[00759] In some embodiments, the deployment processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. In such an example, the deployment processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

[00760] In some embodiments, some forms of static analysis may be used (in conjunction with other features of the data platform) to detect anomalies, vulnerabilities, threats, misconfigurations, violations of regulatory requirements, and many other things. Consider an example in which a cloud deployment is deployed using IaC. In such an example, one or more configuration files may be examined, and the state of the cloud deployment may be monitored to identify situations in which the state of the cloud deployment drifts from the configuration of the cloud deployment that was described in the configuration file. As such, the intended configuration of the cloud deployment (at least as expressed in one or more configuration files) may be used as a baseline to measure the current cloud deployment, such that alerts may be issued or other remediation workflows may be initiated when a customer's cloud deployment deviates from its codified state.

[00761] In some embodiments, other forms of code other than an IaC configuration may be examined to detect anomalies, vulnerabilities, threats, misconfigurations, violations of regulatory requirements, and many other things. For example, the source code that has been deployed in a customer's environment may be examined to determine all the things that the code could do. This information may be compared to a polygraph for the customer's cloud deployment, which identifies all things that a customer's cloud deployment does do, as learned by monitoring and observing the customer's cloud deployment. Consider an example in which a polygraph for a customer's cloud deployment indicates that a first microservice in their cloud deployment only communicates with other internal microservices. In such an example, however, assume that the source code for the first microservice includes a messaging library that it uses to communicate with the other internal microservices. Further assume in this example, however, that examining

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00246

the source code for the first microservice reveals that the messaging library also includes functions that enable a user of the library to send messages to recipients on an external network using standard internet protocols (e.g., TCP/IP, HTTPS, and so on). In such an example, although the polygraph for the customer's cloud deployment reveals that the first microservice only communicates with other internal microservices, the presence of functions that could be used for communications with external services, machines, and other entities may be undesirable. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if executed, would rise to the level of anomalous or otherwise unusual activity. For example, the data platform may prompt the customer to delete or disable the library functions described above that enable undesirable data communications.

[00762] In another example, assume that some source code has all the necessary capabilities (e.g. hashing functions, mathematical calculations, etc.) to mine cryptocurrency, as the mining process may only require performing standard computations. During a crypto miner attack, however, the source code may be subverted such that the code runs in a loop to perform the standard computations required for mining cryptocurrency. An evaluation as to what functions the source code can perform may therefore not reveal anything concerning but monitoring the actual operation of the source code may reveal that the source code is operating in an undesirable way. Likewise, static analysis may reveal things like a non-incrementing counter or other mechanism that would result in an infinite loop or similar operation. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if or when executed, rise to the level of anomalous or otherwise unusual activity.

[00763] In another example, assume that some source code has all the necessary capabilities to carry out the steps required for a ransomware attack, as the ransomware attack may only require performing standard operations like reading data, encrypting data, sending data, communicating externally, and so on. During a ransomware attack, however, the source code may be subverted such that the code does far more encryption than would be expected in a typical code module. An evaluation as to what functions the source code can perform may therefore not reveal anything concerning but monitoring the actual operation of the source code may reveal that the source code is operating in an undesirable way. Likewise, static analysis may reveal that all paths through the source result in data being encrypted, or that all paths through the source code results in some atypical pattern (e.g., equal amounts of reading data, encrypting data, and writing data, or unusual external data flows), which may rise to the level of being unusual or concerning behavior. As such and in accordance with some embodiments of the present disclosure, the data

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00247

platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if or when executed, rise to the level of anomalous or otherwise unusual activity.

[00764] In some embodiments, the data platforms described above may be used to identify discriminates between two entities that may otherwise appear to be similar or identical. In the examples described herein, 'discriminates' may be embodied as characteristics of entities that, when not similar or identical, prevent the entities from being similar or identical. For example, many customers may have many Java processes. If the data platform were to identify the set of things that each Java process does for each customer and identify that set of things as the 'normal' or acceptable set of things that any Java process was allowed to do, this set would be far too large. As such, the data platform may be configured to look at things like command line arguments and know that one or more Java processes with one set of jar files and command line arguments is actually a separate program from one or more Java processes with another set of jar files and command line arguments. In other words, the data platform may identify discriminates (in the example, a first discriminate being distinct command line arguments and a second discriminate may be that the jar files for each process are distinct) between two sets of Java processes to determine that the Java processes are not actually similar or identical entities. As such, any attempt to engage in cross customer learning based on a first set of Java processes in a first customer's environment and a second set of Java processes in a second customer's environment may result in undesirable outcomes. Through the use of discriminates, however, a decision can be reached that these two sets of Java processes are not related and attempts to learn through an examination of these distinct entities would be undesirable.

[00765] In some embodiments, the usage of discriminates described above may be extended to other entities. For example, if a first user in a first customer's environment is a database administrator and a second user in a second customer's environment is also a database administrator, these users may initially be determined to be similar enough such that cross customer learning can take place. In such an example, however, if an examination of their activities, privileges, or something else reveals that they are not actually occupying the same roles, cross customer learning with respect to these two users may be disabled. In such embodiments, the data platform described above, and models leveraged by such a data platform may be used to identify discriminates that may be used to decouple multiple users, applications, microservices, devices, or other entities that would otherwise be candidates for cross customer learning.

[00766] In some embodiments, the data platforms may be configured to analyze, monitor, or otherwise observe environments other than cloud deployments. In fact, the data platforms

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00248

described here could apply the principles and techniques described herein to any environment such as, for example, an on-premises environment, a hybrid cloud environment, or some special purpose environment. As one example of a special purpose environment, consider an example in which the data platforms described herein are used to analyze, monitor, or otherwise observe a container orchestration environment such as a Kubernetes cluster (which may be deployed on-premises, in a public cloud, or in some other way). In such an example, the data platform may be configured to ingest Kubernetes audit logs via one or more agents or in some other way. Through the ingestion and subsequent analysis of such audit logs, the data platform may model normal behaviors of a Kubernetes cluster, normal behavior of a cluster administrator, and so on. As described above, any deviations from such normal behaviors may result in an alert being generated or some other remediation workflow being initiated. For example, if the ingestion and subsequent analysis of audit logs revealed that workloads are deployed and deleted according to some pattern, a customer deploying or deleting a workload in a manner that is inconsistent with the identified pattern may result in an alert being generated or some other remediation workflow being initiated. In fact, by evaluating audit logs from multiple customer's Kubernetes clusters, cross customer learning can be carried out to help define normal behavior for a Kubernetes cluster, a Kubernetes administrator, or some other entity associated with a Kubernetes cluster. For example, evaluating audit logs from the Kubernetes deployment of multiple customers may reveal that one entity (presumed to be a Kubernetes administrator, or related group of such administrators) is generally responsible for creating and deleting nodes from the cluster whereas another entity (perhaps a developer) is responsible for deploying new versions of the code that is executing in a container that is supported by the cluster. In such an example, if a single entity (or related group of entities) is observed creating nodes and modifying the code that is executing on a node by deploying a new container, an alert may be generated or some other remediation workflow may be initiated as a consequence of observing this atypical pattern.

[00767] One or more embodiments may be described herein with the aid of method steps illustrating the performance of specified functions and relationships thereof. The boundaries and sequence of these functional building blocks and method steps have been arbitrarily defined herein for convenience of description. Alternate boundaries and sequences can be defined so long as the specified functions and relationships are appropriately performed. Any such alternate boundaries or sequences are thus within the scope and spirit of the claims. Further, the boundaries of these functional building blocks have been arbitrarily defined for convenience of description. Alternate boundaries could be defined as long as the certain significant functions are appropriately performed. Similarly, flow diagram blocks may also have been arbitrarily defined herein to illustrate certain significant functionality.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00249

[00768] To the extent used, the flow diagram block boundaries and sequence could have been defined otherwise and still perform the certain significant functionality. Such alternate definitions of both functional building blocks and flow diagram blocks and sequences are thus within the scope and spirit of the claims. One of average skill in the art will also recognize that the functional building blocks, and other illustrative blocks, modules and components herein, can be implemented as illustrated or by discrete components, application specific integrated circuits, processors executing appropriate software and the like or any combination thereof.

[00769] While particular combinations of various functions and features of the one or more embodiments are expressly described herein, other combinations of these features and functions are likewise possible. The present disclosure is not limited by the particular examples disclosed herein and expressly incorporates these other combinations.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00250

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	ÚLFAR ERLINGSSON
Serial No.:	17/671,199
Atty Docket No.:	L0041B1CIP2
Group Art Unit:	2442
Filed:	02/14/2022
Customer No.:	182115
Confirmation No.:	1070
Examiner:	ALINA A BOUTAH
Title:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Mail Stop: Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

RESPONSE TO NOTICE TO FILE CORRECTED APPLICATION PAPERS DATED JUNE 05, 2023

This is a Response to the Notice to File Corrected Application Papers dated June 05, 2023 (hereafter "the Notice"), a copy of which is attached. Applicants present the following amendments and remarks demonstrating that the case is in condition for allowance.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00251

AMENDMENT

Amendments To The Specification:

Please replace the original specification with the attached substitute specification. A clean and markup copy is included.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00252

REMARKS

The Notice requires Applicant provide a brief description of each drawing figure presented. To satisfy the requirement, Applicant submits herewith a substitute specification (both clean and markup) to include a brief description of Fig. 12.

Applicant asserts no new matter has been added.

The Commissioner is hereby authorized to charge or credit Deposit Account No. 50-3082 for any fees required or overpaid.

Respectfully submitted,

Date: July 21, 2023

By: /Edward Lenart/

Edward Lenart, Reg. No. 63,665
Kennedy Lenart Spraggins LLP
Tel. (512) 472-9881
Fax (512) 472-9887
ATTORNEY FOR APPLICANT

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00253

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

182115 7590 06/05/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER	
BOUTAH, ALINA A	

ART UNIT	PAPER NUMBER
2442	

NOTIFICATION DATE	DELIVERY MODE
06/05/2023	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

COPY

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

cofficeaction@appcoll.com
katc@klspatents.com
office@klspatents.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

Application No. : 17671199
Applicant : Erlingsson
Filing Date : 02/14/2022
Date Mailed : 06/05/2023

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Notice of Allowance Mailed

COPY

This application has been accorded an Allowance Date and is being prepared for issuance. The application, however, is incomplete for the reasons below.

Applicant is given two (2) months from the mail date of this Notice within which to respond. This time period for reply is extendable under 37 CFR 1.136(a) for only TWO additional MONTHS.

The informalities requiring correction are indicated in the attachment(s). If the informality pertains to the abstract, specification (including claims) or drawings, the informality must be corrected with an amendment in compliance with 37 CFR 1.121 (or, if the application is a reissue application, 37 CFR 1.173). Such an amendment may be filed after payment of the issue fee if limited to correction of informalities noted herein. See Waiver of 37 CFR 1.312 for Documents Required by the Office of Patent Publication, 1280 Off. Gaz. Patent Office 918 (March 23, 2004). In addition, if the informality is not corrected until after payment of the issue fee, for purposes of 35 U.S.C. 154(b)(1)(iv), "all outstanding requirements" will be considered to have been satisfied when the informality has been corrected. A failure to respond within the above-identified time period will result in the application being ABANDONED.

See attachment(s).

A copy of this notice MUST be returned with the reply. Please address response to "Mail Stop Issue Fee, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450".

/Chris Harmon/
Publication Branch
Office of Data Management
(571) 272-4200

Application No. 17671199

IDENTIFICATION OF SPECIFICATION/DRAWING INCONSISTENCIES

☐ On Page of the specification there is a brief description of FIG. , but the drawings filed do not include a drawing with that designation. Applicant must respond either by supplying the omitted drawing or by amending the specification to remove all references to that drawing.

☒ The drawings filed 02/14/2022 include FIG. 12, but the specification's brief description of the drawings does not describe a drawing with that designation. Applicant must respond either by amending the specification to add a brief description of that drawing or by correcting the drawings to remove the drawing in question.

☐ In the utility application, drawings are present in the application and are referred to in the detailed description of the invention, but the specification does not contain a brief description of the several views as required by 37 CFR 1.74 and 37 CFR 1.77(b)(9).

☐ The design application does not contain a description of the figure or figures of the drawing as required by 37 CFR 1.153(a) and 37 CFR 1.154(b)(4).

☐ In the plant application, the specification does not contain a brief description of the drawing as required by 37 CFR 1.163(c)(8).

☐ Page of the specification refers to FIG. , but no drawing with that designation is described in the brief description of the drawings and no drawing with that designation is present in the application. Applicant must respond either by amending the specification to remove all references to that drawing, or by supplying that drawing and amending the specification to add a brief description of it.

☐ In the reissue application, FIG. , is labeled as "New" but is not described in the reissue specification's brief description of the drawings. Applicant must respond by amending the reissue specification's brief description of the drawings to add a brief description of the new drawing.

☐ OTHER:

☐ COMMENTS:

COPY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

182115 7590 06/05/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER	
BOUTAH, ALINA A	

ART UNIT	PAPER NUMBER
2442	

NOTIFICATION DATE	DELIVERY MODE
06/05/2023	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

cofficeaction@appcoll.com
katc@klspatents.com
office@klspatents.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

Application No. : 17671199
Applicant : Erlingsson
Filing Date : 02/14/2022
Date Mailed : 06/05/2023

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Notice of Allowance Mailed

This application has been accorded an Allowance Date and is being prepared for issuance. The application, however, is incomplete for the reasons below.

Applicant is given two (2) months from the mail date of this Notice within which to respond. This time period for reply is extendable under 37 CFR 1.136(a) for only TWO additional MONTHS.

The informalities requiring correction are indicated in the attachment(s). If the informality pertains to the abstract, specification (including claims) or drawings, the informality must be corrected with an amendment in compliance with 37 CFR 1.121 (or, if the application is a reissue application, 37 CFR 1.173). Such an amendment may be filed after payment of the issue fee if limited to correction of informalities noted herein. See Waiver of 37 CFR 1.312 for Documents Required by the Office of Patent Publication, 1280 Off. Gaz. Patent Office 918 (March 23, 2004). In addition, if the informality is not corrected until after payment of the issue fee, for purposes of 35 U.S.C. 154(b)(1)(iv), "all outstanding requirements" will be considered to have been satisfied when the informality has been corrected. A failure to respond within the above-identified time period will result in the application being ABANDONED.

See attachment(s).

A copy of this notice MUST be returned with the reply. Please address response to "Mail Stop Issue Fee, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450".

/Chris Harmon/
Publication Branch
Office of Data Management
(571) 272-4200

Application No. 17671199

IDENTIFICATION OF SPECIFICATION/DRAWING INCONSISTENCIES

☐ On Page of the specification there is a brief description of FIG. , but the drawings filed do not include a drawing with that designation. Applicant must respond either by supplying the omitted drawing or by amending the specification to remove all references to that drawing.

☒ The drawings filed 02/14/2022 include FIG. 12, but the specification's brief description of the drawings does not describe a drawing with that designation. Applicant must respond either by amending the specification to add a brief description of that drawing or by correcting the drawings to remove the drawing in question.

☐ In the utility application, drawings are present in the application and are referred to in the detailed description of the invention, but the specification does not contain a brief description of the several views as required by 37 CFR 1.74 and 37 CFR 1.77(b)(9).

☐ The design application does not contain a description of the figure or figures of the drawing as required by 37 CFR 1.153(a) and 37 CFR 1.154(b)(4).

☐ In the plant application, the specification does not contain a brief description of the drawing as required by 37 CFR 1.163(c)(8).

☐ Page of the specification refers to FIG. , but no drawing with that designation is described in the brief description of the drawings and no drawing with that designation is present in the application. Applicant must respond either by amending the specification to remove all references to that drawing, or by supplying that drawing and amending the specification to add a brief description of it.

☐ In the reissue application, FIG. , is labeled as "New" but is not described in the reissue specification's brief description of the drawings. Applicant must respond by amending the reissue specification's brief description of the drawings to add a brief description of the new drawing.

☐ OTHER:

☐ COMMENTS:

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

182115 7590 05/26/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER
BOUTAH, ALINA A

ART UNIT	PAPER NUMBER
2442	

DATE MAILED: 05/26/2023

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

TITLE OF INVENTION: LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$1200	$0.00	$0.00	$1200	08/28/2023

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 40% the amount of undiscounted fees, and micro entity fees are 20% the amount of undiscounted fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00260

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

182115 7590 05/26/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via EFS-Web or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

TITLE OF INVENTION: LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$1200	$0.00	$0.00	$1200	08/28/2023

EXAMINER	ART UNIT	CLASS-SUBCLASS
BOUTAH, ALINA A	2442	709-224000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/AIA/122 or PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. Fees submitted: ❑ Issue Fee ❑ Publication Fee (if required)

4b. Method of Payment: *(Please first reapply any previously paid fee shown above)*

❑ Electronic Payment via Patent Center or EFS-Web ❑ Enclosed check ❑ Non-electronic payment by credit card (Attach form PTO-2038)

❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

5. **Change in Entity Status** (from status indicated above)

❑ Applicant certifying micro entity status. See 37 CFR 1.29

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

❑ Applicant asserting small entity status. See 37 CFR 1.27

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

❑ Applicant changing to regular undiscounted fee status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00261

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

182115 7590 05/26/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER	
BOUTAH, ALINA A	
ART UNIT	**PAPER NUMBER**
2442	

DATE MAILED: 05/26/2023

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No. 17/671,199	Applicant(s) ERLINGSSON et al.	
	Examiner ALINA N BOUTAH	Art Unit 2442	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☑ This communication is responsive to 4/6/23.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☑ The allowed claim(s) is/are 1-4,7-12 and 15-24 . As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information , please see **http://www.uspto.gov/patents/init_events/pph/index.jsp** or send an inquiry to **PPHfeedback@uspto.gov.**

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) ☐ All b) ☐ Some* c) ☐ None of the:

   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☑ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____.
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material _____.
4. ☐ Interview Summary (PTO-413), Paper No./Mail Date. _____.
5. ☑ Examiner's Amendment/Comment
6. ☐ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

/ALINA A BOUTAH/
Primary Examiner, Art Unit 2442

Notice of Pre-AIA or AIA Status

The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

Information Disclosure Statement

The IDS filed 4/21/2023 has been considered.

Allowable Subject Matter

In view of the amendment, the rejections under 35 USC 101/102/103 are withdrawn. Claims are now in condition for allowance.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ALINA N BOUTAH whose telephone number is (571)272-3908. The examiner can normally be reached M-F 7:00 AM - 3:00 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Trost can be reached on 571-272-7872. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: https://patentcenter.uspto.gov. Visit https://www.uspto.gov/patents/apply/patent-center for more information about Patent Center and https://www.uspto.gov/patents/docx for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ALINA BOUTAH
Primary Examiner
Art Unit 2442

/ALINA A BOUTAH/
Primary Examiner, Art Unit 2442

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	**Art Unit**
	ALINA N BOUTAH	2442

CPC						
Symbol					**Type**	**Version**
H04L	/	67	/	535	F	2022-05-01
G06F	/	16	/	9024	I	2019-01-01
G06F	/	21	/	57	I	2013-01-01
H04L	/	63	/	10	I	2013-01-01
H04L	/	67	/	306	I	2013-01-01
G06F	/	16	/	9537	I	2019-01-01
G06F	/	9	/	455	I	2013-01-01
G06F	/	9	/	545	I	2013-01-01
H04L	/	43	/	06	I	2013-01-01
G06F	/	16	/	9535	I	2019-01-01
G06F	/	16	/	9038	I	2019-01-01
G06F	/	16	/	2456	A	2019-01-01
H04L	/	41	/	08	A	2013-01-01

CPC Combination Sets							
Symbol					**Type**	**Set**	**Ranking**	**Version**
	/		/					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	20	
/ALINA A BOUTAH/ Primary Examiner, Art Unit 2442	05 May 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	11

U.S. Patent and Trademark Office

Part of Paper No.: 20230505

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00267

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	**Art Unit**
	ALINA N BOUTAH	2442

INTERNATIONAL CLASSIFICATION					
CLAIMED					
H04L	/	67	/	50	
G06F	/	21	/	57	

NON-CLAIMED				
	/		/	

US ORIGINAL CLASSIFICATION	
CLASS	**SUBCLASS**

CROSS REFERENCES(S)						
CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**					

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	20	
/ALINA A BOUTAH/ Primary Examiner, Art Unit 2442	05 May 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	11

U.S. Patent and Trademark Office

Part of Paper No.: 20230505

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00268

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	**Art Unit**
	ALINA N BOUTAH	2442

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIMS

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	11	12	20	23										
2	2	12	15	7	24										
3	3	13	16												
4	4	14	17												
5	7	15	18												
6	8	16	19												
8	9	17	20												
9	10	18	21												
10	11	19	22												

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	20	
/ALINA A BOUTAH/ Primary Examiner, Art Unit 2442	05 May 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	11

U.S. Patent and Trademark Office

Part of Paper No.: 20230505

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00269

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	Art Unit
	ALINA N BOUTAH	2442

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	02/06/2023	05/05/2023							
1	1	✓	=							
2	2	✓	=							
3	3	✓	=							
4	4	✓	=							
	5	✓	-							
	6	✓	-							
5	7	✓	=							
6	8	✓	=							
8	9	✓	=							
9	10	✓	=							
10	11	✓	=							
11	12	✓	=							
	13	✓	-							
	14	✓	-							
12	15	✓	=							
13	16	✓	=							
14	17	✓	=							
15	18	✓	=							
16	19	✓	=							
17	20	✓	=							
18	21		=							
19	22		=							
20	23		=							
7	24		=							

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00270

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	Art Unit
	ALINA N BOUTAH	2442

CPC - Searched*		
Symbol	Date	Examiner
H04L63/1425,1441,20,1416	05/05/2023	ANB
H04L67/535	05/05/2023	ANB

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
inventor and assignee search	01/29/2023	ANB
Global Dossier	01/29/2023	ANB
SEARCH	01/29/2023	ANB
Google	01/29/2023	ANB
STIC	02/06/2023	ANB
IEEE	02/06/2023	ANB
ACM	02/06/2023	ANB
IP.com	02/06/2023	ANB
Google Patents	02/06/2023	ANB
updated SEARCH	05/05/2023	ANB

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00271

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	Art Unit
	ALINA N BOUTAH	2442

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
H04L	63/1425,1441,20,1416	05/05/2023	ANB
H04L	67/535	05/05/2023	ANB

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00272

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	2442
Examiner Name	Alina A BOUTAH
Attorney Docket Number	L0041B1CIP2

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	10122740	B1	2018-11-06	Finkelshtein et al.	
	2	10237294	B1	2019-03-19	Zadeh et al.	
	3	10367704	B2	2019-07-30	Giura et al.	
	4	10382303	B2	2019-08-13	Khanal et al.	
	5	10419468	B2	2019-09-17	Glatfelter et al.	
	6	10447526	B2	2019-10-15	Tucker et al.	
	7	10505818	B1	2019-12-10	Yona et al.	
	8	10521584	B1	2019-12-31	Mehr	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	2442
Examiner Name	Alina A BOUTAH
Attorney Docket Number	L0041B1CIP2

	9	10565373	B1	2020-02-18	Rao et al.	
	10	10791131	B2	2020-09-29	Nor et al.	
	11	10797974	B2	2020-10-06	Giura et al.	
	12	10812497	B2	2020-10-20	Venkatramani et al.	
	13	10904270	B2	2021-01-26	Muddu et al.	
	14	10911470	B2	2021-02-02	Muddu et al.	
	15	11575693	B1	2023-02-07	Muddu et al.	
	16	6363411	B1	2002-03-26	Dugan et al.	
	17	7054873	B2	2006-05-30	Nordström et al.	
	18	8352589	B2	2013-01-08	Ridel et al.	
	19	9225730	B1	2015-12-29	Brezinski	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

	20	9497224	B2	2016-11-15	Sweet et al.	
	21	9596254	B1	2017-03-14	Muddu et al.	
	22	9729416	B1	2017-08-08	Khanal et al.	
	23	9811790	B2	2017-11-07	Ahern et al.	
	24	9923911	B2	2018-03-20	Vasseur et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20070130330	A1	2007-06-07	Ridel et al.	
	2	20070168696	A1	2007-07-19	Ridel et al.	
	3	20080270451	A1	2008-10-30	Thomsen et al.	
	4	20150058619	A1	2015-02-26	Sweet et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

	5	20170063906	A1	2017-03-02	Muddu et al.	
	6	20170063908	A1	2017-03-02	Muddu et al.	
	7	20170063910	A1	2017-03-02	Muddu et al.	
	8	20170063912	A1	2017-03-02	Muddu et al.	
	9	20170230183	A1	2017-08-10	Sweet et al.	
	10	20170277553	A1	2017-09-28	Zada et al.	
	11	20180013650	A1	2018-01-11	Khanal et al.	
	12	20180019932	A1	2018-01-18	Giura et al.	
	13	20180069885	A1	2018-03-08	Patterson et al.	
	14	20180123864	A1	2018-05-03	Tucker et al.	
	15	20190158524	A1	2019-05-23	Zadeh et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

	16	20190312796	A1	2019-10-10	Giura et al.	
	17	20190342311	A1	2019-11-07	Muddu et al.	
	18	20190364067	A1	2019-11-28	Yona et al.	
	19	20200065857	A1	2020-02-27	Lagi et al.	
	20	20200228555	A1	2020-07-16	Wittenschlaeger	
	21	20200287927	A1	2020-09-10	Zadeh et al.	
	22	20200404008	A1	2020-12-24	Venkatramani et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	2442
Examiner Name	Alina A BOUTAH
Attorney Docket Number	L0041B1CIP2

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/ALINA A BOUTAH/	Date Considered	05/05/2023

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	2442
Examiner Name	Alina A BOUTAH
Attorney Docket Number	L0041B1CIP2

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

☒ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Edward Lenart/	Date (YYYY-MM-DD)	2023-04-21
Name/Print	Edward J. Lenart	Registration Number	63,665

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.P/

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L61	46	("10122740" OR "10237294" OR "10367704" OR "10382303" OR "10419468" OR "10447526" OR "10505818" OR "10521584" OR "10565373" OR "10791131" OR "10797974" OR "10812497" OR "10904270" OR "10911470" OR "11575693" OR "20070130330" OR "20070168696" OR "20080270451" OR "20150058619" OR "20170063906" OR "20170063908" OR "20170063910" OR "20170063912" OR "20170230183" OR "20170277553" OR "20180013650" OR "20180019932" OR "20180069885" OR "20180123864" OR "20190158524" OR "20190312796" OR "20190342311" OR "20190364067" OR "20200065857" OR "20200228555" OR "20200287927" OR "20200404008" OR "6363411" OR "7054873" OR "8352589" OR "9225730" OR "9497224" OR "9596254" OR "9729416" OR "9811790" OR "9923911").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 11:55 AM
L62	1060813	(permission permit$4 admit$4 allow$4 admission approv$4 endors$7 privileg$$ warrant allowance) WITH level	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:10 PM

L63	613	L38 AND L62	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:11 PM
L64	84	L63 AND (L7 L18)	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:11 PM
L65	106404	(permission permit$4 admit$4 allow$4 admission approv$4 endors$7 privileg$$ warrant allowance) WITH level WITH (less reduc$4)	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:17 PM
L66	1379	L19 AND L65	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:18 PM
L67	0	L52 AND L66	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:18 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N1	51202	H04L63/1425,1441,20,1416.cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:18 PM
N2	24154	h04l67/535.cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:18 PM
N3	76380	(H04L41/145 OR H04L67/10 OR H04L63/1408 OR H04L41/024 OR H04L41/0846 OR H04L41/0806 OR H04L41/0895).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:19 PM
N4	1414	cloud WITH similar$7 WITH deploy$4	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:19 PM
N5	106404	(permission permit$4 admit$4 allow$4 admission approv$4 endors$7 privileg$$ warrant allowance) WITH level WITH (less reduc$4)	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:19 PM
N6	103	N5 AND N4	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:20 PM
N7	1	(N1 N2 N3) AND N6	(US-PGPUB; USPAT)	OR	ON	ON	2023/05/05 01:20 PM
N8	37	(Artificial Intelligence) More like doc: US-20220247769-A1 Text: (US-20220200869-A1 OR US-20220279004-A1 OR US-20220232025-A1 OR US-20220329616-A1 OR WO-2023038957-	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO,	OR	ON	ON	2023/05/05 01:20 PM

		A1 OR US-20220232024-A1 OR US-20230032686-A1 OR US-20220224707-A1 OR US-20220400130-A1 OR WO-2023034444-A1 OR US-20230075355-A1 OR US-20220400129-A1 OR WO-2023034419-A1 OR US-11470172-B1 OR US-20220360600-A1 OR US-11134093-B1 OR US-20220215101-A1 OR US-20220311794-A1 OR US-20220294816-A1 OR US-20220303295-A1).did.	NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				

Bibliographic Data

Application No: 17/671,199

Foreign Priority claimed: ○ Yes ◉ No

35 USC 119 (a-d) conditions met: ☐ Yes ☐ No ☐ Met After Allowance

Verified and Acknowledged: /ALINA A BOUTAH/ Examiner's Signature | Initials

Title: LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
02/14/2022	709	2442	
RULE			

APPLICANTS

LACEWORK, INC., San Jose, CA,

INVENTORS

ÚLFAR ERLINGSSON, PALO ALTO, CA, UNITED STATES

YIJOU CHEN, CUPERTINO, CA, UNITED STATES

CONTINUING DATA

This application is a CIP of 17504311 10/18/2021

This application has PRO of 63243013 09/10/2021

17504311 is a CON of 16665961 10/28/2019 PAT 11153339

16665961 is a CON of 16134794 09/18/2018 PAT 10581891

16134794 has PRO of 62650971 03/30/2018

16134794 has PRO of 62590986 11/27/2017

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

02/18/2022

STATE OR COUNTRY

UNITED STATES

ADDRESS

FILING FEE RECEIVED

$2,240

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	2442
Examiner Name	Alina A BOUTAH
Attorney Docket Number	L0041B1CIP2

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	10122740	B1	2018-11-06	Finkelshtein et al.	
	2	10237294	B1	2019-03-19	Zadeh et al.	
	3	10367704	B2	2019-07-30	Giura et al.	
	4	10382303	B2	2019-08-13	Khanal et al.	
	5	10419468	B2	2019-09-17	Glatfelter et al.	
	6	10447526	B2	2019-10-15	Tucker et al.	
	7	10505818	B1	2019-12-10	Yona et al.	
	8	10521584	B1	2019-12-31	Mehr	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

	9	10565373	B1	2020-02-18	Rao et al.	
	10	10791131	B2	2020-09-29	Nor et al.	
	11	10797974	B2	2020-10-06	Giura et al.	
	12	10812497	B2	2020-10-20	Venkatramani et al.	
	13	10904270	B2	2021-01-26	Muddu et al.	
	14	10911470	B2	2021-02-02	Muddu et al.	
	15	11575693	B1	2023-02-07	Muddu et al.	
	16	6363411	B1	2002-03-26	Dugan et al.	
	17	7054873	B2	2006-05-30	Nordström et al.	
	18	8352589	B2	2013-01-08	Ridel et al.	
	19	9225730	B1	2015-12-29	Brezinski	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

	20	9497224	B2	2016-11-15	Sweet et al.	
	21	9596254	B1	2017-03-14	Muddu et al.	
	22	9729416	B1	2017-08-08	Khanal et al.	
	23	9811790	B2	2017-11-07	Ahern et al.	
	24	9923911	B2	2018-03-20	Vasseur et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20070130330	A1	2007-06-07	Ridel et al.	
	2	20070168696	A1	2007-07-19	Ridel et al.	
	3	20080270451	A1	2008-10-30	Thomsen et al.	
	4	20150058619	A1	2015-02-26	Sweet et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

	5	20170063906	A1	2017-03-02	Muddu et al.	
	6	20170063908	A1	2017-03-02	Muddu et al.	
	7	20170063910	A1	2017-03-02	Muddu et al.	
	8	20170063912	A1	2017-03-02	Muddu et al.	
	9	20170230183	A1	2017-08-10	Sweet et al.	
	10	20170277553	A1	2017-09-28	Zada et al.	
	11	20180013650	A1	2018-01-11	Khanal et al.	
	12	20180019932	A1	2018-01-18	Giura et al.	
	13	20180069885	A1	2018-03-08	Patterson et al.	
	14	20180123864	A1	2018-05-03	Tucker et al.	
	15	20190158524	A1	2019-05-23	Zadeh et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

	16	20190312796	A1	2019-10-10	Giura et al.	
	17	20190342311	A1	2019-11-07	Muddu et al.	
	18	20190364067	A1	2019-11-28	Yona et al.	
	19	20200065857	A1	2020-02-27	Lagi et al.	
	20	20200228555	A1	2020-07-16	Wittenschlaeger	
	21	20200287927	A1	2020-09-10	Zadeh et al.	
	22	20200404008	A1	2020-12-24	Venkatramani et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	2442
	Examiner Name	Alina A BOUTAH
	Attorney Docket Number	L0041B1CIP2

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

☒ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Edward Lenart/	Date (YYYY-MM-DD)	2023-04-21
Name/Print	Edward J. Lenart	Registration Number	63,665

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/671,199	**04/21/2023 01:45:02 PM ET**	**L0041B1CIP2**

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1070	FILED BY	Kate Hill
PATENT CENTER #	61963259	FILING DATE	02/14/2022
CUSTOMER #	182115	FIRST NAMED INVENTOR	ÚLFAR ERLINGSSON
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Edward Lenart

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
2023-04-21_L0041B1CIP2_IDS.pdf	8	Information Disclosure Statement (IDS) Form (SB08)	1253 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
2023-04-21_L0041B1CIP2_IDS.pdf	CA97031AEF92DC21922821F00B1862756B0E3DBE4979C5AE3AC3F6453AF3D49D0303538FCC2F1EB3161EF5766C0AAB6A04FC9BF6E99D7776424629E65FC01A1D

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized

by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/671,199	04/21/2023 01:45:02 PM ET	L0041B1CIP2

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1070	FILED BY	Kate Hill
PATENT CENTER #	61963259	AUTHORIZED BY	Edward Lenart
CUSTOMER #	182115	FILING DATE	02/14/2022
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	ÚLFAR ERLINGSSON

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1003	E20234KD46088123	Kate Hill

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1806	SUBMISSION OF AN INFORMATION DISCLOSURE STATEMENT	260.00	1	260.00
			TOTAL AMOUNT:	$260.00

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	ÚLFAR ERLINGSSON
Serial No.:	17/671,199
Atty Docket No.:	L0041B1CIP2
Group Art Unit:	2442
Filed:	02/14/2022
Customer No.:	182115
Confirmation No.:	1070
Examiner:	ALINA A BOUTAH
Title:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Mail Stop: Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

RESPONSE TO OFFICE ACTION DATED FEBRUARY 10, 2023

This is a Response to the Office Action dated February 10, 2023 (hereafter "the Office Action"). Applicants present the following amendments and remarks demonstrating that the case is in condition for allowance.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00297

AMENDMENT

Amendments To The Claims:

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of learning from similar cloud deployments, the method comprising:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning;

receiving information describing configurations associated with the additional cloud deployments, including receiving information describing permissions for one or more users of the additional cloud deployments;

determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment, including identifying a reduced privilege level to give to the one or more users.

2. (Original) The method of claim 1 further comprising:

receiving information describing one or more actions associated with the additional cloud deployments; and

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00298

wherein identifying the one or more configurations to adopt for the first cloud deployment is further based on the one or more actions.

3. (Original) The method of claim 2 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat.

4. (Original) The method of claim 2 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a detected vulnerability associated with one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to address the vulnerability.

5. (Canceled)

6. (Canceled)

7. (Original) The method of claim 1 further comprising identifying abnormally configured components in the first cloud deployment.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00299

8. (Original) The method of claim 1 further comprising receiving information describing one or more deployment processes associated with the additional cloud deployments.

9. (Currently Amended) A system for learning from similar cloud deployments, the system ~~including~~ comprising at least one processor and memory storing computer program instructions that, when executed, cause the system to carry out the steps of:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning;

receiving information describing configurations associated with the additional cloud deployments, including receiving information describing permissions for one or more users of the additional cloud deployments;

determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment, including identifying a reduced privilege level to give to the one or more users.

10. (Currently Amended) The system of claim 9 ~~further comprising~~ wherein the computer program instructions[[that]], when executed, cause the system to carry out the step of:

receiving information describing one or more actions associated with the additional cloud deployments; and

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00300

wherein identifying the one or more configurations to adopt for the first cloud deployment is further based on the one or more actions.

11. (Original) The system of claim 10 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat.

12. (Original) The system of claim 10 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a detected vulnerability associated with one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to address the vulnerability.

13. (Canceled)

14. (Canceled)

15. (Currently Amended) The system of claim 9 ~~further comprising~~ wherein the computer program instructions[[that]], when executed, cause the system to carry

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00301

out the step of identifying abnormally configured components in the first cloud deployment.

16. (Currently Amended) The system of claim 9 ~~further comprising~~ wherein the computer program instructions[[that]], when executed, cause the system to carry out the step of receiving information describing one or more deployment processes associated with the additional cloud deployments.

17. (Currently Amended) A computer program product for learning from similar cloud deployments, the computer program product disposed on a non-transitory computer readable medium, the computer program product including computer program instructions that, when executed, carry out the steps of:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning;

receiving information describing configurations associated with the additional cloud deployments, including receiving information describing permissions for one or more users of the additional cloud deployments;

determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment, including identifying a reduced privilege level to give to the one or more users.

18. (Original) The computer program product of claim 17 further comprising computer program instructions that, when executed, carry out the step of:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00302

receiving information describing one or more actions associated with the additional cloud deployments; and

wherein identifying the one or more configurations to adopt for the first cloud deployment is further based on the one or more actions.

19. (Original) The computer program product of claim 18 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat.

20. (Original) The computer program product of claim 18 further comprising computer program instructions that, when executed, carry out the step of creating a polygraph associated with the first cloud deployment.

21. (New) The computer program product of claim 18 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a detected vulnerability associated with one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to address the vulnerability.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00303

22. (New) The computer program product of claim 18 further comprising computer program instructions that, when executed, carry out the step of identifying abnormally configured components in the first cloud deployment.

23. (New) The computer program product of claim 18 further comprising computer program instructions that, when executed, carry out the step of receiving information describing one or more deployment processes associated with the additional cloud deployments.

24. (New) The method of claim 1 wherein determining that one or more users of the first cloud deployment are over-permissioned comprises:

identifying the one or more users of the additional cloud deployments based on a similarity to the one or more users of the first cloud deployment; and

determining that the one or more users of the additional cloud deployments have permissions different or lesser than the one or more users of the first cloud deployment.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00304

REMARKS

Claims 1-4, 7-12, and 15-24 are pending. In this Amendment, Applicants have amended claims 1, 9, 10, and 15-17 and canceled claims 5, 6, 13, and 14 from further consideration in this application to facilitate expeditious prosecution of the application. Applicants are not conceding that the subject matter encompassed by the claims prior to this Amendment is unpatentable over the art cited by the Examiner. Applicants respectfully reserve the right to pursue claims in one or more continuing applications, including claims capturing the subject matter encompassed by the claims prior to this Amendment and additional claims. Claims 21-24 are newly added. No new matter has been added.

Examiner Interview

Applicant thanks the Examiner for the time spent with Applicant's representative, Jason Fletcher (Reg. No. 68,044), during the telephonic interview held March 22, 2023. An agreement was reached regarding amendments that would distinguish the claims from the cited art. This agreement is reflected in the amendments and arguments presented herein.

Claim Rejections – 35 U.S.C. § 101

Claims 9-20 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Specifically, claims 9 and 17, directed to a system and a computer program product, respectively, allegedly lack any evidence of hardware. Claim 9 has been amended to recite "at least one processor and memory" and claim 17 has been amended to recite a "non-transitory" computer-readable medium. Accordingly, Applicant asserts that the claims as amended are directed to statutory subject matter and request that their rejection be withdrawn.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00305

Claim Rejections – 35 U.S.C. § 102 and § 103

Claims 1-2, 9-10, and 17-18 stand rejected under 35 U.S.C. § 102 as being anticipated by Jeuk (U.S. Patent Publication No. 2019/0123973). Claims 3-6, 8, 11-14, 16, and 19-20 stand rejected under 35 U.S.C. § 103 as being unpatentable over Jeuk in view of Kirti (U.S. Patent Publication No. 2017/0251013). Claims 7 and 15 stand rejected under 35 U.S.C. § 103 as being unpatentable over Jeuk in view of Jayanthi (U.S. Patent Publication No. 2021/0188248).

Applicants respectfully note in response that no single cited reference can anticipate independent claims 1, 9, and 17 of the present application, under 35 U.S.C. § 102 and no combination of only the cited references can be used to establish a prima facie case of obviousness against independent claims 1, 9, and 17 of the present application under 35 U.S.C. § 103 because the cited references, individually or in combination, do not disclose, teach, or suggest every element and limitation of independent claims 1, 9, and 17 of the present application.

For example, the cited references, individually or in combination, do not disclose, teach, or suggest at least the following limitations of independent claims 1, 9, and 17 of the present application:

> receiving information describing configurations associated with the additional cloud deployments, including receiving information describing permissions for one or more users of the additional cloud deployments;
>
> determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and
>
> identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment, including identifying a reduced privilege level to give to the one or more users.

Applicant notes that these elements incorporate subject matter similar to canceled claims 5, 6, 13, and 14. The Office Action admits that Jeuk fails to disclose these canceled

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00306

claims, citing Kirti at [0078, 0172] as curing the deficiencies of Jeuk. Applicant disagrees. The cited portions of Kirti describe that data for an application may be accessed by a third-party application if given permission by a user of the application. Kirti at [0172]. Applicant notes that, to the extent that Kirti discloses permissions, Kirti limits its discussion to the permissions of a third-party application to access data from another application when granted by a user. In other words, Kirti does not disclose the permissions of a user, but rather permissions of a third-party application granted by a user. One skilled in the art would understand that user permissions relate to actions permissible by the user, not a third-party application as described in Kirti. Moreover, as the cited portions of Kirti does not disclose user permissions at all, Kirti does not disclose any determination as to whether users of a cloud deployment are overpermissioned, let alone based on information describing permissions of users of other cloud deployments as claimed.

Kirti at [0078] discloses that a control manager may control access permitted to applications by a device in a computing environment based on various policies. Policies may be configured by a user with respect to one or more users or a tenancy. Even if, for the sake of argument, these policies were to disclose user permissions, Kirti does not disclose any determination as to whether a user of a cloud deployment are overpermissioned as claimed. Moreover, Kirti fails to disclose identifying a reduced privilege level for users at all, let alone based on a determination that those users are overpermissioned using information describing the permissions of users of other cloud deployments. Therefore, Kirti fails to disclose these elements.

Because the cited references, individually or in combination, do not disclose, teach, or suggest every element and limitation of independent claims 1, 9, and 17 of the present application, no single cited reference can anticipate independent claims 1, 9, and 17 of the present application, under 35 U.S.C. § 102 and no combination of only the cited references can be used to establish a prima facie case of obviousness against independent claims 1, 9, and 17 of the present application under 35 U.S.C. § 103. Therefore, independent claims 1, 9, and 17 and their dependent claims are allowable. Applicants

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00307

therefore respectfully request that the rejections under 35 U.S.C. § 102 and § 103 be withdrawn.

The Commissioner is hereby authorized to charge or credit Deposit Account No. 50-3082 for any fees required or overpaid.

Respectfully submitted,

Date: April 6, 2023

By: /Jason Fletcher/

Jason Fletcher, Reg. No. 68,044
Kennedy Lenart Spraggins LLP
Tel. (512) 472-9881
Fax (512) 472-9887
ATTORNEY FOR APPLICANT

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00308

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/671,199	**04/06/2023 03:30:02 PM ET**	**L0041B1CIP2**

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1070	FILED BY	Ashley Jenkins
PATENT CENTER #	61880362	FILING DATE	02/14/2022
CUSTOMER #	182115	FIRST NAMED INVENTOR	ÚLFAR ERLINGSSON
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Jason Fletcher

Documents

TOTAL DOCUMENTS: 3

DOCUMENT		PAGES	DESCRIPTION	SIZE (KB)
2023-04-06_L0041B1CIP2_ResponsetoNFRejection.pdf		12	-	113 KB
2023-04-06_L0041B1CIP2_ResponsetoNFRejection-A....pdf	(1-1)	1	Amendment/Request for Reconsideration-After Non-Final Rejection	77 KB
2023-04-06_L0041B1CIP2_ResponsetoNFRejection-CLM.pdf	(2-8)	7	Claims	100 KB
2023-04-06_L0041B1CIP2_ResponsetoNFRejection-REM.pdf	(9-12)	4	Applicant Arguments/Remarks Made in an Amendment	83 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
2023-04-06_L0041B1CIP2_ResponsetoNFRejection.pdf	2D0433B6A8D66FC004028FEBE88CC0A52C5501E552A9823F9D5B15D472ED2401B811B1E17D936814CC03F816707E2A30BC105D39B0182B7E5B85D9342F9DFF4E
2023-04-06_L0041B1CIP2_ResponsetoNFRejection-A....pdf	A05CD5E91F69444EA7BF783C4822DC40C041AEED23807244DD5A407F466B9C2CC24754BCEE2D13AFDD73DB8654AA4E7354EE2C86025FB7167202AECB0384B1AD
2023-04-06_L0041B1CIP2_ResponsetoNFRejection-CLM.pdf	1D0EF74C873403936D64735190A2FBB75C01D60317064BBE9F2845EFF3F61D20C1F1EDC2B8BD3CC567C85D32FF037574E1C5405DCCF7084D4FFDF735B5A7525F
2023-04-06_L0041B1CIP2_ResponsetoNFRejection-REM.pdf	12098E4D1DA8657FA94C213A8720AAEDF1AE3141A060AA1D5378927E0874EB8C120F9F1660405217AF9DCCC613DC9C3CC1BE9C7A6F53FCEA0D8F12E06D00A6F0

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

182115 7590 03/28/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER	
BOUTAH, ALINA A	

ART UNIT	PAPER NUMBER
2442	

NOTIFICATION DATE	DELIVERY MODE
03/28/2023	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@appcoll.com
kate@klspatents.com
office@klspatents.com

Applicant-Initiated Interview Summary	**Application No.** 17/671,199	**Applicant(s)** ERLINGSSON et al.		
	Examiner ALINA N BOUTAH	**Art Unit** 2442	**AIA (First Inventor to File) Status** Yes	**Page** 1 of 1

All Participants (applicant, applicants representative, PTO personnel)	**Title**	**Type**
ALINA N BOUTAH	Primary Examiner	Telephonic
JASON FLETCHER	Attorney	

Date of Interview: 22 March 2023

Issues Discussed:

Proposed Amendment(s)

Proposed amendment appears to overcome the cited references. However, allowability is subject to further search.

☑ Attachment

/ALINA A BOUTAH/ Primary Examiner, Art Unit 2442	

Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04
Please further see:
MPEP 713.04
Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b)
37 CFR § 1.2 Business to be transacted in writing

Applicant recordation instructions: The formal written reply to the last Office action must include the substance of the interview. (See MPEP section 713.04). If a reply to the last Office action has already been filed, applicant is given a non-extendable period of the longer of one month or thirty days from this interview date, or the mailing date of this interview summary form, whichever is later, to file a statement of the substance of the interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	ÚLFAR ERLINGSSON
Serial No.:	17/671,199
Atty Docket No.:	L0041B1CIP2
Group Art Unit:	2442
Filed:	02/14/2022
Customer No.:	182115
Confirmation No.:	1070
Examiner:	ALINA A BOUTAH
Title:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

INTERVIEW AGENDA

Applicant wishes to discuss the 102 and 103 rejections in light of the proposed amendments.

Claim Rejections – 35 U.S.C. § 102 and § 103

Claims 1-2, 9-10, and 17-18 stand rejected under 35 U.S.C. § 102 as being anticipated by Jeuk (U.S. Patent Publication No. 2019/0123973). Claims 3-6, 8, 11-14, 16, and 19-20 stand rejected under 35 U.S.C. § 103 as being unpatentable over Jeuk in view of Kirti (U.S. Patent Publication No. 2017/0251013). Claims 7 and 15 stand rejected under 35 U.S.C. § 103 as being unpatentable over Jeuk in view of Jayanthi (U.S. Patent Publication No. 2021/0188248).

Applicants respectfully note in response that no single cited reference can anticipate independent claims 1, 9, and 17 of the present application, under 35 U.S.C. § 102 and no combination of only the cited references can be used to establish a prima facie case of

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00313

obviousness against independent claims 1, 9, and 17 of the present application under 35 U.S.C. § 103 because the cited references, individually or in combination, do not disclose, teach, or suggest every element and limitation of independent claims 1, 9, and 17 of the present application.

For example, the cited references, individually or in combination, do not disclose, teach, or suggest at least the following limitations of independent claims 1, 9, and 17 of the present application:

> receiving information describing configurations associated with the additional cloud deployments, including receiving information describing permissions for one or more users of the additional cloud deployments;
>
> determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and
>
> identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment, including identifying a reduced privilege level to give to the one or more users.

Applicant notes that these elements incorporate subject matter similar to canceled claims 5, 6, 13, and 14. The Office Action admits that Jeuk fails to disclose these canceled claims, citing Kirti at [0078, 0172] as curing the deficiencies of Jeuk. Applicant disagrees. The cited portions of Kirti describe that data for an application may be accessed by a third-party application if given permission by a user of the application. Jeuk at [0172]. Applicant notes that, to the extent that Jeuk discloses permissions, Jeuk limits its discussion to the permissions of a third-party application to access data from another application when granted by a user. In other words, Jeuk does not disclose the permissions of a user, but rather permissions of a third-party application granted by a user. One skilled in the art would understand that user permissions relate to actions permissible by the user, not a third-party application as described in Jeuk. Moreover, as the cited portions of Jeuk does not disclose user permissions at all, Jeuk does not disclose any determination as to whether users of a cloud deployment are overpermissioned, let

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00314

alone based on information describing permissions of users of other cloud deployments as claimed.

Jeuk at [0078] discloses that a control manager may control access permitted to applications by a device in a computing environment based on various policies. Policies may be configured by a user with respect to one or more users or a tenancy. Even if, for the sake of argument, these policies were to disclose user permissions, Jeuk does not disclose any determination as to whether a user of a cloud deployment are overpermissioned as claimed. Moreover, Jeuk fails to disclose identifying a reduced privilege level for users at all, let alone based on a determination that those users are overpermissioned using information describing the permissions of users of other cloud deployments. Therefore, Jeuk fails to disclose these elements.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00315

AMENDMENT

Amendments To The Claims:

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of learning from similar cloud deployments, the method comprising:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning;

receiving information describing configurations associated with the additional cloud deployments, including receiving information describing permissions for one or more users of the additional cloud deployments;

determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment, including identifying a reduced privilege level to give to the one or more users.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00316

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	L0041B1CIP2	1070

182115 7590 02/10/2023

Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

EXAMINER	
BOUTAH, ALINA A	

ART UNIT	PAPER NUMBER
2442	

NOTIFICATION DATE	DELIVERY MODE
02/10/2023	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

eofficeaction@appcoll.com
kate@klspatents.com
office@klspatents.com

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00317

Office Action Summary	Application No. 17/671,199	Applicant(s) ERLINGSSON et al.	
	Examiner ALINA N BOUTAH	Art Unit 2442	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☑ Responsive to communication(s) filed on 2/14/22.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL.** 2b) ☑ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☑ Claim(s) 1-20 is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6) ☐ Claim(s) _____ is/are allowed.

7) ☑ Claim(s) 1-20 is/are rejected.

8) ☐ Claim(s) _____ is/are objected to.

9) ☐ Claim(s) _____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov.**

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☑ The drawing(s) filed on 4/22/22 is/are: a)☑ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☑ Notice of References Cited (PTO-892)

2) ☑ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date _____.

4) ☐ Other: _____.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00318

Notice of Pre-AIA or AIA Status

The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

Information Disclosure Statement

The IDS filed 10/3/2022 has been considered.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 9-20 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claim(s) does/do not fall within at least one of the four categories of patent eligible subject matter because claims 9-16 recite a system and claims 17-20 recite computer program product. Neither the system nor the computer program product contains any evidence of hardware. Instead, one can reasonably interpret the system to be software, per se, and the computer program product to be signal. These are both non-statutory.

Claim Rejections - 35 USC § 102

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of

rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a)(1) the claimed invention was patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention.

Claim(s) 1-2, 9-10 and 17-18 is/are rejected under 35 U.S.C. 102(a)(1) as being anticipated by Jeuk et al. (US 1029/0123973, hereafter referred to as "Jeuk").

Regarding claim 1, Jeuk teaches a method of learning from similar cloud deployments, the method comprising:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning (abstract - The workload engine may utilize the received performance information to detect performance patterns of an application across the different tenant deployments.);

receiving information describing configurations associated with the additional cloud deployments ([0036] the workload analysis engine 208 may correlate an identified performance pattern with a particular configuration of cloud resources. This information may also be correlated to one or more performance measurements for the workload or application. For example, once a workload is identified as utilizing a particular resource or device, the metrics and/or logs from that particular device may be obtained and correlated to the identified workload. In still other examples, the workload analysis

engine 208 may correlate generic performance measurements with identified workloads, such as time for completion, overall CPU consumption, memory storage, and the like); and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment ([0018] The recommendation engine may analyze the performance characteristics across the multiple tenant applications and determine a "best fit" or optimized deployment of the application. The optimized deployment may or may not include particular policies of the application deployment, such as service level agreements, limitations on resource consumption, cost, etc., in general, the recommendation engine generates an optimized deployment for any iteration or type of application deployed on the cloud environment with any number of policies associated with the optimized deployment.).

Regarding claim 2, Jeuk teaches the method of claim 1 further comprising: receiving information describing one or more actions associated with the additional cloud deployments ([0028] the workload analysis engine 208 of the cloud environment 202 or cloud operator may analyze the performance information received from the tenants 204, 206 of the cloud environment to identify performance patterns of the executing applications and/or correlate performance patterns across tenant configurations of the cloud.); and wherein identifying the one or more configurations to adopt for the first cloud deployment is further based on the one or more actions ([0018] The recommendation engine may analyze the performance characteristics across the multiple tenant applications and determine a "best fit" or optimized deployment of the

application. The optimized deployment may or may not include particular policies of the application deployment, such as service level agreements, limitations on resource consumption, cost, etc., in general, the recommendation engine generates an optimized deployment for any iteration or type of application deployed on the cloud environment with any number of policies associated with the optimized deployment.).

Claims 9-10 and 17-18 are system and computer readable production versions of claim 1, respectively, therefore are rejected under the same rationale.

Claim Rejections - 35 USC § 103

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

The following is a quotation of 35 U.S.C. 103 which forms the basis for all obviousness rejections set forth in this Office action:

> A patent for a claimed invention may not be obtained, notwithstanding that the claimed invention is not identically disclosed as set forth in section 102, if the differences between the claimed invention and the prior art are such that the claimed invention as a whole would have been obvious before the effective filing date of the claimed invention to a person having ordinary skill in the art to which the claimed invention pertains. Patentability shall not be negated by the manner in which the invention was made.

Claim(s) 3-6, 8, 11-14, 16, and 19-20 is/are rejected under 35 U.S.C. 103 as being unpatentable over Jeuk in view of Kirti et al. (US 2017/0251013, hereinafter referred to as “Kirti”).

Regarding claim 3, Jeuk does not teach the method of claim 2 wherein: receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments; and receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat.

Kirti teaches receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments ([0010] The security monitoring and control system can perform analysis and correlation, including use of one or more data sources, to determine information about an application. Information may include organizational information about a provider of the application. Information may include security information about a security risk indicator about the application. The information may include one or more indicators (e.g., a value) of a feature related to use of the application, such as a security aspect. The information about an application may be used to compute a measure of security for an application (“an application risk score”) and a user (“a user risk score”). The measure of security may be computed using one or more pieces of information (e.g., indicators) combined with a weight value attribute to each piece of information. The score may be analyzed with respect to a

security policy to determine a threat of security posed by the application or user based on use of an application.); and receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat ([0011] In some embodiments, a graphical interface may be provided to a user (e.g., a security administrator) to view information about usage of applications. The information may provide details about a service provider of an application and/or a visualization of different types of security risk and may indicate a measure of severity for each security risk. The graphical interface may be interactive to configure remediation actions to be performed based on a security policy with respect to each security risk).

Before the effective filing date of the invention, one of ordinary skill in the art would have been motivated to receive information describing security threat and describe configuration settings used to combat the security threat in order to aid the administrator in monitoring and identifying risks and taking appropriate actions ([0011]).

Regarding claim 4, Jeuk does not teach the method of claim 2 wherein: receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a detected vulnerability associated with one or more of the additional cloud deployments; and receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to address the vulnerability.

Kiriti teaches receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a detected vulnerability associated with one or more of the additional cloud deployments ([0205] A measure of security of a user (also referred to herein as a “user risk score”) may provide an indication as the risk or threats to security a user may pose to an organization. As discussed above, a user may pose a security threat to an organization's network based on use of an application in an unauthorized or unsecure manner. Such use may expose an organization to vulnerabilities of its private network and data. In some instances, an application may pose a threat to an organization based on inefficient or improper use of an organization's resources, such as a corporate network and/or computing resources. A user risk score may provide a measure that indicates the severity of security threats related to a user in the organization. A user risk score may be generated continuously by profiling the user actions for applications.); and receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to address the vulnerability [[0206] The graphical interfaces enable users (e.g., a security administrator) to see associated risk indicators to understand why risk scores for some users are high, which includes unusual actions performed in the apps, risky unapproved apps accessed etc. This helps a user to take or configure remediation actions like blocking apps in the firewall, educate the user to avoid the app, or suspend the user account. The graphical interfaces may enable a user to configure customized alerts based on creating policies that matches certain conditions such as risk app score, app category, user risk score etc.].

Before the effective filing date of the invention, one of ordinary skill in the art would have been motivated to receive information describing vulnerability and describe configuration settings used to address the vulnerability in order to aid the administrator in monitoring and identifying risks and taking appropriate actions ([0011]).

Regarding claim 5, Jeuk does not teach the method of claim 1 wherein receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing permissions for one or more users of the additional cloud deployments.

Kirti teaches receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing permissions for one or more users of the additional cloud deployments ([0172] Accessing an application may include permitting data for an application to be accessed by a different application. For example, data for an application may be accessed by a third-party application to which consent was given by a user to access the user's data from a service provider of the application. The applications may be authorized (e.g., sanctioned) or unauthorized (e.g., unsanctioned) by an organization for which a user accessed the applications. The information may include data about an application that is used. The data may be an application name or a link to an application being accessed.).

Before the effective filing date of the invention, one of ordinary skill in the art would have been motivated to receive information describing permissions for one or more users of the additional cloud deployments in order to allow administrators to identify which user has what kind of permission, thus enabling security management of the cloud.

Regarding claim 6, Jeuk does not teach the method of claim 4 further comprising: determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and wherein identifying one or more configurations to adopt for the first cloud deployment further comprises identifying a reduced privilege level to give to the one or more users.

Kirti teaches determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned ([0172] Accessing an application may include permitting data for an application to be accessed by a different application. For example, data for an application may be accessed by a third-party application to which consent was given by a user to access the user's data from a service provider of the application. The applications may be authorized (e.g., sanctioned) or unauthorized (e.g., unsanctioned) by an organization for which a user accessed the applications. The information may include data about an application that is used. The data may be an application name or a link to an application being accessed.); and wherein identifying one or more configurations to adopt for the first cloud deployment further comprises identifying a reduced privilege level to give to the one or more users ([0078] Control manager 272 in security monitoring and control system 102 may handle management and control of access to applications in a computing environment. Security monitoring and control system may use one or more policies (e.g., security policies) to control access permitted to applications by a device with respect to a computing environment of

an organization. A policy may be configured by a user with respect to one or more users, or collectively a tenancy. A policy may indicate remediation actions to be performed based on security analysis of application usage by users. Remediation may include sending a notification, displaying information (e.g., a report), and/or limiting or preventing access to an application.).

Before the effective filing date of the invention, one of ordinary skill in the art would have been motivated to determine users who are over-permissioned and reducing privilege level in order to ensure that all users have appropriate level of privilege, thus reducing the chance of users accessing data not intended for them, thus making the cloud more secured.

Regarding claim 8, Jeuk teaches the method of claim 1 further comprising receiving information describing one or more deployment processes associated with the additional cloud deployments (abstract - applications deployed and executed on the cloud environment may provide performance logs and/or metrics to an inter-tenant workload engine of the cloud environment.).

Claims 11-14 and 16 are system version of claims 3-6 and 8, respectively, therefore are rejected under the same rationale.

Claim 19 is computer program product version of claim 3, therefore is rejected under the same rationale.

Regarding claim 20, Jeuk does not teach the computer program product of claim 18 further comprising computer program instructions that, when executed, carry out the step of creating a polygraph associated with the first cloud deployment. Kirti teaches creating a polygraph associated with the first cloud deployment ([011] a graphical interface may be provided to a user (e.g., a security administrator) to view information about usage of applications. The information may provide details about a service provider of an application and/or a visualization of different types of security risk and may indicate a measure of severity for each security risk. The graphical interface may be interactive to configure remediation actions to be performed based on a security policy with respect to each security risk. For organizations having difficulty in identifying and managing security risks by unknown usage of applications, the graphical interface can enable an organization to efficiently and reliably discover all if not most application usage to minimize security risks and maximize resource consumption of computing-related resources in a computing environment.). Before the effective filing date of the invention, one of ordinary skill in the art would have been motivated to create a polygraph associated with the cloud deployment in order to allow administrator to visualize activities in the cloud, thus enabling him or her to take necessary actions in managing the cloud.

Claim(s) 7 and 15 is/are rejected under 35 U.S.C. 103 as being unpatentable over Jeuk in view of Jayanthi (US 2021/0182248).

Regarding claim 7, Jeuk does not teach the method of claim 1 further comprising identifying abnormally configured components in the first cloud deployment. Jayanthi teaches identifying abnormally configured components in the first cloud deployment [0214] Embodiments of the present invention remove the need for human intervention for managing the deployment of a database in a cloud computing environment, improves the capability for identifying which database component causes one or more problems, and improves traceability and serviceability of database deployments. Embodiments further improve overall usage of resources in the cloud computing environment and reduce the cost of deployment failures. Embodiments minimize chances of misconfiguring of database architecture and software components, thus improving Quality of Service (QoS) of workload deployment. Embodiments provide a generic solution applicable to deployment of heterogeneous resources by extending the model to technologies from various database and cloud computing vendors. Embodiments minimize security threats due to misconfigured database resources and software in cloud environments since thorough syntax and semantics checks and validity scans are performed during the database deployment.). Before the effective filing date of the invention, one of ordinary skill in the art would have been motivated to identify abnormally configured components in order to remedy the configuration before deployment, thus minimizing security threats due to misconfigured resources ([0214]).

Claim 15 is system version of claim 7, therefore is rejected under the same rationale.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

1. Hu et al., WO 2021052379 – adopting cloud deployment based on behavior recognition.
2. Doron et al., US 10951648 – deploying cloud based on normalized behavior cloud service.
3. Wolff et al., US 20200259852 A1 – detecting threats in cloud service.
4. D M et al., US 20200218579 – selecting cloud deployment based on highest ranking cloud provider.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ALINA N BOUTAH whose telephone number is (571)272-3908. The examiner can normally be reached M-F 7:00 AM - 3:00 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Trost can be reached on 571-272-7872. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is

available to registered users. To file and manage patent submissions in Patent Center, visit: https://patentcenter.uspto.gov. Visit https://www.uspto.gov/patents/apply/patent-center for more information about Patent Center and https://www.uspto.gov/patents/docx for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ALINA BOUTAH
Primary Examiner
Art Unit 2442

/ALINA A BOUTAH/
Primary Examiner, Art Unit 2442

Notice of References Cited	Application/Control No. 17/671,199	Applicant(s)/Patent Under Reexamination ERLINGSSON et al.	
	Examiner ALINA N BOUTAH	Art Unit 2442	Page 1 of 2

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-11080392-B2	08-2021	Bennett; Juan Carlos	G06F21/566	1/1
*	B	US-10951648-B2	03-2021	Doron; Ehud	H04L63/1458	1/1
*	C	US-20200259852-A1	08-2020	Wolff; Matt	H04L63/1425	1/1
*	D	US-20190068627-A1	02-2019	Thampy; Sajjit	G06N20/20	1/1
*	E	US-20170251013-A1	08-2017	Kirti; Ganesh	H04L63/1441	1/1
*	F	US-20150319185-A1	11-2015	Kirti; Ganesh	H04L63/1416	726/23
*	G	US-20150172321-A1	06-2015	Kirti; Ganesh	H04L63/1425	726/1
*	H	US-20220417273-A1	12-2022	Levin; Roy	H04L63/1425	1/1
*	I	US-20220092668-A1	03-2022	Lu; Cindy Han	G06N5/04	1/1
*	J	US-20200218579-A1	07-2020	D M; Vikas	G06F16/24578	1/1
*	K	US-20180124189-A1	05-2018	Edgington; Jeff	H04L41/5096	1/1
*	L	US-20150302440-A1	10-2015	Monden; Jason Peter	G06Q30/0206	705/7.35
*	M	US-20190123973-A1	04-2019	Jeuk; Sebastian	G06F9/5027	1/1

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N	CN-110999250-B	11-2021	CN	BISWAS K	H04L41/28
	O	CN-114598840-A	06-2022	CN	HUA Y	H04L 67/10
	P	WO-2021052379-A1	03-2021	WO	FUSHINYU	H04L47/20
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Notice of References Cited	Application/Control No. 17/671,199	Applicant(s)/Patent Under Reexamination ERLINGSSON et al.	
	Examiner ALINA N BOUTAH	Art Unit 2442	Page 2 of 2

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20140229607-A1	08-2014	Jung; Gueyoung	H04L41/5051	709/224
*	B	US-20210182248-A1	06-2021	JAYANTHI; Swaroop	G06F9/45558	1/1
*	C	US-20160080204-A1	03-2016	MISHRA; Rajnish	H04L41/0853	709/220
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	**Art Unit**
	ALINA N BOUTAH	2442

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	02/06/2023								
	1	✓								
	2	✓								
	3	✓								
	4	✓								
	5	✓								
	6	✓								
	7	✓								
	8	✓								
	9	✓								
	10	✓								
	11	✓								
	12	✓								
	13	✓								
	14	✓								
	15	✓								
	16	✓								
	17	✓								
	18	✓								
	19	✓								
	20	✓								

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00335

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	Art Unit
	ALINA N BOUTAH	2442

CPC - Searched*		
Symbol	Date	Examiner
H04L63/1425,1441,20,1416	01/29/2023	ANB
H04L67/535	01/29/2023	ANB

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
inventor and assignee search	01/29/2023	ANB
Global Dossier	01/29/2023	ANB
SEARCH	01/29/2023	ANB
Google	01/29/2023	ANB
STIC	02/06/2023	ANB
IEEE	02/06/2023	ANB
ACM	02/06/2023	ANB
IP.com	02/06/2023	ANB
Google Patents	02/06/2023	ANB

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00336

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/671,199	ERLINGSSON et al.
	Examiner	Art Unit
	ALINA N BOUTAH	2442

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00337

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	OPAP
Examiner Name	DOCKET CENTRAL
Attorney Docket Number	L0041B1CIP2

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	10033611	B1	2018-07-24	Linkous et al.	
	2	10115111	B2	2018-10-30	Miltonberger	
	3	10127273	B2	2018-11-13	Dickey	
	4	10142357	B1	2018-11-27	Tamersoy et al.	
	5	10205736	B2	2019-02-12	Rieke et al.	
	6	10339309	B1	2019-07-02	Kling et al.	
	7	10389742	B2	2019-08-20	Reddy et al.	
	8	10581891	B1	2020-03-03	Kapoor et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	9	10594718	B1	2020-03-17	Deaguero et al.	
	10	10708082	B1	2020-07-07	Bakiaraj et al.	
	11	10735329	B2	2020-08-04	WANG et al.	
	12	11153339	B1	2021-10-19	Kapoor et al.	
	13	5584024	A	1996-12-10	Shwartz	
	14	5806062	A	1998-09-08	Chen et al.	
	15	8103906	B1	2012-01-24	Alibakhsh et al.	
	16	8122122	B1	2012-02-21	Clingenpeel et al.	
	17	8443442	B2	2013-05-14	Wang et al.	
	18	8671453	B2	2014-03-11	Underwood et al.	
	19	8862524	B2	2014-10-14	Zheng et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	OPAP
Examiner Name	DOCKET CENTRAL
Attorney Docket Number	L0041B1CIP2

	20	9021583	B2	2015-04-28	Wittenstein et al.	
	21	9037273	B2	2015-05-19	Mikkelsen	
	22	9332020	B2	2016-05-03	Thomas et al.	
	23	9515999	B2	2016-12-06	Ylonen	
	24	9516053	B1	2016-12-06	Muddu et al.	
	25	9558265	B1	2017-01-31	Tacchi et al.	
	26	9596253	B2	2017-03-14	Chauhan et al.	
	27	9654503	B1	2017-05-16	Kowalyshyn	
	28	9734040	B2	2017-08-15	Gounares	
	29	9749339	B2	2017-08-29	Kadambe et al.	
	30	9853968	B2	2017-12-26	Shen et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	31	9953014	B1	2018-04-24	Reshadi et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20020184225	A1	2002-12-05	Ghukasyan	
	2	20030037136	A1	2003-02-20	Labovitz et al.	
	3	20030179227	A1	2003-09-25	Ahmad et al.	
	4	20040015470	A1	2004-01-22	Smith et al.	
	5	20050102284	A1	2005-05-12	Srinivasan et al.	
	6	20050188222	A1	2005-08-25	Motsinger et al.	
	7	20060085437	A1	2006-04-20	Brodhun et al.	
	8	20060259470	A1	2006-11-16	Chandrasekharan et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	9	20060288415	A1	2006-12-21	Wong	
	10	20070050497	A1	2007-03-01	Haley et al.	
	11	20070118909	A1	2007-05-24	Hertzog et al.	
	12	20080148180	A1	2008-06-19	Liu et al.	
	13	20080151893	A1	2008-06-26	Nordmark et al.	
	14	20090019160	A1	2009-01-15	Schuler	
	15	20090228474	A1	2009-09-10	Chiu et al.	
	16	20090271504	A1	2009-10-29	Ginter et al.	
	17	20100094767	A1	2010-04-15	Miltonberger	
	18	20100114931	A1	2010-05-06	Xie et al.	
	19	20100172261	A1	2010-07-08	Shinbo et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	20	20100274785	A1	2010-10-28	Procopiuc et al.	
	21	20110055138	A1	2011-03-03	Khanduja et al.	
	22	20120005243	A1	2012-01-05	Merwe et al.	
	23	20120317151	A1	2012-12-13	Ruf et al.	
	24	20120323956	A1	2012-12-20	Dumitru et al.	
	25	20130205357	A1	2013-08-08	Bahnck et al.	
	26	20130304915	A1	2013-11-14	KAWAI	
	27	20140115001	A1	2014-04-24	Arroyo et al.	
	28	20140115011	A1	2014-04-24	Buerner et al.	
	29	20140208191	A1	2014-07-24	Zaric et al.	
	30	20140245443	A1	2014-08-28	Chakraborty	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	OPAP
Examiner Name	DOCKET CENTRAL
Attorney Docket Number	L0041B1CIP2

	31	20140280068	A1	2014-09-18	DHOOPAR et al.	
	32	20140359558	A1	2014-12-04	Chamberlain	
	33	20150135312	A1	2015-05-14	Wada et al.	
	34	20150188751	A1	2015-07-02	Vasseur et al.	
	35	20150341379	A1	2015-11-26	Lefebvre et al.	
	36	20150356144	A1	2015-12-10	Chawla et al.	
	37	20160078365	A1	2016-03-17	Baumard	
	38	20160080404	A1	2016-03-17	KOHOUT et al.	
	39	20160110434	A1	2016-04-21	Kakaraddi et al.	
	40	20160149937	A1	2016-05-26	KATMOR et al.	
	41	20160205125	A1	2016-07-14	Kim et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	42	20160218911	A1	2016-07-28	WESSELS et al.	
	43	20160359592	A1	2016-12-08	Kulshreshtha et al.	
	44	20160359872	A1	2016-12-08	Yadav et al.	
	45	20170118240	A1	2017-04-27	Reddy et al.	
	46	20170134240	A1	2017-05-11	Hévizi et al.	
	47	20170147646	A1	2017-05-25	LEE et al.	
	48	20170163666	A1	2017-06-08	Venkatramani et al.	
	49	20170251013	A1	2017-08-31	Kirti et al.	
	50	20170272344	A1	2017-09-21	Tang et al.	
	51	20170279827	A1	2017-09-28	Savalle et al.	
	52	20170288974	A1	2017-10-05	Yoshihira et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	53	20170359361	A1	2017-12-14	Modani et al.	
	54	20170366492	A1	2017-12-21	Ho et al.	
	55	20180020015	A1	2018-01-18	Munro et al.	
	56	20180034840	A1	2018-02-01	Marquardt et al.	
	57	20180063178	A1	2018-03-01	Jadhav et al.	
	58	20180103052	A1	2018-04-12	Choudhury et al.	
	59	20180115578	A1	2018-04-26	Subbarayan et al.	
	60	20180137858	A1	2018-05-17	Saxena et al.	
	61	20180173789	A1	2018-06-21	Llagostera et al.	
	62	20180174062	A1	2018-06-21	Simo et al.	
	63	20180181750	A1	2018-06-28	Lamothe-Brassard	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	64	20180248901	A1	2018-08-30	Rieke	
	65	20180268078	A1	2018-09-20	Gianetto et al.	
	66	20180288063	A1	2018-10-04	Koottayi et al.	
	67	20180329958	A1	2018-11-15	Choudhury et al.	
	68	20180336353	A1	2018-11-22	Manadhata et al.	
	69	20180357422	A1	2018-12-13	Telang et al.	
	70	20180367548	A1	2018-12-20	Stokes III et al.	
	71	20190028327	A1	2019-01-24	SILVA et al.	
	72	20190109870	A1	2019-04-11	BEDHAPUDI et al.	
	73	20190132224	A1	2019-05-02	VERMA et al.	
	74	20190222597	A1	2019-07-18	Crabtree et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	75	20190259033	A1	2019-08-22	Reddy et al.	
	76	20190339965	A1	2019-11-07	Garvey et al.	
	77	20190349305	A1	2019-11-14	WANG et al.	
	78	20190378050	A1	2019-12-12	Edkin et al.	
	79	20200076685	A1	2020-03-05	Vaidya et al.	
	80	20200192690	A1	2020-06-18	Gupta et al.	
	81	20200252376	A1	2020-08-06	Feng et al.	
	82	20200272740	A1	2020-08-27	Obee et al.	
	83	20200278892	A1	2020-09-03	Nainar et al.	
	84	20200280592	A1	2020-09-03	Ithal et al.	
	85	20200311644	A1	2020-10-01	WILLARD III et al.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	OPAP
Examiner Name	DOCKET CENTRAL
Attorney Docket Number	L0041B1CIP2

	86	20200403860	A1	2020-12-24	Lewis et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	AI-YASEEN, et al., "Real-Time Intrusion Detection System Using Multi-Agent System", IAENG International Journal of Computer Science, volume 43, no. 1, February 2016, pp. 80-90, International Association of Engineers (IAENG), Hong Kong.	
	2	AKOGLU, et al., "Graph-based Anomaly Detection and Description: A Survey", April 28, 2014.	
	3	AMMAR, et al., "Query Optimization Techniques in Graph Databases", International Journal of Database Management Systems (IIDMS), Vol. 8, No. 4, August 2016, pages 1-14 (Year: 2016).	
	4	BALASUBRAMANIYAN, et al., "An Architecture For Intrusion Detection Using Autonomous Agents", In Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217), 19 pages, June 1998, IEEE, DOI: 10.1109/CSAC.1998.738563.	
	5	BEUTEL, et al., "User Behavior Modeling with Large-Scale Graph Analysis, Computer Science Department, Carnegie Mellon University, May 2016.	

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	6	BUGIEL, et al., "Towards Taming Privilege-Escalation Attacks on Android", In NOSS (Vol. 17, p. 19), 2012, February.
	7	CHANG, et al., "Reality Bites - Progressive Querying and Result Visualization in Logical and VR Spaces", Proceedings of 1994 IEEE Symposium on Visual Languages, pp. 100-109, October 1994, IEEE, DOI: 10.1109NL.1994.363635,
	8	CHESSON, et al., "Communication And Control In A Cluster Network", ACM '74: Proceedings of the 1974 annual ACM conference - Volume 2, January 1974, pp 509-514, http://doi.org/10.1145/1408839 (Year 1974).
	9	CROSBIE et al., "Defending a Computer System using Autonomous Agents", docs.lib.purdue.edu (online), March 1995, 11 pages.
	10	HAUTAMAKI, et al., "Outlier Detection Using k-Nearest Neighbour Graph", Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), Vol. 3, August 2004, IEEE, DOI: 10.1109/ICPR.2004.1334558.
	11	HOOPER, et al., "Medusa: a simple tool for interaction graph analysis", Bioinformatics, Vol. 21 no. 24, September 2005, pages 4432-4433, Oxford University Press (online), URL: https://academic.oup.com/bioinformatics/article/21/24/4432/179694.
	12	KOUTRA, et al., "Exploring and Making Sense of Large Graphs", Computer Science Department, Carnegie Mellon University, August 2015.
	13	LEOPOLD, et al., "A Visual Query System for the Specification and Scientific Analysis off Continual Queries", Proceedings, IEEE Symposia on Human-Centric Computing Languages and Environments (Cat. No.01TH8587), September 2001, pp. 203-211, IEEE, doi: 10.1109/HCC.2001.995260.
	14	LIAO, et al., "Visualizing Graph Dynamics And Similarity For Enterprise Network Security And Management", VizSec '10: Proceedings of the Seventh International Symposium on Visualization for Cyber Security, September 2010, pp. 34-45, URL: https://doi.org/10.1145/1850795.1850799.
	15	MATEESCU, et al., "Join-Graph Propagation Algorithms", Journal of Artificial Intelligence Research, vol. 37, March 2010, pages 279-328, AI Access Foundation, Inc. (online), URL: https://doi.org/10.1613/jair.2842.
	16	MORIANO, et al., "Insider Threat Event Detection in User-System Interactions", MIST '17: Proceedings of the 2017 International Workshop on Managing Insider Security Threats, October 2017, pp 1-12, ACM Digital Library (online), URL: https://doi.Jrg/10.1145/3139923.3139928.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	17	RANSHOUS, et al., "Anomaly detection in dynamic networks: a survey", WIREs Computational Statistics, volume 7, May/June 2015, pp. 223-247, Wiley Periodicals, Inc, United States.
	18	TAMASSIA, et al., "Graph Drawing for Security Visualization", In: Graph Drawing (GD 2008), Lecture Notes in Computer Science, vol 5417, Springer, Berlin, Heidelberg (online), URL: https://doi.org/10.1007/978-3-642-00219-9_2.
	19	VAAS, et al., "Detecting disguised processes using Application-Behavior Profiling", In 2017 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-6, June 2017, IEEE, DOI: 10.1109/THS.2017.7943508.
	20	YU, et al., "Recommending Join Queries Based on Path Frequency", 2015 12th Web Information System and Application Conference (WISA), September 2015, pp. 21-26, IEEE, DOI: 10.1109/WISA.2015.52.

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/ALINA A BOUTAH/	Date Considered	02/05/2023

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	OPAP
Examiner Name	DOCKET CENTRAL
Attorney Docket Number	L0041B1CIP2

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Edward Lenart/	Date (YYYY-MM-DD)	2022-10-03
Name/Print	Edward Lenart	Registration Number	63665

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /A.A.B/

(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114598840 A
(43)申请公布日 2022.06.07

(21)申请号 202210043236.8

(22)申请日 2022.01.14

(71)申请人 华元
地址 广东省深圳市福田区中康路128号卓越城一期一栋1205

(72)发明人 华元

(74)专利代理机构 北京知无忧专利代理有限公司 11880
专利代理师 李伟新

(51)Int.Cl.
H04N 7/18(2006.01)
H04W 4/029(2018.01)
H04L 67/10(2022.01)
H04L 67/52(2022.01)
G06N 3/04(2006.01)
G06N 3/08(2006.01)
G06K 9/62(2022.01)

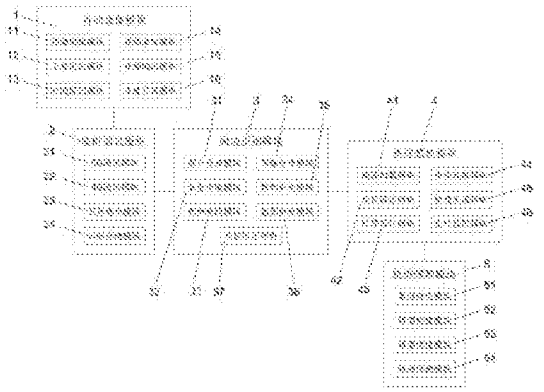
权利要求书2页 说明书5页 附图2页

(54)发明名称

智能徽章行动轨迹监控系统及方法

(57)摘要

本发明公开了智能徽章行动轨迹监控系统及方法，包括自动录像模块、实时通讯模块、周边识别模块、轨迹监控模块和轨迹预测模块，所述自动录像模块的输出端连接实时通讯模块的输入端，实时通讯模块的输出端连接周边识别模块的输入端，周边识别模块的输出端连接轨迹监控模块的输入端，轨迹监控模块的输出端连接轨迹预测模块的输入端；该发明可实时监控徽章佩带者的行动轨迹，在佩带者走丢或失踪的情况下，他人易于寻找和搜救，保障了佩带者的人身安全，实用性强，且利用人工智能算法构建并优化出行动轨迹预测模型，即便徽章失去了信号，也能根据当前行动轨迹预测徽章佩带者的未来行动轨迹，工作可靠，使用范围广。



CN 114598840 A

1.智能徽章行动轨迹监控系统，包括自动录像模块(1)、实时通讯模块(2)、周边识别模块(3)、轨迹监控模块(4)和轨迹预测模块(5)，其特征在于：所述自动录像模块(1)的输出端连接实时通讯模块(2)的输入端，实时通讯模块(2)的输出端连接周边识别模块(3)的输入端，周边识别模块(3)的输出端连接轨迹监控模块(4)的输入端，轨迹监控模块(4)的输出端连接轨迹预测模块(5)的输入端。

2.根据权利要求1所述的智能徽章行动轨迹监控系统，其特征在于：所述自动录像模块(1)由图像采集模块(11)、卫星定位模块(12)、时间标记模块(13)、视频录制模块(14)、视频编码模块(15)和本地存储模块(16)组成，图像采集模块(11)的输出端连接卫星定位模块(12)的输入端，卫星定位模块(12)的输出端连接时间标记模块(13)的输入端，时间标记模块(13)的输出端连接视频录制模块(14)的输入端，视频录制模块(14)的输出端连接视频编码模块(15)的输入端，视频编码模块(15)的输出端连接本地存储模块(16)的输入端，本地存储模块(16)的输出端连接实时通讯模块(2)的输入端。

3.根据权利要求2所述的智能徽章行动轨迹监控系统，其特征在于：所述实时通讯模块(2)由5G通讯模块(21)、4G通讯模块(22)、卫星通讯模块(23)、云端存储模块(24)组成，5G通讯模块(21)、4G通讯模块(22)和卫星通讯模块(23)的输入端均连接本地存储模块(16)的输出端，5G通讯模块(21)、4G通讯模块(22)和卫星通讯模块(23)的输出端均连接云端存储模块(24)的输入端，云端存储模块(24)的输出端连接周边识别模块(3)的输入端。

4.根据权利要求3所述的智能徽章行动轨迹监控系统，其特征在于：所述周边识别模块(3)由用户登录模块(31)、录像调取模块(32)、视频解码模块(33)、图像处理模块(34)、场景识别模块(35)、地图参照模块(36)和点位校正模块(37)组成，用户登录模块(31)的输入端连接云端存储模块(24)的输出端，用户登录模块(31)的输出端连接录像调取模块(32)的输入端，录像调取模块(32)的输出端连接视频解码模块(33)的输入端，视频解码模块(33)的输出端连接图像处理模块(34)的输入端，图像处理模块(34)的输出端连接场景识别模块(35)的输入端，场景识别模块(35)的输出端连接地图参照模块(36)的输入端，地图参照模块(36)的输出端连接点位校正模块(37)的输入端，点位校正模块(37)的输出端连接轨迹监控模块(4)的输入端。

5.根据权利要求4所述的智能徽章行动轨迹监控系统，其特征在于：所述轨迹监控模块(4)由地图加载模块(41)、点位锚定模块(42)、时序标注模块(43)、点位连线模块(44)、轨迹生成模块(45)和实时监控模块(46)组成，地图加载模块(41)的输入端连接点位校正模块(37)的输出端，地图加载模块(41)的输出端连接点位锚定模块(42)的输入端，点位锚定模块(42)的输出端连接时序标注模块(43)的输入端，时序标注模块(43)的输出端连接点位连线模块(44)的输入端，点位连线模块(44)的输出端连接轨迹生成模块(45)的输入端，轨迹生成模块(45)的输出端连接实时监控模块(46)的输入端，实时监控模块(46)的输出端连接轨迹预测模块(5)的输入端。

6.根据权利要求5所述的智能徽章行动轨迹监控系统，其特征在于：所述轨迹预测模块(5)由数据筛选模块(51)、模型构建模块(52)、模型优化模块(53)和实时预测模块(54)组成，数据筛选模块(51)的输入端连接实时监控模块(46)的输出端，数据筛选模块(51)的输出端连接模型构建模块(52)的输入端，模型构建模块(52)的输出端连接模型优化模块(53)的输入端，模型优化模块(53)的输出端连接实时预测模块(54)的输入端。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00355

7.智能徽章行动轨迹监控方法，包括以下步骤：步骤一，录像；步骤二，通讯；步骤三，识别；步骤四，监控；步骤五，预测；其特征在于：

其中上述步骤一中，通过图像采集模块(11)采集到徽章前方区域的光影图像，并通过卫星定位模块(12)确定光影图像的拍摄地点，再通过时间标记模块(13)给光影图像标记上时间戳，并通过视频录制模块(14)录制徽章前方区域一段时间的影像，经视频编码模块(15)编码后保存到本地存储模块(16)中；

其中上述步骤二中，根据信号强度通过5G通讯模块(21)、4G通讯模块(22)或卫星通讯模块(23)将编码录像上传给云端，并保存在云端存储模块(24)中；

其中上述步骤三中，通过用户登录模块(31)登入云端的监控系统，并通过录像调取模块(32)调取编码录像，经视频解码模块(33)解码、图像处理模块(34)灰度处理后，再通过场景识别模块(35)识别出录像中的各种场景事物，并通过地图参照模块(36)比照卫星地图中的街景数据，进而通过点位校正模块(37)校正徽章在电子地图上所处的点位；

其中上述步骤四中，通过地图加载模块(41)加载卫星地图，并通过点位锚定模块(42)确定各个行动点位在卫星地图上的位置，再通过时序标注模块(43)标注上各个行动点位的时序，并根据时序通过点位连线模块(44)将各个行动点位连接成线，经轨迹生成模块(45)处理后持续的在卫星地图上生成徽章的行动轨迹，进而通过实时监控模块(46)实时监控徽章的行动轨迹；

其中上述步骤五中，通过数据筛选模块(51)从徽章的历史行动轨迹中筛选出与当前行动相关性较强的历史数据，并通过模型构建模块(52)利用深度学习算法构建出轨迹预测初始模型，再通过模型优化模块(53)利用当前行动已有数据对轨迹预测初始模型进行训练，优化出行动轨迹预测模型，并通过实时预测模块(54)预测出徽章之后的行动轨迹。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00356

智能徽章行动轨迹监控系统及方法

技术领域

[0001] 本发明涉及行动轨迹监控技术领域，具体为智能徽章行动轨迹监控系统及方法。

背景技术

[0002] 徽章，是一种佩带在身上用来表示身份、职业、荣誉等含义的标志。而智能徽章，顾名思义指的是一种内嵌有微芯片的徽章。目前，市面上常见的智能徽章主要具有灯光、定位、通讯、录像等功能。

[0003] 然而，现有的智能徽章大多无法监控徽章佩带者的行动轨迹，在佩带者走丢或失踪的情况下，他人寻找、搜救困难，且不具备图像识别功能，无法对徽章佩带者周边的场景事物进行识别，增加了他人寻找、搜救的难度，满足了不同佩带者的使用需求，另外徽章失去信号时，他人无法根据当前行动轨迹预测徽章佩带者的未来行动轨迹，工作可靠性差，使用范围有限。

发明内容

[0004] 本发明的目的在于提供智能徽章行动轨迹监控系统及方法，以解决上述背景技术中提出的问题。

[0005] 为实现上述目的，本发明提供如下技术方案：智能徽章行动轨迹监控系统，包括自动录像模块、实时通讯模块、周边识别模块、轨迹监控模块和轨迹预测模块，所述自动录像模块的输出端连接实时通讯模块的输入端，实时通讯模块的输出端连接周边识别模块的输入端，周边识别模块的输出端连接轨迹监控模块的输入端，轨迹监控模块的输出端连接轨迹预测模块的输入端。

[0006] 优选的，所述自动录像模块由图像采集模块、卫星定位模块、时间标记模块、视频录制模块、视频编码模块和本地存储模块组成，图像采集模块的输出端连接卫星定位模块的输入端，卫星定位模块的输出端连接时间标记模块的输入端，时间标记模块的输出端连接视频录制模块的输入端，视频录制模块的输出端连接视频编码模块的输入端，视频编码模块的输出端连接本地存储模块的输入端，本地存储模块的输出端连接实时通讯模块的输入端。

[0007] 优选的，所述实时通讯模块由5G通讯模块、4G通讯模块、卫星通讯模块、云端存储模块组成，5G通讯模块、4G通讯模块和卫星通讯模块的输入端均连接本地存储模块的输出端，5G通讯模块、4G通讯模块和卫星通讯模块的输出端均连接云端存储模块的输入端，云端存储模块的输出端连接周边识别模块的输入端。

[0008] 优选的，所述周边识别模块由用户登录模块、录像调取模块、视频解码模块、图像处理模块、场景识别模块、地图参照模块和点位校正模块组成，用户登录模块的输入端连接云端存储模块的输出端，用户登录模块的输出端连接录像调取模块的输入端，录像调取模块的输出端连接视频解码模块的输入端，视频解码模块的输出端连接图像处理模块的输入端，图像处理模块的输出端连接场景识别模块的输入端，场景识别模块的输出端连接地图

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00357

参照模块的输入端，地图参照模块的输出端连接点位校正模块的输入端，点位校正模块的输出端连接轨迹监控模块的输入端。

[0009] 优选的，所述轨迹监控模块由地图加载模块、点位锚定模块、时序标注模块、点位连线模块、轨迹生成模块和实时监控模块组成，地图加载模块的输入端连接点位校正模块的输出端，地图加载模块的输出端连接点位锚定模块的输入端，点位锚定模块的输出端连接时序标注模块的输入端，时序标注模块的输出端连接点位连线模块的输入端，点位连线模块的输出端连接轨迹生成模块的输入端，轨迹生成模块的输出端连接实时监控模块的输入端，实时监控模块的输出端连接轨迹预测模块的输入端。

[0010] 优选的，所述轨迹预测模块由数据筛选模块、模型构建模块、模型优化模块和实时预测模块组成，数据筛选模块的输入端连接实时监控模块的输出端，数据筛选模块的输出端连接模型构建模块的输入端，模型构建模块的输出端连接模型优化模块的输入端，模型优化模块的输出端连接实时预测模块的输入端。

[0011] 智能徽章行动轨迹监控方法，包括以下步骤：步骤一，录像；步骤二，通讯；步骤三，识别；步骤四，监控；步骤五，预测；

[0012] 其中上述步骤一中，通过图像采集模块采集到徽章前方区域的光影图像，并通过卫星定位模块确定光影图像的拍摄地点，再通过时间标记模块给光影图像标记上时间戳，并通过视频录制模块录制徽章前方区域一段时间的影像，经视频编码模块编码后保存到本地存储模块中；

[0013] 其中上述步骤二中，根据信号强度通过5G通讯模块、4G通讯模块或卫星通讯模块将编码录像上传给云端，并保存在云端存储模块中；

[0014] 其中上述步骤三中，通过用户登录模块登入云端的监控系统，并通过录像调取模块调取编码录像，经视频解码模块解码、图像处理模块灰度处理后，再通过场景识别模块识别出录像中的各种场景事物，并通过地图参照模块比照卫星地图中的街景数据，进而通过点位校正模块校正徽章在电子地图上所处的点位；

[0015] 其中上述步骤四中，通过地图加载模块加载卫星地图，并通过点位锚定模块确定各个行动点位在卫星地图上的位置，再通过时序标注模块标注上各个行动点位的时序，并根据时序通过点位连线模块将各个行动点位连接成线，经轨迹生成模块处理后持续的在卫星地图上生成徽章的行动轨迹，进而通过实时监控模块实时监控徽章的行动轨迹；

[0016] 其中上述步骤五中，通过数据筛选模块从徽章的历史行动轨迹中筛选出与当前行动相关性较强的历史数据，并通过模型构建模块利用深度学习算法构建出轨迹预测初始模型，再通过模型优化模块利用当前行动已有数据对轨迹预测初始模型进行训练，优化出行动轨迹预测模型，并通过实时预测模块预测出徽章之后的行动轨迹。

[0017] 与现有技术相比，本发明的有益效果是：该智能徽章行动轨迹监控系统及方法，实时监控徽章佩带者的行动轨迹，在佩带者走丢或失踪的情况下，他人易于寻找和搜救，保障了佩带者的人身安全；具备图像识别功能，能对徽章佩带者周边的场景事物进行识别，定位更加准确，降低了他人寻找、搜救的难度，满足了不同佩带者的使用需求；利用人工智能算法构建并优化出行动轨迹预测模型，徽章失去信号时，也能根据当前行动轨迹预测徽章佩带者的未来行动轨迹，工作可靠，使用范围广。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00358

附图说明

[0018] 图1为本发明的系统结构图；

[0019] 图2为本发明的系统流程图；

[0020] 图3为本发明的方法流程图；

[0021] 图中：1、自动录像模块；11、图像采集模块；12、卫星定位模块；13、时间标记模块；14、视频录制模块；15、视频编码模块；16、本地存储模块；2、实时通讯模块；21、5G通讯模块；22、4G通讯模块；23、卫星通讯模块；24、云端存储模块；3、周边识别模块；31、用户登录模块；32、录像调取模块；33、视频解码模块；34、图像处理模块；35、场景识别模块；36、地图参照模块；37、点位校正模块；4、轨迹监控模块；41、地图加载模块；42、点位锚定模块；43、时序标注模块；44、点位连线模块；45、轨迹生成模块；46、实时监控模块；5、轨迹预测模块；51、数据筛选模块；52、模型构建模块；53、模型优化模块；54、实时预测模块。

具体实施方式

[0022] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0023] 请参阅图1-2，本发明提供的一种实施例：智能徽章行动轨迹监控系统，包括自动录像模块1、实时通讯模块2、周边识别模块3、轨迹监控模块4和轨迹预测模块5，自动录像模块1的输出端连接实时通讯模块2的输入端，实时通讯模块2的输出端连接周边识别模块3的输入端，周边识别模块3的输出端连接轨迹监控模块4的输入端，轨迹监控模块4的输出端连接轨迹预测模块5的输入端，自动录像模块1由图像采集模块11、卫星定位模块12、时间标记模块13、视频录制模块14、视频编码模块15和本地存储模块16组成，图像采集模块11的输出端连接卫星定位模块12的输入端，卫星定位模块12的输出端连接时间标记模块13的输入端，时间标记模块13的输出端连接视频录制模块14的输入端，视频录制模块14的输出端连接视频编码模块15的输入端，视频编码模块15的输出端连接本地存储模块16的输入端，本地存储模块16的输出端连接实时通讯模块2的输入端，实时通讯模块2由5G通讯模块21、4G通讯模块22、卫星通讯模块23、云端存储模块24组成，5G通讯模块21、4G通讯模块22和卫星通讯模块23的输入端均连接本地存储模块16的输出端，5G通讯模块21、4G通讯模块22和卫星通讯模块23的输出端均连接云端存储模块24的输入端，云端存储模块24的输出端连接周边识别模块3的输入端，周边识别模块3由用户登录模块31、录像调取模块32、视频解码模块33、图像处理模块34、场景识别模块35、地图参照模块36和点位校正模块37组成，用户登录模块31的输入端连接云端存储模块24的输出端，用户登录模块31的输出端连接录像调取模块32的输入端，录像调取模块32的输出端连接视频解码模块33的输入端，视频解码模块33的输出端连接图像处理模块34的输入端，图像处理模块34的输出端连接场景识别模块35的输入端，场景识别模块35的输出端连接地图参照模块36的输入端，地图参照模块36的输出端连接点位校正模块37的输入端，点位校正模块37的输出端连接轨迹监控模块4的输入端，具备图像识别功能，能对徽章佩带者周边的场景事物进行识别，定位更加准确，降低了他人寻找、搜救的难度，满足了不同佩带者的使用需求；轨迹监控模块4由地图加载模块41、点位

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00359

锚定模块42、时序标注模块43、点位连线模块44、轨迹生成模块45和实时监控模块46组成，地图加载模块41的输入端连接点位校正模块37的输出端，地图加载模块41的输出端连接点位锚定模块42的输入端，点位锚定模块42的输出端连接时序标注模块43的输入端，时序标注模块43的输出端连接点位连线模块44的输入端，点位连线模块44的输出端连接轨迹生成模块45的输入端，轨迹生成模块45的输出端连接实时监控模块46的输入端，实时监控模块46的输出端连接轨迹预测模块5的输入端，实时监控徽章佩带者的行动轨迹，在佩带者走丢或失踪的情况下，他人易于寻找和搜救，保障了佩带者的人身安全；轨迹预测模块5由数据筛选模块51、模型构建模块52、模型优化模块53和实时预测模块54组成，数据筛选模块51的输入端连接实时监控模块46的输出端，数据筛选模块51的输出端连接模型构建模块52的输入端，模型构建模块52的输出端连接模型优化模块53的输入端，模型优化模块53的输出端连接实时预测模块54的输入端，徽章失去信号时，也能根据当前行动轨迹预测徽章佩带者的未来行动轨迹，工作可靠，使用范围广。

[0024] 请参阅图3，本发明提供的一种实施例：智能徽章行动轨迹监控方法，包括以下步骤：步骤一，录像；步骤二，通讯；步骤三，识别；步骤四，监控；步骤五，预测；

[0025] 其中上述步骤一中，通过图像采集模块11采集到徽章前方区域的光影图像，并通过卫星定位模块12确定光影图像的拍摄地点，再通过时间标记模块13给光影图像标记上时间戳，并通过视频录制模块14录制徽章前方区域一段时间的影像，经视频编码模块15编码后保存到本地存储模块16中；

[0026] 其中上述步骤二中，根据信号强度通过5G通讯模块21、4G通讯模块22或卫星通讯模块23将编码录像上传给云端，并保存在云端存储模块24中；

[0027] 其中上述步骤三中，通过用户登录模块31登入云端的监控系统，并通过录像调取模块32调取编码录像，经视频解码模块33解码、图像处理模块34灰度处理后，再通过场景识别模块35识别出录像中的各种场景事物，并通过地图参照模块36比照卫星地图中的街景数据，进而通过点位校正模块37校正徽章在电子地图上所处的点位；

[0028] 其中上述步骤四中，通过地图加载模块41加载卫星地图，并通过点位锚定模块42确定各个行动点位在卫星地图上的位置，再通过时序标注模块43标注上各个行动点位的时序，并根据时序通过点位连线模块44将各个行动点位连接成线，经轨迹生成模块45处理后持续的在卫星地图上生成徽章的行动轨迹，进而通过实时监控模块46实时监控徽章的行动轨迹；

[0029] 其中上述步骤五中，通过数据筛选模块51从徽章的历史行动轨迹中筛选出与当前行动相关性较强的历史数据，并通过模型构建模块52利用深度学习算法构建出轨迹预测初始模型，再通过模型优化模块53利用当前行动已有数据对轨迹预测初始模型进行训练，优化出行动轨迹预测模型，并通过实时预测模块54预测出徽章之后的行动轨迹。

[0030] 工作原理：该发明使用时，先通过图像采集模块11采集到徽章前方区域的光影图像，并通过卫星定位模块12确定光影图像的拍摄地点，再通过时间标记模块13给光影图像标记上时间戳，并通过视频录制模块14录制徽章前方区域一段时间的影像，经视频编码模块15编码后保存到本地存储模块16中；接着根据信号强度通过5G通讯模块21、4G通讯模块22或卫星通讯模块23将编码录像上传给云端，并保存在云端存储模块24中；然后通过用户登录模块31登入云端的监控系统，并通过录像调取模块32调取编码录像，经视频解码模块

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00360

33解码、图像处理模块34灰度处理后，再通过场景识别模块35识别出录像中的各种场景事物，并通过地图参照模块36比照卫星地图中的街景数据，进而通过点位校正模块37校正徽章在电子地图上所处的点位；最后通过地图加载模块41加载卫星地图，并通过点位锚定模块42确定各个行动点位在卫星地图上的位置，再通过时序标注模块43标注上各个行动点位的时序，并根据时序通过点位连线模块44将各个行动点位连接成线，经轨迹生成模块45处理后持续的在卫星地图上生成徽章的行动轨迹，进而通过实时监控模块46实时监控徽章的行动轨迹；同时可通过数据筛选模块51从徽章的历史行动轨迹中筛选出与当前行动相关性较强的历史数据，并通过模型构建模块52利用深度学习算法构建出轨迹预测初始模型，再通过模型优化模块53利用当前行动已有数据对轨迹预测初始模型进行训练，优化出行动轨迹预测模型，并通过实时预测模块54预测出徽章之后的行动轨迹，徽章失去信号时，也能根据当前行动轨迹预测徽章佩带者的未来行动轨迹，工作可靠，使用范围广，降低了他人寻找、搜救的难度，保障了佩带者的人身安全。

[0031] 对于本领域技术人员而言，显然本发明不限于上述示范性实施例的细节，而且在不背离本发明的精神或基本特征的情况下，能够以其他的具体形式实现本发明。因此，无论从哪一点来看，均应将实施例看作是示范性的，而且是非限制性的，本发明的范围由所附权利要求而不是上述说明限定，因此旨在将落在权利要求的等同要件的含义和范围内的所有变化囊括在本发明内。不应将权利要求中的任何附图标记视为限制所涉及的权利要求。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00361

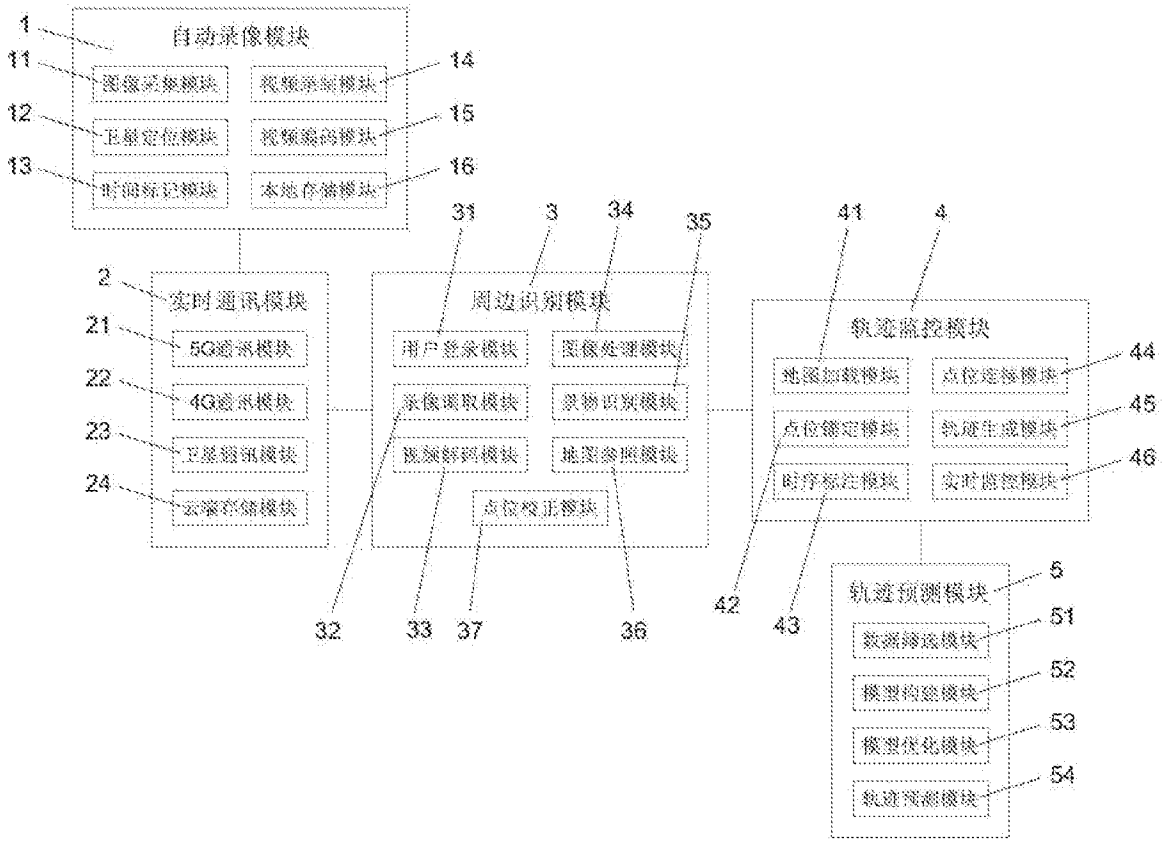


图1

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00362

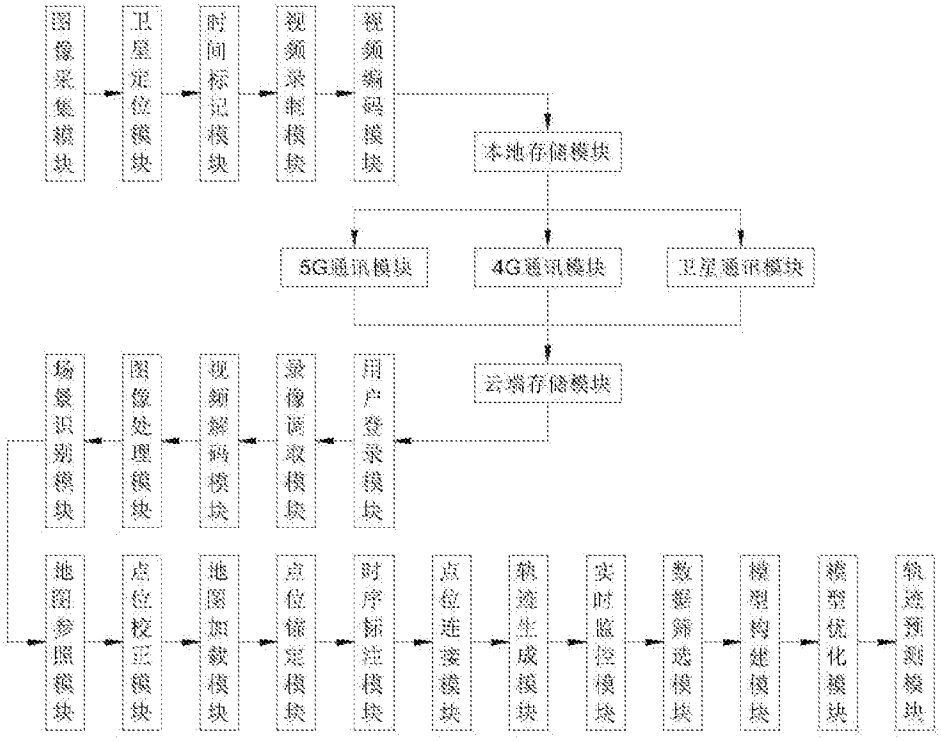


图2

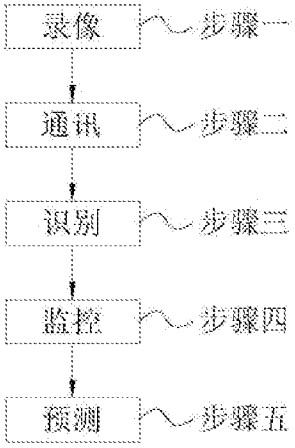


图3

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00363

Access provided by: United States Patent and Trademark Office

Sign Out

Showing 1-25 of 115 results for cloud deploy* learn* similar* ×

Conferences (79) Journals (33) Magazines (2) Books (1)

Filter

Alert Set Search Alerts | sort: 1 Sort Relevance

Show

All Results

Subscribed Content

Open Access Only

Year

Single Year Range

2013 2013 2023 2023

From 2013 To 2023

Author

Affiliation

Publication Title

Publisher

Supplemental Items

Conference Location

A Learning Automata-based Scheduling for Deadline Sensitive Task in The Cloud

Sampa Sahoo; Bibhudatta Sahoo; Ashok Kumar Turuk

2021 IEEE World Congress on Services (SERVICES)

Year: 2021 | Conference Paper | Publisher: IEEE

Cited by: Papers (2)

Abstract

A Learning Automata-based Scheduling for Deadline Sensitive Task in The Cloud

Sampa Sahoo; Bibhudatta Sahoo; Ashok Kumar Turuk

2021 IEEE World Congress on Services (SERVICES)

Year: 2021 | Conference Paper | Publisher: IEEE

Accelerating Platform Deployments in the Cloud: A Qualitative Assessment Based on CORD

Angelos Mimidis Kentis; Jose Soler; Adam Broadbent; Paul Veitch

2020 Seventh International Conference on Software Defined Systems (SDS)

Year: 2020 | Conference Paper | Publisher: IEEE

Abstract HTML

Accelerating Platform Deployments in the Cloud: A Qualitative Assessment Based on CORD

Angelos Mimidis Kentis; Jose Soler; Adam Broadbent; Paul Veitch

2020 Seventh International Conference on Software Defined Systems (SDS)

Year: 2020 | Conference Paper | Publisher: IEEE

IMITA: Imitation Learning for Generalizing Cloud Orchestration

Kamal Hakimzadehy; Patrick K. Nicholsonz; Diego Lugonesz; Amir H. Payberahy

2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)

Year: 2021 | Conference Paper | Publisher: IEEE

Publication Topics

Abstract HTML

IMITA: Imitation Learning for Generalizing Cloud Orchestration

Kamal Hakimzadehy; Patrick K. Nicholsonz; Diego Lugonesz; Amir H. Payberahy

2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)

Year: 2021 | Conference Paper | Publisher: IEEE

Service Clustering for Autonomic Clouds Using Random Forest

Rafael Brundo Uriarte; Sotirios Tsaftaris; Francesco Tiezzi

2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing

Year: 2015 | Conference Paper | Publisher: IEEE

Cited by: Papers (14) | Patents (1)

Abstract HTML

Service Clustering for Autonomic Clouds Using Random Forest

Rafael Brundo Uriarte; Sotirios Tsaftaris; Francesco Tiezzi

2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing

Year: 2015 | Conference Paper | Publisher: IEEE

Federated Imitation Learning: A Novel Framework for Cloud Robotic Systems With Heterogeneous Sensor Data

Boyi Liu; Lujia Wang; Ming Liu; Cheng-Zhong Xu

IEEE Robotics and Automation Letters

Year: 2020 | Volume: 5, Issue: 2 | Journal Article | Publisher: IEEE

Cited by: Papers (24)

Abstract HTML Media

Federated Imitation Learning: A Novel Framework for Cloud Robotic Systems With Heterogeneous Sensor Data

Boyi Liu; Lujia Wang; Ming Liu; Cheng-Zhong Xu

IEEE Robotics and Automation Letters

Year: 2020 | Volume: 5, Issue: 2 | Journal Article | Publisher: IEEE

Learn to Optimize: Adaptive VNF Provisioning in Mobile Edge Clouds

Qiufen Xia; Wenhao Ren; Zichuan Xu; Pan Zhou; Wenzheng Xu; Guowei Wu

2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)

Year: 2020 | Conference Paper | Publisher: IEEE

Cited by: Papers (2)

Abstract HTML

Learn to Optimize: Adaptive VNF Provisioning in Mobile Edge Clouds

Qiufen Xia; Wenhao Ren; Zichuan Xu; Pan Zhou; Wenzheng Xu; Guowei Wu

2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)

Year: 2020 | Conference Paper | Publisher: IEEE

FedMax: Enabling a Highly-Efficient Federated Learning Framework

Haohang Xu; Jin Li; Hongkai Xiong; Hui Lu

2020 IEEE 13th International Conference on Cloud Computing (CLOUD)

Year: 2020 | Conference Paper | Publisher: IEEE

Cited by: Papers (6)

Abstract HTML

FedMax: Enabling a Highly-Efficient Federated Learning Framework

Haohang Xu; Jin Li; Hongkai Xiong; Hui Lu

2020 IEEE 13th International Conference on Cloud Computing (CLOUD)

Year: 2020 | Conference Paper | Publisher: IEEE

Ensemble Deep Learning Assisted VNF Deployment Strategy for Next-Generation IoT Services

Mahzabeen Emu; Salimur Choudhury

IEEE Open Journal of the Computer Society

Year: 2021 | Volume: 2 | Journal Article | Publisher: IEEE

Cited by: Papers (3)

Abstract HTML

Ensemble Deep Learning Assisted VNF Deployment Strategy for Next-Generation IoT Services

Mahzabeen Emu; Salimur Choudhury

IEEE Open Journal of the Computer Society

Year: 2021 | Volume: 2 | Journal Article | Publisher: IEEE

Infrastructure as Code for Heterogeneous Computing

Daniel Vladusic; Dragan Radolovic

2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)

Year: 2020 | Conference Paper | Publisher: IEEE

Cited by: Papers (1)

Abstract

Infrastructure as Code for Heterogeneous Computing

Daniel Vladusic; Dragan Radolovic

2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)

Year: 2020 | Conference Paper | Publisher: IEEE

Exploiting Computation Reuse in Cloud-Based Deep Learning via Input Reordering

Enting Guo; Peng Li; Kun Wang; Huibin Feng; Jingyuan Lu; Song Guo

ICC 2020 - 2020 IEEE International Conference on Communications (ICC)

Year: 2020 | Conference Paper | Publisher: IEEE

Cited by: Papers (2)

Abstract HTML

Exploiting Computation Reuse in Cloud-Based Deep Learning via Input Reordering

Enting Guo; Peng Li; Kun Wang; Huibin Feng; Jingyuan Lu; Song Guo

ICC 2020 - 2020 IEEE International Conference on Communications (ICC)

Year: 2020 | Conference Paper | Publisher: IEEE

Supporting Autonomic Management of Clouds: Service Clustering With Random Forest

Rafael Brundo Uriarte; Francesco Tiezzi; Sotirios A. Tsaftaris

IEEE Transactions on Network and Service Management

Year: 2016 | Volume: 13, Issue: 3 | Journal Article | Publisher: IEEE

Cited by: Papers (14)

Abstract HTML

Supporting Autonomic Management of Clouds: Service Clustering With Random Forest

Rafael Brundo Uriarte; Francesco Tiezzi; Sotirios A. Tsaftaris

IEEE Transactions on Network and Service Management

Year: 2016 | Volume: 13, Issue: 3 | Journal Article | Publisher: IEEE

Poster Abstract: Guardian - A Smart Cloud Fabric for Virtual Patching and Customized Defenses

Rashid Tahir; Abdur Rahman Al-Ahmadi; Bashayyer Al-Sharyoufi; Rajin Azzouni; Fareed Zaffar

IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)

Year: 2019 | Conference Paper | Publisher: IEEE

Abstract HTML

Poster Abstract: Guardian - A Smart Cloud Fabric for Virtual Patching and Customized Defenses

Rashid Tahir; Abdur Rahman Al-Ahmadi; Bashayyer Al-Sharyoufi; Rajin Azzouni; Fareed Zaffar

IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)

Year: 2019 | Conference Paper | Publisher: IEEE

On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps

Satvik Garg; Pradyumn Pundir; Geetanjali Rathee; P.K. Gupta; Somya Garg; Saransh Ahlawat

2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)

Year: 2021 | Conference Paper | Publisher: IEEE

Cited by: Papers (2)

Abstract HTML

On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps

Satvik Garg; Pradyumn Pundir; Geetanjali Rathee; P.K. Gupta; Somya Garg; Saransh Ahlawat

2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)

Year: 2021 | Conference Paper | Publisher: IEEE

Reinforcement-Learning-Empowered MLaaS Scheduling for Serving Intelligent Internet of Things

Heyang Qin; Syed Zawad; Yanqi Zhou; Sanjay Padhi; Lei Yang; Feng Yan

IEEE Internet of Things Journal

Year: 2020 | Volume: 7, Issue: 7 | Journal Article | Publisher: IEEE

Cited by: Papers (7)

Abstract HTML

Reinforcement-Learning-Empowered MLaaS Scheduling for Serving Intelligent Internet of Things

Heyang Qin; Syed Zawad; Yanqi Zhou; Sanjay Padhi; Lei Yang; Feng Yan

IEEE Internet of Things Journal

Year: 2020 | Volume: 7, Issue: 7 | Journal Article | Publisher: IEEE

Ensemble Deep Learning Aided VNF Deployment for IoT Services

Mahzabeen Emu; Salimur Choudhury

2020 16th International Conference on Network and Service Management (CNSM)

Year: 2020 | Conference Paper | Publisher: IEEE
Cited by: Papers (2)

Abstract HTML

Ensemble Deep Learning Aided VNF Deployment for IoT Services

Mahzabeen Emu; Salimur Choudhury
2020 16th International Conference on Network and Service Management (CNSM)
Year: 2020 | Conference Paper | Publisher: IEEE

Measurement and Analysis of Cloud User Interest: A Glance From BitTorrent
Lei Ding; Yang Li; Haiyang Wang; Ke Xu
IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)
Year: 2020 | Conference Paper | Publisher: IEEE

Abstract HTML

Measurement and Analysis of Cloud User Interest: A Glance From BitTorrent

Lei Ding; Yang Li; Haiyang Wang; Ke Xu
IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)
Year: 2020 | Conference Paper | Publisher: IEEE

A Practical Approach to Hard Disk Failure Prediction in Cloud Platforms: Big Data Model for Failure Management in Datacenters
Sandipan Ganguly; Ashish Consul; Ali Khan; Brian Bussone; Jacqueline Richards; Alejandro Miguel
2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)
Year: 2016 | Conference Paper | Publisher: IEEE
Cited by: Papers (22) | Patents (1)

Abstract HTML

A Practical Approach to Hard Disk Failure Prediction in Cloud Platforms: Big Data Model for Failure Management in Datacenters

Sandipan Ganguly; Ashish Consul; Ali Khan; Brian Bussone; Jacqueline Richards; Alejandro Miguel
2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)
Year: 2016 | Conference Paper | Publisher: IEEE

Cloud Task Scheduling Based on Policy Gradient Algorithm in Heterogeneous Cloud Data Center for Energy Consumption Optimization
CuiHua He; YongQuan Yang; BoWei Hong
2020 International Conference on Internet of Things and Intelligent Applications (ITIA)
Year: 2020 | Conference Paper | Publisher: IEEE

Abstract HTML

Cloud Task Scheduling Based on Policy Gradient Algorithm in Heterogeneous Cloud Data Center for Energy Consumption Optimization

CuiHua He; YongQuan Yang; BoWei Hong
2020 International Conference on Internet of Things and Intelligent Applications (ITIA)
Year: 2020 | Conference Paper | Publisher: IEEE

Deep Residual Network with Multi-Image Attention for Imputing Under Clouds in Satellite Imagery

Xian Yang; Yifan Zhao; Ranga Raju Vatsavai
2022 26th International Conference on Pattern Recognition (ICPR)
Year: 2022 | Conference Paper | Publisher: IEEE

Abstract HTML

Deep Residual Network with Multi-Image Attention for Imputing Under Clouds in Satellite Imagery

Xian Yang; Yifan Zhao; Ranga Raju Vatsavai
2022 26th International Conference on Pattern Recognition (ICPR)
Year: 2022 | Conference Paper | Publisher: IEEE

Secure Framework for Cloud based E-Education using Deep Neural Networks
K Devi Priya; L Sumalatha
2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)
Year: 2021 | Conference Paper | Publisher: IEEE
Cited by: Papers (2)

Abstract HTML

Secure Framework for Cloud based E-Education using Deep Neural Networks

K Devi Priya; L Sumalatha
2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)
Year: 2021 | Conference Paper | Publisher: IEEE

Machine learning for network automation: overview, architecture, and applications [Invited Tutorial]
Danish Rafique; Luis Velasco
Journal of Optical Communications and Networking
Year: 2018 | Volume: 10, Issue: 10 | Journal Article | Publisher: IEEE
Cited by: Papers (81)

Abstract

Machine learning for network automation: overview, architecture, and applications [Invited Tutorial]

Danish Rafique; Luis Velasco
Journal of Optical Communications and Networking
Year: 2018 | Volume: 10, Issue: 10 | Journal Article | Publisher: IEEE

Prediction model for suitability of resource deployment using complex templates in OpenStack
Shefali Jain; Sateesh K Peddoju
2017 International Conference on Computing, Communication and Automation (ICCCA)
Year: 2017 | Conference Paper | Publisher: IEEE
Cited by: Papers (1)

Abstract

Prediction model for suitability of resource deployment using complex templates in OpenStack

Shefali Jain; Sateesh K Peddoju
2017 International Conference on Computing, Communication and Automation (ICCCA)
Year: 2017 | Conference Paper | Publisher: IEEE

Salient Object Detection in the Distributed Cloud-Edge Intelligent Network
Zhifan Gao; Heye Zhang; Shizhou Dong; Shanhui Sun; Xin Wang; Guang Yang; Wanqing Wu; Shuo Li; Victor Hugo C. de Albuquerque
IEEE Network

Year: 2020 | Volume: 34, Issue: 2 | Magazine Article | Publisher: IEEE
Cited by: Papers (49)

Abstract HTML

Salient Object Detection in the Distributed Cloud-Edge Intelligent Network

Zhifan Gao; Heye Zhang; Shizhou Dong; Shanhui Sun; Xin Wang; Guang Yang; Wanqing Wu; Shuo Li; Victor Hugo C. de Albuquerque
IEEE Network
Year: 2020 | Volume: 34, Issue: 2 | Magazine Article | Publisher: IEEE

Containerized Architecture for Edge Computing in Smart Home : A consistent architecture for model deployment
Nitu Gupta; Katpagavalli Anantharaj; Karthikeyan Subramani
2020 International Conference on Computer Communication and Informatics (ICCCI)
Year: 2020 | Conference Paper | Publisher: IEEE
Cited by: Papers (10)

Abstract HTML

Containerized Architecture for Edge Computing in Smart Home : A consistent architecture for model deployment

Nitu Gupta; Katpagavalli Anantharaj; Karthikeyan Subramani
2020 International Conference on Computer Communication and Informatics (ICCCI)
Year: 2020 | Conference Paper | Publisher: IEEE

Towards educational school cloud **for India: A case study with Kendriya Vidyalayas**
A. Paventhan; Hari Krishna; V. Gayathri; Neena Pahuja
2014 IEEE Region 10 Humanitarian Technology Conference (R10 HTC)
Year: 2014 | Conference Paper | Publisher: IEEE
Cited by: Papers (3)

Abstract HTML

Towards educational school cloud for India: A case study with Kendriya Vidyalayas

A. Paventhan; Hari Krishna; V. Gayathri; Neena Pahuja
2014 IEEE Region 10 Humanitarian Technology Conference (R10 HTC)
Year: 2014 | Conference Paper | Publisher: IEEE

Load More

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00370

IEEE Personal Account

CHANGE USERNAME/PASSWORD

Purchase Details

PAYMENT OPTIONS

VIEW PURCHASED DOCUMENTS

Profile Information

COMMUNICATIONS PREFERENCES

PROFESSION AND EDUCATION

TECHNICAL INTERESTS

Need Help?

US & CANADA: +1 800 678 4333

WORLDWIDE: +1 732 981 0060

CONTACT & SUPPORT

Follow

About IEEE *Xplore* | Contact Us | Help | Accessibility | Terms of Use | Nondiscrimination Policy | IEEE Ethics Reporting | Sitemap | IEEE Privacy Policy

A not-for-profit organization, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

© Copyright 2023 IEEE - All rights reserved.

IEEE Account

- Change Username/Password
- Update Address

Purchase Details

- Payment Options
- Order History
- View Purchased Documents

Profile Information

- Communications Preferences
- Profession and Education
- Technical Interests

Need Help?

- **US & Canada:** +1 800 678 4333
- **Worldwide:** +1 732 981 0060
- Contact & Support

About IEEE *Xplore* | Contact Us | Help | Accessibility | Terms of Use | Nondiscrimination Policy | Sitemap | Privacy & Opting Out of Cookies

A not-for-profit organization, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

© Copyright 2023 IEEE - All rights reserved. Use of this web site signifies your agreement to the terms and conditions.

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L1	3	17/671199	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/04 09:42 AM
L2	1	17/671056	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/28 04:24 PM
L3	16	((("LACEWORK") near3 ("INC"))).AS,AANM.	(USPAT)	OR	ON	ON	2023/01/28 04:25 PM
L4	43	((("ERLINGSSON") near3 ("LFAR")) OR (("CHEN") near3 ("YIJOU"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2023/01/28 04:26 PM
L5	0	(L3 L4) AND normal.clm.	(USPAT)	OR	ON	ON	2023/01/29 12:52 PM
L6	19	(L3 L4) AND normal	(USPAT)	OR	ON	ON	2023/01/29 12:53 PM
L7	33530	(H04L41/145 OR H04L67/10 OR H04L63/1408 OR H04L41/024 OR H04L41/0846 OR H04L41/0806 OR H04L41/0895).cpc.	(USPAT)	OR	ON	ON	2023/01/29 01:05 PM
L8	5	cloud near8 deploy$4 SAME normal near8	(USPAT)	OR	ON	ON	2023/01/29 01:05 PM

		behav$5					
L9	2	cloud near8 deploy$4 SAME behav$5 SAME recommend$4	(USPAT)	OR	ON	ON	2023/01/29 01:14 PM
L10	7	cloud SAME deploy$4 SAME behav$5 SAME recommend$4	(USPAT)	OR	ON	ON	2023/01/29 01:15 PM
L11	92	cloud SAME deploy$4 SAME behav$5 SAME recommend$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 01:15 PM
L12	329	cloud SAME deploy$4 SAME (normal normali$7) SAME recommend$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 01:40 PM
L13	166	cloud SAME behav$5 SAME (normal normali$7) SAME recommend$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 01:44 PM
L14	195379	H04L63/1425,1441,20,1	(US-PGPUB; USPAT;	OR	ON	ON	2023/01/29

		416.cpc.	USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				02:07 PM
L15	445658	L7 L14	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:08 PM
L16	411	compar$4 WITH trajectory WITH cloud	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:10 PM
L17	21	L16 AND L15	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL,	OR	ON	ON	2023/01/29 02:10 PM

			PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L18	54825	h04l67/535.cpc.	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:13 PM
L19	491952	L18 L15	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:13 PM
L20	21	L16 AND L19	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:13 PM
L21	3	L12 AND L20	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY,	OR	ON	ON	2023/01/29 02:14 PM

			CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L22	40	(Artificial Intelligence) More like doc: US-20220247769-A1 Text: (US-20220200869-A1 OR US-20220232025-A1 OR US-20220329616-A1 OR US-20220232024-A1 OR US-20220224707-A1 OR US-20220279004-A1 OR US-10425437-B1 OR US-20220400130-A1 OR US-20220400129-A1 OR US-10614071-B1 OR US-11470172-B1 OR US-11157502-B1 OR US-11134093-B1 OR US-11153339-B1 OR US-20220360600-A1 OR US-10498845-B1 OR US-20220311794-A1 OR US-20220215101-A1 OR US-20220294816-A1 OR US-20220303295-A1).did.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:16 PM
L23	3505	cloud WITH similar$7 WITH deploy$4	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT;	OR	ON	ON	2023/01/29 02:16 PM

			IBM_TDB)				
L24	772	L23 AND L19	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:16 PM
L25	68	cloud WITH similar$7 WITH deploy$4 WITH (behav$5 normal$7)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:17 PM
L26	21	L25 AND L19	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 02:17 PM
L27	3728	adopt$4 WITH cloud WITH deploy$5	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV,	OR	ON	ON	2023/01/29 08:43 PM

			MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L28	590	L27 AND L15	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 08:44 PM
L29	49	adopt$4 WITH cloud WITH deploy$5 WITH behav$5	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 08:44 PM
L30	12	L29 AND L15	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 08:44 PM
L31	3558	similar$4 WITH cloud	(US-PGPUB; USPAT;	OR	ON	ON	2023/01/29

		WITH deploy$5	USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				08:47 PM
L32	780	L31 AND L15	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 08:47 PM
L33	39	similar$4 WITH cloud WITH deploy$5 WITH adopt$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/29 08:49 PM
L34	75	(alike similar$4) WITH cloud WITH deploy$5 WITH learn$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL,	OR	ON	ON	2023/01/30 08:46 AM

			PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L35	26	(alike similar$4) WITH cloud WITH deploy$5 WITH learn$4 WITH (other another)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/30 11:17 AM
L36	1519	(good high$4 best) WITH (rating rate rank$4) WITH cloud WITH deploy$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/30 12:46 PM
L37	467	(good high$4 best) WITH (rating rate? rank$4) WITH cloud WITH deploy$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/01/30 12:46 PM
L38	5166	cloud SAME deploy$4 SAME compar$4 SAME (similar$4 "same"	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY,	OR	ON	ON	2023/02/02 01:22 PM

		common identical comparable related alike like analogous uniform resembl$4)	CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L39	826	L19 AND 37	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/02 01:23 PM
L40	826	L19 AND L38	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/02 01:23 PM
L41	8	L12 AND L40	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH,	OR	ON	ON	2023/02/02 01:24 PM

			TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L42	3	L16 AND L40	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/02 01:28 PM
L43	9570	cloud SAME deploy$4 SAME (compar$4 learn$4) SAME (similar$4 "same" common identical comparable related alike like analogous uniform resembl$4)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/02 01:31 PM
L44	1248	cloud SAME (normal normali$7) SAME recommend$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/02 01:32 PM
L45	324	L44 AND L43	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA,	OR	ON	ON	2023/02/02 01:32 PM

			EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L46	504	cloud SAME deploy$4 SAME (benchmark$4 standard) SAME recommend$6	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/02 01:39 PM
L47	372	L43 AND L46	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/02 01:42 PM
L48	19	L47 AND L19	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO;	OR	ON	ON	2023/02/02 01:44 PM

			JPO; DERWENT; IBM_TDB)				
L49	12	15/483398	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/03 11:41 AM
L50	1161	cloud SAME (typical$4) SAME recommend$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/03 11:49 AM
L51	73	L50 AND L19	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/03 11:49 AM
L52	88	cloud SAME (typical$4) SAME recommend$4 SAME deploy$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS,	OR	ON	ON	2023/02/03 11:51 AM

			IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L53	16	L52 AND L19	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/03 11:51 AM
L54	117	("10033611" OR "10115111" OR "10127273" OR "10142357" OR "10205736" OR "10339309" OR "10389742" OR "10581891" OR "10594718" OR "10708082" OR "10735329" OR "11153339" OR "20020184225" OR "20030037136" OR "20030179227" OR "20040015470" OR "20050102284" OR "20050188222" OR "20060085437" OR "20060259470" OR "20060288415" OR "20070050497" OR "20070118909" OR "20080148180" OR "20080151893" OR "20090019160" OR "20090228474" OR "20090271504" OR "20100094767" OR "20100114931" OR "20100172261" OR "20100274785" OR	(US-PGPUB; USPAT)	OR	ON	ON	2023/02/05 10:28 AM

		"20110055138" OR "20120005243" OR "20120317151" OR "20120323956" OR "20130205357" OR "20130304915" OR "20140115001" OR "20140115011" OR "20140208191" OR "20140245443" OR "20140280068" OR "20140359558" OR "20150135312" OR "20150188751" OR "20150341379" OR "20150356144" OR "20160078365" OR "20160080404" OR "20160110434" OR "20160149937" OR "20160205125" OR "20160218911" OR "20160359592" OR "20160359872" OR "20170118240" OR "20170134240" OR "20170147646" OR "20170163666" OR "20170251013" OR "20170272344" OR "20170279827" OR "20170288974" OR "20170359361" OR "20170366492" OR "20180020015" OR "20180034840" OR "20180063178" OR "20180103052" OR "20180115578" OR "20180137858" OR "20180173789" OR "20180174062" OR "20180181750" OR "20180248901" OR "20180268078" OR "20180288063" OR "20180329958" OR "20180336353" OR "20180357422" OR "20180367548" OR "20190028327" OR "20190109870" OR "20190132224" OR "20190222597" OR "20190259033" OR "20190339965" OR "20190349305" OR "20190378050" OR					

		"20200076685" OR "20200192690" OR "20200252376" OR "20200272740" OR "20200278892" OR "20200280592" OR "20200311644" OR "20200403860" OR "5584024" OR "5806062" OR "8103906" OR "8122122" OR "8443442" OR "8671453" OR "8862524" OR "9021583" OR "9037273" OR "9332020" OR "9515999" OR "9516053" OR "9558265" OR "9596253" OR "9654503" OR "9734040" OR "9749339" OR "9853968" OR "9953014").pn.					
L55	49	L54 AND (L18 L14)	(US-PGPUB; USPAT)	OR	ON	ON	2023/02/05 10:33 AM
L56	3741	cloud SAME (normal normali$7 typical$4 standard benchmark baseline) SAME recommend$4	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/05 10:36 AM
L57	1	L56 AND L54	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH,	OR	ON	ON	2023/02/05 10:37 AM

			TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L58	17	((US-11080392-B2 OR US-10951648-B2).did. AND USPT.dbnm.) OR ((CN-110999250-B OR CN-114598840-A OR WO-2021052379-A1).did. AND FTDB.dbnm.) OR ((US-20200259852-A1 OR US-20190068627-A1 OR US-20170251013-A1 OR US-20150319185-A1 OR US-20150172321-A1 OR US-20220417273-A1 OR US-20220092668-A1 OR US-20200218579-A1 OR US-20180124189-A1 OR US-20150302440-A1 OR US-20190123973-A1 OR US-20140229607-A1).did. AND PGPB.dbnm.)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/05 01:46 PM
L59	9	L58 AND threat	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/02/05 01:46 PM
L60	27	cloud WITH deploy$4 WITH (misconfigur$5 mis-configur$5)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH,	OR	ON	ON	2023/02/06 03:01 PM

			TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				

PE2E SEARCH - Search History (Interference)

There are no Interference searches to show.

(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 110999250 B
(45)授权公告日 2021.11.05

(21)申请号 201880052275.6
(22)申请日 2018.06.19
(65)同一申请的已公布的文献号
申请公布号 CN 110999250 A
(43)申请公布日 2020.04.10
(30)优先权数据
62/523,668 2017.06.22 US
16/011,538 2018.06.18 US
(85)PCT国际申请进入国家阶段日
2020.02.12
(86)PCT国际申请的申请数据
PCT/US2018/038347 2018.06.19
(87)PCT国际申请的公布数据
WO2018/236915 EN 2018.12.27
(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚
(72)发明人 G·基尔提 K·比斯瓦斯 M·S·N·佩雷拉
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
代理人 周衡威
(51)Int.Cl.
H04L 29/06(2006.01)
H04W 12/086(2021.01)
H04L 12/24(2006.01)
(56)对比文件
US 2015319185 A1,2015.11.05
US 2016350173 A1,2016.12.01
WO 2016177437 A1,2016.11.10
审查员 鲁秋艳

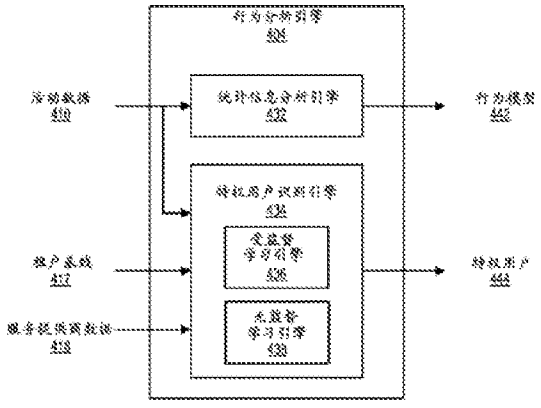
权利要求书3页 说明书53页 附图9页

(54)发明名称

在计算环境中监视特权用户和检测异常活动的方法、系统、介质

(57)摘要

在各种实施方式中，用于监视和管理云服务的安全性的安全性管理和控制系统能够包括用于识别给定云服务的特权用户的自动化技术。在各种示例中，安全性管理和控制系统能够从云服务获得活动日志，其中活动日志记录由组织的用户在使用云服务时执行的动作。在各种示例中，安全性管理和控制系统能够识别活动日志中相对于云服务具有特权的动作。在这些和其它示例中，安全性管理和控制系统能够使用活动日志中的这些动作来识别特权用户。一旦识别出特权用户，安全性管理和控制系统就能够以更高的审查程度监视特权用户。



CN 110999250 B

1.一种计算机实现的方法,包括:在安全性管理系统的计算机系统处:

从服务提供商系统获得活动数据,其中所述活动数据描述在使用云服务期间执行的动作,其中所述动作由与租户相关联的一个或多个用户执行,其中所述服务提供商系统向所述租户提供租户账户,并且其中所述租户账户使所述一个或多个用户能够访问所述云服务;

在所述活动数据中识别相对于所述云服务具有特权的一个或多个动作,作为一个或多个特权动作;

使用所述活动数据来识别执行了所述一个或多个动作的用户集,其中所述用户集是从与所述租户相关联的所述一个或多个用户中确定的;

将所述用户集识别为特权用户;

使用所述活动数据,确定所述一个或多个用户的一个或多个风险得分;

确定所述用户集中的用户的风险得分大于阈值;

确定用于所述服务提供商系统的安全性控制,其中所述安全性控制被所述服务提供商系统用于配置对所述云服务的访问;

确定要发送到所述服务提供商系统的一个或多个指令;以及

向所述服务提供商系统发送所述一个或多个指令,其中所述一个或多个指令使用于所述用户的所述安全性控制改变,其中所述用户对所述云服务的访问由于所述安全性控制的所述改变而被修改,使得由所述用户造成的安全性风险被监视、减轻和/或停止。

2.如权利要求1所述的计算机实现的方法,其中所述一个或多个动作是使用与所述云服务相关联的动作的列表来识别的,其中所述动作的列表中的动作被识别为针对所述云服务具有特权的特权动作。

3.如权利要求1所述的计算机实现的方法,其中,所述一个或多个动作是使用管理性动作的列表来识别的。

4.如权利要求1所述的计算机实现的方法,还包括:

使用所述一个或多个动作以及过去的活动数据来生成模型,所述模型描述相对于所述云服务具有特权的所述云服务的使用模式;以及

使用所述模型来识别所述用户集。

5.如权利要求1所述的计算机实现的方法,还包括:

对在所述云服务的使用期间执行的所述动作进行分组;以及

识别包括具有特权的动作的一组动作,其中所述用户集是使用所述一组动作来识别的。

6.如权利要求1所述的计算机实现的方法,其中风险得分指示用户在使用所述云服务时所执行的动作对所述租户的安全性风险的程度。

7.如权利要求1所述的计算机实现的方法,其中将风险得分计算为风险指标的加权和。

8.如权利要求1所述的计算机实现的方法,其中,与非特权用户的风险得分相比,以更大的权重计算被识别为特权用户的用户的风险得分。

9.如权利要求1所述的计算机实现的方法,其中特权动作是在由第一用户执行时能够以影响其他用户对所述云服务的使用的方式来修改所述云服务的动作。

10.如权利要求1所述的计算机实现的方法,其中特权动作是当由第一用户执行时能够

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00391

影响所述云服务的其他用户的用户账户的动作。

11.一种安全性管理系统的计算系统，包括：

一个或多个处理器；以及

存储器，耦合到所述一个或多个处理器并能够由所述一个或多个处理器读取，所述存储器包括指令，所述指令在由所述一个或多个处理器执行时使所述一个或多个处理器执行以下操作，包括：

从服务提供商系统获得活动数据，其中所述活动数据描述在使用云服务期间执行的动作，其中所述动作由与租户相关联的一个或多个用户执行，其中所述服务提供商系统向所述租户提供租户账户，并且其中所述租户账户使所述一个或多个用户能够访问所述云服务；

在所述活动数据中识别相对于所述云服务具有特权的一个或多个动作，作为一个或多个特权动作；

使用所述活动数据来识别执行了所述一个或多个动作的用户集，其中所述用户集是从与所述租户相关联的所述一个或多个用户中确定的；

将所述用户集识别为特权用户；

使用所述活动数据，确定所述一个或多个用户的一个或多个风险得分；

确定所述用户集中的用户的风险得分大于阈值；

确定用于所述服务提供商系统的安全性控制，其中所述安全性控制被所述服务提供商系统用于配置对所述云服务的访问；

确定要发送到所述服务提供商系统的一个或多个指令；以及

向所述服务提供商系统发送所述一个或多个指令，其中所述一个或多个指令使用于所述用户的所述安全性控制改变，其中所述用户对所述云服务的访问由于所述安全性控制的所述改变而被修改，使得由所述用户造成的安全性风险被监视、减轻和/或停止。

12.如权利要求11所述的计算系统，其中所述一个或多个动作是使用与所述云服务相关联的动作的列表来识别的，其中所述动作的列表被识别为针对所述云服务具有特权的特权动作。

13.如权利要求11所述的计算系统，其中所述一个或多个动作是使用管理性动作的列表来识别的。

14.如权利要求11所述的计算系统，所述存储器还包括在由所述一个或多个处理器执行时使所述一个或多个处理器执行以下操作的指令：

使用所述一个或多个动作以及过去的活动数据来生成模型，所述模型描述相对于所述云服务具有特权的所述云服务的使用模式；以及

使用所述模型来识别所述用户集。

15.如权利要求11所述的计算系统，所述存储器还包括在由所述一个或多个处理器执行时使所述一个或多个处理器执行以下操作的指令：

对在所述云服务的使用期间执行的所述动作进行分组；以及

识别包括具有特权的动作的一组动作，其中所述用户集是使用所述一组动作来识别的。

16.一种其上存储有指令的非暂态计算机可读介质，所述指令在由安全性管理系统的

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00392

计算系统的一个或多个处理器执行时使所述一个或多个处理器执行以下操作：

从服务提供商系统获得活动数据，其中所述活动数据描述在使用云服务期间执行的动作，其中所述动作由与租户相关联的一个或多个用户执行，其中所述服务提供商系统向所述租户提供租户账户，并且其中所述租户账户使所述一个或多个用户能够访问所述云服务；

在所述活动数据中识别相对于所述云服务具有特权的一个或多个动作，作为一个或多个特权动作；

使用所述活动数据来识别执行了所述一个或多个动作的用户集，其中所述用户集是从与所述租户相关联的所述一个或多个用户中确定的；

将所述用户集识别为特权用户；

使用所述活动数据，确定所述一个或多个用户的一个或多个风险得分；

确定所述用户集中的用户的风险得分大于阈值；

确定用于所述服务提供商系统的安全性控制，其中所述安全性控制被所述服务提供商系统用于配置对所述云服务的访问；

确定要发送到所述服务提供商系统的一个或多个指令；以及

向所述服务提供商系统发送所述一个或多个指令，其中所述一个或多个指令使用于所述用户的所述安全性控制改变，其中所述用户对所述云服务的访问由于所述安全性控制的所述改变而被修改，使得由所述用户造成的安全性风险被监视、减轻和/或停止。

17.如权利要求16所述的非暂态计算机可读介质，其中所述一个或多个动作是使用与所述云服务相关联的动作的列表来识别的，其中所述动作的列表被识别为相对于所述云服务具有特权的特权动作。

18.如权利要求16所述的非暂态计算机可读介质，其中，所述一个或多个动作是使用管理性动作的列表来识别的。

19.如权利要求16所述的非暂态计算机可读介质，还包括在由所述一个或多个处理器执行时使所述一个或多个处理器执行以下操作的指令：

使用所述一个或多个动作以及过去的活动数据来生成模型，所述模型描述相对于所述云服务具有特权的所述云服务的使用模式；以及

使用所述模型来识别所述用户集。

20.如权利要求16所述的非暂态计算机可读介质，所述存储器还包括在由所述一个或多个处理器执行时使所述一个或多个处理器执行以下操作的指令：

对在所述云服务的使用期间执行的所述动作进行分组；以及

识别包括具有特权的动作的一组动作，其中所述用户集是使用所述一组动作来识别的。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00393

在计算环境中监视特权用户和检测异常活动的方法、系统、介质

[0001] 相关申请的交叉引用

[0002] 本申请要求于2017年6月22日提交的美国临时申请No.62/523,668和2018年6月18日提交的美国申请No.16/011,538的优先权，其全部内容通过引用整体并入本文。

背景技术

[0003] 云服务提供商在“云”中提供各种服务，即，通过诸如公用互联网之类的网络，并且任何网络连接的客户端设备均可远程访问。由云服务提供商(本文中也称为“云提供商”或“提供商”)使用的服务模型的示例包括基础设施即服务(IaaS)、平台即服务(PaaS)、软件即服务(SaaS)和网络即服务(NaaS)。IaaS提供商为客户提供基础设施资源，诸如客户能够用来运行软件的处理、存储、网络和其它计算资源。客户不管理基础设施，但尤其可以控制操作系统、存储装置和已部署的应用，并且可以能够控制一些网络部件(诸如防火墙)。PaaS提供商为客户提供了客户可以在其上开发、运行和管理应用的平台，而无需维护底层的计算基础设施。SaaS是软件许可和交付模型，其中软件是按订阅方式向客户许可的，并且由云提供商集中托管。在这种模型下，可以例如使用web浏览器访问应用。NaaS提供商通过例如在另一方运营的网络基础设施上供应(provision)虚拟网络来向客户提供网络服务。在这些服务模型中的每一种中，云服务提供商都维护并管理提供服务的硬件和/或软件，而很少有(如果有的话)软件在用户的设备上执行。

[0004] 云服务提供商的客户(可以被称为用户或租户)可以向服务提供商订阅以获得对由服务提供商提供的特定服务的访问。服务提供商可以为用户或租户维护账户，用户和/或租户可以通过该账户来访问提供商的服务。服务提供商还可以为各个用户维护与租户相关联的用户账户。服务提供商的示例包括Box、Dropbox、Microsoft、Docusign、Google、Salesforce、Oracle、Amazon等。诸如这些之类的服务提供商可以提供多种不同的服务，但是不同的服务提供商不需要彼此具有任何从属关系，包括不共享基础设施或安全边界。服务提供商系统常常也受到高度保护并且对非租户是关闭的。

[0005] 组织对计算环境的依赖已经导致云服务被广泛地用于诸如协作、销售和客户服务、基础设施等之类的操作。通过云环境提供的应用可以使组织能够更快地推出服务，而无需在数据中心、硬件、软件和部署项目上进行大量前期投资。由于可从许多地点(例如，在工作地点、家中、在宾馆和其它地点)使用支持云的服务，因此应用的可访问性可以提高员工的生产率。

[0006] 因为组织和/或组织的用户可以订阅许多不同的云服务提供商的服务，所以组织可能需要确保组织自己的系统不会因使用云服务而受到损害的方式。使用云服务会导致在组织在组织自身内部托管和管理服务时不存在的安全性风险。

发明内容

[0007] 在各种实施方式中，提供了用于云安全性系统的系统和方法，其可以识别相对于

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00394

由云服务提供商提供的应用或服务具有特权能力的用户。具有特权功能的用户(可以被称为管理性用户或特权用户)可以具有访问或修改云服务的能力,这可以改变云服务操作的方式和/或其他用户使用云服务的方式。因为特权用户相对于普通用户在云服务方面具有更多的能力,所以可能期望以更高的审查程度监视特权用户,并快速确定特权用户账户是否已受到损害。

[0008] 但是,对于云服务,知道哪些用户是特权用户和哪些用户是普通用户可能并不容易。对于什么使用户具有特权,不同的云服务可以具有不同的参数。此外,云服务可能无法为客户提供确定哪些用户账户具有特权的方式。

[0009] 提供了使安全性管理系统能够识别云服务的特权用户的系统、方法和计算机可读介质。在各种实施方式中,安全性管理系统可以包括用于识别云服务的特权用户的技术,其中该技术包括执行各种步骤。步骤可以包括从服务提供商系统获得活动数据。活动数据可以描述在使用云服务期间执行的动作。动作可以由与租户相关联的一个或多个用户执行,其中服务提供商系统向租户提供租户账户。租户账户使一个或多个用户能够访问云服务。步骤还可以包括在活动数据中识别相对于云服务具有特权的一个或多个动作。步骤还可以包括使用活动数据来识别执行一个或多个动作的用户集。可以从与租户相关联的一个或多个用户中确定该用户集。步骤还可以包括将用户集分类为具有特权。步骤还可以包括使用活动数据来确定一个或多个用户的一个或多个风险得分。步骤还可以包括确定用户集中的用户的风险得分大于阈值。步骤还可以包括确定用于服务提供商系统的安全性控制,其中该安全性控制被服务提供商系统用于配置对云服务的访问。步骤还可以包括确定一个或多个指令以发送到服务提供商系统。步骤还可以包括将一个或多个指令发送到服务提供商系统。一个或多个指令可以使安全性控制相对于用户改变,其中用户对云服务的访问由于对安全性控制的改变而被修改。

[0010] 在各个方面,使用与云服务相关联的动作的列表来识别该一个或多个动作,其中动作的列表中的动作相对于云服务被分类为具有特权。在各个方面,使用管理性动作列表来识别该一个或多个动作。

[0011] 在各个方面,由上述系统、方法和计算机可读介质实现的技术还包括步骤,诸如使用一个或多个动作和过去的活动数据来生成模型,其中该模型描述云服务的相对于云服务具有特权的使用模式。步骤还可以包括使用模型来识别该用户集。

[0012] 在各个方面,由上述系统、方法和计算机可读介质实现的技术还包括步骤,诸如对在云服务的使用期间执行的动作进行分组。步骤还可以包括识别包括具有特权的动作的一组动作,其中用户集是使用该组动作来识别的。

[0013] 在各个方面,风险得分指示用户在使用云服务时所执行的动作对租户的安全性风险的程度。在各个方面,风险得分被计算为风险指标(indicator)的加权和。在各种示例中,被分类为具有特权的用户的风险得分以比非特权用户的风险得分的权重更大的权重进行计算。

[0014] 在各个方面,特权动作是在由第一用户执行时可以以影响其他用户对云服务的使用的方式来修改云服务的动作。在各个方面,特权动作是当由第一用户执行时可以影响云服务的其他用户的用户账户的动作。

[0015] 通过参考以下说明书、权利要求书和附图,前述内容以及其它特征和实施方式将

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00395

变得更加清楚。

附图说明

[0016] 图1包括框图，该框图图示包括安全性监视和控制系统的计算环境的示例。

[0017] 图2图示了可以由安全性管理和控制系统实现的示例云安全性系统的框图。

[0018] 图3图示了安全性管理和控制系统的示例分析引擎的框图。

[0019] 图4图示了行为分析引擎的框图。

[0020] 图5图示了表6中的数据的图的示例。

[0021] 图6包括流程图，该流程图图示了用于确定云服务的特权用户并管理特权用户的活动可能造成的安全性风险的处理的示例。

[0022] 图7描绘了其中可以实现上述各种示例的分布式系统的简化图。

[0023] 图8是在其中可以将服务作为云提供的系统环境的一个或多个部件的简化框图。

[0024] 图9图示了可以被用于实现各种示例的计算机系统的示例。

具体实施方式

[0025] 在下面的描述中，出于解释的目的，阐述了具体细节以便提供对各种实施方式和示例的透彻理解。但是，将清楚的是，可以在没有这些具体细节的情况下实践各种实施方式。例如，电路、系统、算法、结构、技术、网络、处理和其它部件可以被示为框图形式的部件，以便不以不必要的细节使实施方式模糊。附图和描述不意图是限制性的。

[0026] 一些示例(诸如关于本公开中的附图所公开的那些)可以被描述为被绘制为流程图、流图、数据流图、结构图、顺序图或框图的处理。虽然顺序图或流程图可以将操作描述为顺序处理，但是许多操作可以并行或并发地执行。此外，操作的次序可以被重新安排。处理在其操作完成时终止，但是可以具有图中不包括的附加步骤。处理可以与方法、函数、过程、子例程、子程序等对应。当处理与函数对应时，其终止可以与函数返回到调用函数或主函数对应。

[0027] 本文描绘的处理(诸如参考本公开中的附图描述的那些)可以用由一个或多个处理单元(例如，处理器核)、硬件或其组合执行的软件(例如，代码、指令、程序)来实现。软件可以存储在存储器中(例如，存储在存储设备上、存储在非瞬态计算机可读存储介质上)。在一些示例中，本文中以序列图和流程图描绘的处理可以通过本文公开的任何系统来实现。本公开中的处理步骤的特定系列并不旨在限制。步骤的其它顺序也可以根据替代示例执行。例如，本公开的替代示例可以以不同的次序执行上面概述的步骤。而且，附图中所示的各个步骤可以包括多个子步骤，这些子步骤可以以对个体步骤适当的各种顺序执行。此外，取决于特定的应用，可以添加或移除附加的步骤。本领域的普通技术人员将认识到许多变化、修改和替代。

[0028] 在一些示例中，本公开的图中的每个处理可以由一个或多个处理单元执行。处理单元可以包括一个或多个处理器，包括单核或多核处理器、处理器的一个或多个核，或其组合。在一些示例中，处理单元可以包括一个或多个专用协处理器，诸如图形处理器、数字信号处理器(DSP)等。在一些示例中，可以使用定制电路来实现一些或所有处理单元，诸如专用集成电路(ASIC)或现场可编程门阵列(FPGA)。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00396

[0029] 在各种示例中，云服务可以定义将服务的管理性用户与服务的普通用户区分开的特权。在这个上下文中，特权是指对诸如网络、应用或服务等的计算资源进行改变的能力，和/或影响受限制以便不被非特权用户访问的资源的能力。普通用户可以具有最少的特权集，而管理性用户具有更高的特权，使得管理性用户可以对管理性用户具有更高特权的计算资源进行配置和管理。管理性用户在本文中也被称为特权用户。

[0030] 云服务可以以特定于该云服务的各种方式细分和分配特权。例如，文件共享服务Box具有访问控制模型，该模型定义管理性用户、共同管理性用户和组管理性用户。在这个示例中，管理性用户可以管理用户和组、查看和编辑组织的所有文件和文件夹、登录到组织中的任何用户的用户账户、编辑组织的设置，以及运行或访问报告。共同管理性用户可以执行与管理性用户相同的操作，但是不能改变管理性用户的权限。组管理性用户可以管理组织内的组，并且可以将用户添加到这些用户所管理的组，可以创建将被指派给组的新用户，并且可以将访问权指派给组，但是在该组之外没有能力。

[0031] 因为管理性用户具有比普通用户更高的特权，所以与普通用户的活动相比，可能需要对管理性用户的活动给予更高的审查程度。出于各种原因，可能需要进行更高的审查程度。例如，合规审核员可以请求针对管理性活动的审核踪迹(trail)，以确保云服务符合各种财务安全法律。作为另一个示例，对流氓管理员的识别可以防止滥用或篡改组织的机密数据。作为另一个示例，网络攻击者经常将管理性用户账户作为目标，网络攻击者寻求获得管理性用户账户可用的更高特权。

[0032] 但是，识别特定服务的特权用户可能不容易。对于在企业网络内执行的企业应用，组织有时可以将这些应用与组织的内部公司身份管理系统集成在一起。这样做可以使组织能够管理企业应用的用户以及指派给这些用户的特权。但是，对于云服务，对用户和用户的特权的管理是在企业网络外部的云服务内被处置的。此外，每个云服务可以使用不同的模型来向云服务的用户授予特权，使得使用户成为管理性用户的特权对于每个云服务可以是不同的。因此，可能没有统一的定义来识别云服务的特权用户。

[0033] 而且，组织的用户可以使用不受管理的设备(例如，与该组织拥有的设备相对的，个人拥有的设备)来使用云服务，和/或可以从不受管理的网络(例如，家里、机场、宾馆或其它在企业网络外部的互联网连接)来访问云服务。在这些情况下，用户对云服务的使用会造成组织的网络的安全漏洞，从而使组织容易受到网络攻击。

[0034] 来自各个供应商(诸如Imperva、Igrhythem、Cyberark和Oracle)的产品可以执行特权用户的监视。但是，这些产品要求特权用户事先被识别并且被录入该产品以进行监视。云服务的特权用户可能是事先不知道的，因为对该信息的跟踪可能不是无瑕疵的。此外，管理性用户还可以随着时间而改变，新的管理性用户由组织的用户、由自动化工具、通过API和/或由恶意用户创建。同样经常发生的是，创建了并使用管理性账户的人可会停止被组织雇佣，使得关于哪些账户是管理性账户的知识随着时间而丢失。

[0035] 在各种实施方式中，用于监视和管理云服务的安全性的安全性管理和控制系统可以包括用于识别给定云服务的特权用户的自动化技术。在各种示例中，安全性管理和控制系统可以从云服务获得日志文件，在本文中被称为活动日志，其中活动日志记录由组织的用户在使用云服务时执行的动作。在各种示例中，安全性管理和控制系统可以识别活动日志中相对于云服务具有特权的动作。例如，安全性管理和控制系统可以使用受监督学习技

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00397

术和无监督学习技术来开发模型，该模型描述组织的用户使用云服务的方式。在这个和其它示例中，安全性管理和控制系统可以使用活动日志中的操作来识别特权用户。一旦识别出特权用户，安全性管理和控制系统就可以以更高的审查程度来监视这些特权用户。

[0036] 图1包括图示计算环境100的示例的框图，该计算环境100包括安全性监视和控制系统102(在本文中也称为“安全性管理系统”和“安全性系统”)。安全性监视和控制系统102可以为使用由云服务提供商110提供的服务112a-112b的组织130提供威胁分析和补救。因为服务112a-112b是在组织130的网络的外部被管理的，所以组织130的网络安全性系统可能无法确保对服务112a-112b的使用不会对组织130产生安全性风险。在各种示例中，组织130的用户可以通过网络150或网络的组合来访问和使用服务112a-112b。网络150可以包括例如公共互联网。组织130可以类似地通过网络150来访问和使用安全性监视和控制系统102的服务。在各种示例中，安全性监视和控制系统102也可以被描述为由云服务提供商管理的云服务。安全性监视和控制系统的示例是Oracle CASB，它是由Oracle公司提供的产品和服务。

[0037] 服务提供商是被配置为向他人提供计算服务的硬件和软件的集合。计算服务可以包括例如硬件资源、处理资源、数据存储装置、硬件平台、软件平台和/或各种复杂级别的应用等计算资源。在一些情况下，服务提供商操作启用提供商的服务的硬件和软件。在一些情况下，服务提供商使用另一个服务提供商的硬件和软件。例如，第一服务提供商可以是数据中心的运营商，其将软件托管服务资源出租给第二服务提供商，第二服务提供商然后向组织的用户提供诸如协作文字处理应用之类的应用。

[0038] 在各种示例中，服务提供商遵循订阅模型，并将服务的使用出租给他人，其中出租可以持续一段时间。可以向订户(在本文中也称为租户)给予针对服务提供商和/或特定服务的账户，租户通过该账户使用服务。当租户是组织时，租户可以具有针对服务提供商的账户(在本文中称为租户账户或企业账户)并且组织的用户可以具有针对服务提供商和/或特定服务的个人用户账户。在一些情况下，用户账户与租户的账户绑定，并且租户账户可以控制和管理用户账户。

[0039] 在一些示例中，服务提供商可以提供服务而无需来自订户的直接报酬。例如，服务提供商可以提供免费的电子邮件应用，并通过其它渠道(诸如广告)获得服务的收入。在这个和其它示例中，订户可以在没有租约的情况下并且可能无限期地获得针对服务提供商的账户。

[0040] 在图1的示例中，服务提供商110提供两个服务112a-112b，它们可以通过网络150让服务提供商110的客户访问。服务112a-112b可以包括例如基础设施、平台、网络和应用以及其它服务。在一些示例中，两个服务112a-112b可以是相互关联的，诸如能够共享数据或可通过单一登录机制访问(例如，用户凭证的一个集合使用户能够访问两个服务112a-112b，而无需分别对每个服务进行认证)。在一些示例中，服务112a-112b完全独立地操作。例如，第一服务112a可以是银行应用，而第二服务112b可以是软件开发平台。在这个示例中，服务提供商110可以将两个服务112a-112b作为分离且不相关的产品提供。

[0041] 服务提供商110的客户可以包括个人用户和/或组织，其中组织可以包括多个用户。组织是将人和资源聚集在一起以实现一个或多个共同目的的实体。组织的示例包括公司、大学、公用事业和政府实体，等等。组织的人可以在组织中被表示为用户。在计算环境

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00398

100的上下文中，用户是数字实体，其可以包括用户名、用户标识符、用户账户、凭证(例如，密码、安全令牌或另一种形式的凭证)和/或可以被用于访问和使用组织的计算系统的其它数据。一个或多个人可以与同一用户账户关联，或者一个人可以具有多个用户账户。例如，网络管理团队的成员可以使用与网络管理团队的其他成员所使用的相同的管理性用户账户，同时还具有仅由该网络管理员使用的用户账户。在一些示例中，代替由人类驱动或作为其补充，用户的活动可以由自动化程序(诸如“机器人”)驱动。

[0042] 在各种示例中，组织的用户和组织的资源处于共同的管理之下，并且可以在相同的安全范围内操作。例如，在图1的示例中，组织130的资源包括企业网络104和多个客户端设备106a-106c。客户端设备106a-106c可以包括例如台式计算机、膝上型计算机、智能电话、平板电脑和其它计算设备。在一些示例中，客户端设备106a-106c可以由组织130的员工个人拥有，但是当这些设备连接到企业网络104时，这些设备由组织130管理。企业网络104还可以包括其它计算设备，诸如服务器、打印机、路由器、交换机和其它网络设备。组织130的资源还可以包括数据(例如，文档、网页、业务数据、用户数据等)、数据库、应用、处理能力、存储能力、联网能力以及其它硬件、软件或数字资源。

[0043] 在各种示例中，组织130的客户端设备106a-106b可以连接到企业网络104并在其中操作。例如，客户端设备106a-106b可以连接到企业网络104内的交换机，其将客户端设备106a-106b放置在企业网络104的防火墙108后面。通过位于防火墙108后面，将客户端设备106a-106b置于企业网络104的安全范围内。在安全范围内，防火墙108和其它网络安全性系统(诸如安全性信息和事件管理(SIEM)应用、入侵检测系统(IDS)和入侵防御系统(IPS)等等)可以保护企业网络104中的设备免受网络威胁。

[0044] 在各种示例中，组织130的客户端设备106c可以从企业网络104的外部连接到组织130。例如，客户端设备106c可以通过互联网服务提供商(ISP)连接到网络150，并且通过网络150，客户端设备106c可以能够连接到企业网络104的虚拟专用网(VPN)，或类似的机制。一旦连接到VPN，客户端设备106c就可以作为企业网络104的一部分进行操作，并受到企业网络104的安全范围的保护。但是，在这个示例中，客户端设备106c同时连接到客户端设备106c与企业网络104之间的网络150，因此仍然会暴露于可能来自网络150的安全性风险。此外，在一些示例中，客户端设备106c可能不需要连接到企业网络104以便能够访问服务提供商110的服务112a-112b。

[0045] 在各种示例中，组织130的用户可以通过客户端设备106a-106c利用组织130的资源以及组织130订阅的服务112a-112b，客户端设备106a-106c可以是组织130拥有的并由组织130提供给组织的成员。在各种示例中，成员可以使用组织130的任何客户端设备，并且可以能够使用多个客户端设备。例如，企业网络104可以包括用户工作站，任何成员都可以使用该用户工作站登录并访问该成员的用户账户。作为另一个示例，成员可以访问膝上型计算机和智能电话，并且可以登录到任一设备以访问同一个用户账户。可替代地或附加地，成员可以使用成员个人拥有的客户端设备来连接到组织的网络，以利用组织130的资源。

[0046] 如上所述，服务提供商的服务112a-112b的订户也可以是不隶属于组织130或不作为组织130的一部分的个人。个人可以访问启用网络的客户端设备106d，个人可以通过该设备访问服务112a-112b。个人可以具有针对ISP的用户账户，该用户账户使个人能够访问网络150。可替代地或附加地，个人可以具有针对服务112a-112b中的一个或多个的用户账户。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00399

但是，个人不能使用客户端设备106d连接到企业网络104，因为该个人没有针对组织130的用户账户或使该用户账户获得连接到企业网络104的权限的凭证。

[0047] 在各种示例中，个人和组织可以订阅由不同服务提供商提供的服务。例如，组织可以使用来自一个服务提供商的电子邮件服务(例如，来自Google®的Gmail)和来自另一个服务提供商的文件共享服务(例如，Dropbox)。在这个和其它示例中，不同的服务提供商可以是不相关的，包括具有用于支持其相应服务并由不同实体控制的分开的计算系统。在一些示例中，用户可以具有针对每个服务提供商和/或每个服务提供商的服务的分开的账户。在一些示例中，用户可以能够使用公共用户账户来访问不同服务提供商的服务。

[0048] 在一些示例中，云服务可以被授权或不被授权在组织130中使用。经授权的服务是组织130已经批准使用的服务。批准可以包括例如通过认证处理对服务进行审查(vet)以确保服务是安全的、与服务提供商110建立服务合同、将服务提供商110置于经批准的服务提供商列表中、将服务提供商110识别为知名且受信任的服务提供商、和/或控制为组织130的用户生成针对服务的用户账户等活动。例如，服务提供商110可以被服务提供商110分类为“受信任的”服务提供商。在一些示例中，组织130可以将其它服务提供商分类为“不信任的”，或者将不在受信任的列表上的所有服务提供商分类为不信任的。未经授权的服务是组织还没有特别批准并且由用户自行决定使用的服务。例如，用户可以正在使用组织130还没有特别授权的文件共享服务，可能组织130没有意识到文件共享服务正在被使用。

[0049] 在一些示例中，服务提供商110的服务112a-112b可以通过可以从组织130内执行并且可以被授权在组织130内使用的应用来执行或访问。例如，组织130可以具有经授权的web浏览器应用，用户可以通过该应用来访问诸如文件共享服务或数据库服务之类的服务。在这个和其它示例中，web浏览器应用可以被称为内部应用。在一些示例中，内部应用可以与云服务112a-112b协作地操作，包括例如允许服务112a-112b访问组织130内的数据、用户账户信息或其它信息。因为内部应用正在组织130内执行(例如，在组织130的客户端设备106a-106c上)，所以组织130可以监视和控制内部应用的使用情况。但是，组织130可能不知道或不能够监视用户通过内部应用对服务提供商110的服务112a-112b的使用情况。

[0050] 在一些示例中，用户可以通过第三方服务提供商114来访问服务提供商110的服务112a-112b。例如，用户可以首先访问由第三方服务提供商114提供的服务116，并且通过这个服务116，访问另一个服务提供商110的服务112b(在这里用虚线箭头示出)。第三方服务提供商114的服务116可以是例如使用户能够定位和访问其它云服务提供商的应用和服务的门户服务。在一些示例中，第三方服务提供商的服务116通过网络150提供对其它服务的访问(例如，去往和来自服务116的针对另一个服务112b的数据通过网络150传输到另一个服务112b和从另一个服务112b传输)，但是从用户的角度来看，访问可以看起来是直接的。在一些示例中，服务116使用户能够建立针对另一个服务112b的订阅，此后，用户直接访问另一个服务112b，而无需访问第三方服务提供商114的服务116。

[0051] 云服务112a-112b的使用可以对服务112a-112b的订户产生安全性风险。例如，在大多数情况下，在组织内操作的硬件、平台和软件可以由组织控制并由组织保护，例如使用物理屏障和/或网络安全工具。但是，云服务112a-112b在组织130之外操作并且在组织130的直接控制之外。组织130可能对用户在使用服务112a-112b时执行的活动具有几乎零可见性，或者不能控制用户执行的动作。此外，组织130可能几乎没有监视或控制以下用户动作

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00400

的能力，即，这些用户动作使得可疑数据或未经授权的用户通过服务112a-112b进入组织130，或使得组织的数据通过服务112a-112b的使用而离开组织130。

[0052] 在各种实施方式中，安全性监视和控制系统102可以为云服务112a-112b的订户提供网络威胁检测和补救服务。在各种实施方式中，安全性监视和控制系统102可以分析服务112a-112b的使用并识别可能对组织或个人订户构成威胁的活动。在一些实施方式中，安全性监视和控制系统102还可以建议补救动作，和/或可以自动执行补救动作以隔离或停止威胁。在一些示例中，由安全性监视和控制系统102执行的分析可以包括确定用户活动中的正常和/或异常行为的模型，并使用这些模型来检测可疑活动的模式。在一些示例中，安全性监视和控制系统102可以同时分析来自不同服务和/或来自不同服务提供商的数据。在这些示例中，安全性监视和控制系统102可以能够检测仅在使用不同服务执行的动作发生时才明显的可疑活动。在各种示例中，当分析确定可能需要在其它服务提供商处采取动作时，安全性监视和控制系统102可以确定可以在检测到可疑活动的服务提供商处或在不同服务提供商处执行的补救措施。

[0053] 在一些示例中，安全性管理和控制系统102可以被集成到组织130的计算环境中。例如，安全性监视和控制系统102可以在企业网络104中的服务器上并且在组织130的防火墙108后执行。在这些示例中，安全性管理和控制系统102可以由组织的网络管理员和/或由与安全性管理和控制系统102的开发人员相关联的人员来管理。

[0054] 可替代地或附加地，在各种示例中，可以将安全性监视和控制系统102的能力作为服务提供给个人和组织。例如，可以将安全性监视系统102的网络安全服务作为基于web的云服务和/或在软件即服务(SaaS)模型下提供。在这些和其它示例中，客户可以使用由安全性监视和控制系统102提供的应用，其中这些应用提供各种威胁检测和补救功能。与服务提供商110的服务112a-112b一样，个人和组织可以订阅由安全性监视和控制系统102提供的安全服务。在一些示例中，组织130的组用户可以被指定为管理性用户，并且可以管理由安全性监视和控制系统102在监视组织130的安全性时所进行的操作。这些用户可以能够访问信息(诸如由安全性管理和控制系统102生成的报告)，并且具有执行安全性管理和控制系统102建议的补救动作的能力，等等。

[0055] 在各种实施方式中，可以使用计算系统来实现安全性监视和控制系统102。在这些实施方式中，计算系统可以包括一个或多个计算机和/或服务器(例如，一个或多个访问管理器服务器)，其可以是通用计算机、专用服务器计算机(诸如台式服务器、UNIX服务器、中型服务器、大型机、机架式服务器等)、服务器场、服务器集群、分布式服务器或任何其它适当布置和/或计算硬件的组合。安全性监视和控制系统102可以运行操作系统和/或各种附加服务器应用和/或中间层应用，包括超文本传输协议(HTTP)服务器、文件传输服务(FTP)服务器、通用网关接口(CGI)服务器、Java服务器、数据库服务器和其它计算系统。数据库服务器的示例包括可从Oracle、Microsoft等获得的数据库服务器。可以使用硬件、固件、软件或硬件、固件和软件的组合来实现安全性监视和控制系统102。

[0056] 在各种实施方式中，安全性监视和控制系统102可以包括至少一个存储器、一个或多个处理单元(例如，(一个或多个)处理器)和/或存储装置。(一个或多个)处理单元可以适当地以硬件(例如，集成电路)、计算机可执行指令、固件或硬件和指令的组合来实现。在一些示例中，安全性监视和控制系统102可以包括若干子系统和/或模块。安全性监视和控制

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00401

系统102中的子系统和/或模块可以以硬件、在硬件上执行的软件(例如,可由处理器执行的程序代码或指令)或其组合来实现。在一些示例中,软件可以存储在存储器(例如,非暂态计算机可读介质)中、存储器设备上或某种其它物理存储器中,并且可以由一个或多个处理单元(例如,一个或多个处理器、一个或多个处理器核心、一个或多个图形处理单元(GPU)等)执行。(一个或多个)处理单元的计算机可执行指令或固件实施方式可以包括以任何合适编程语言编写的可以执行本文所述的各种操作、功能、方法和/或处理的计算机可执行指令或机器可执行指令。存储器可以存储在(一个或多个)处理单元上可加载和可执行的程序指令,以及在这些程序的执行期间生成的数据。存储器可以是易失性的(诸如随机存取存储器(RAM))和/或非易失性的(诸如只读存储器(ROM)、闪存等)。可以使用任何类型的持久性存储设备(诸如计算机可读存储介质)来实现该存储器。在一些示例中,计算机可读存储介质可以被配置为保护计算机免受包含恶意代码的电子通信的影响。计算机可读存储介质可以包括存储在其上的指令,该指令在处理器上执行时执行安全性监视和控制系统102的操作。

[0057] 在各种实施方式中,安全性监视和控制系统102可以包括实现安全性监视和控制系统102的不同特征的各种模块。在图1的示例中,这些模块包括扫描器174、模式分析器176、学习系统178、数据访问器182、数据分析系统136、信息处置器系统138、映射生成器170、控制管理器172、日志收集器系统134和接口120。安全性监视和控制系统102还包括存储装置122,其存储安全性监视和控制系统102使用的各种数据。在一些示例中,安全性监视和控制系统102还可以连接到附加的数据存储库180,附加数据存储库180可以存储安全性监视和控制系统102可能需要的附加数据。

[0058] 在各种示例中,安全性监视和控制系统102的存储装置122可以存储一个或多个数据存储库,这些存储库存储租户配置信息124、安全性信息126、域信息128和应用信息132。在各种示例中,存储装置122可以包括一个或多个数据库(例如,文档数据库、关系数据库或其它类型的数据库)、一个或多个文件存储库、一个或多个文件系统,或用于存储数据的系统的组合。

[0059] 在各种示例中,租户配置信息124(“租户配置信息”)可以包括租户和租户账户以及与每个租户账户相关联的用户账户的配置信息。例如,当组织130订阅安全性管理和控制系统102的服务时,组织可以向安全性管理和控制系统102提供识别组织针对云服务提供商110的租户账户的信息。在这个示例中,安全性管理和控制系统102可以将租户账户信息存储在租户配置信息124中。在一些情况下,组织130也可以提供针对服务提供商110的用户账户的列表,该列表也可以存储在租户配置信息124中。可替代地或附加地,在一些示例中,安全性管理和控制系统102可以查询服务提供商110以确定用户账户。在一些示例中,安全性监视和控制系统102可以以各种方式使用租户账户和用于该租户的用户的用户账户之间的关联,包括检索关于与该租户相关联的用户的活动的信息。

[0060] 在一些示例中,组织130还可以向安全性管理和控制系统102提供认证信息,其中认证信息可以用于登录或以其它方式访问服务提供商110的服务112a-112b。在各种示例中,安全性管理和控制系统102可以使用授权信息来访问组织130针对服务提供商110的租户账户。授权信息可以采取例如令牌(例如,OAuth令牌)、用户名和密码的形式或另一种形式的凭证。在一些示例中,组织130可以用授权来指定许可或特权,这可以定义安全性管理和控制系统102对组织的租户账户的访问级别。例如,组织130可以指定安全性管理和控制

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00402

系统102可以改变用于由服务提供商110提供的服务的安全设置，但是安全性管理和控制系统102不能生成新的用户账户或删除用户账户。

[0061] 在各种示例中，租户配置信息124可以存储组织130的其它信息，诸如例如安全性、用于访问服务提供商110的服务112a-112b的设置、日志设置和/或访问设置(例如，安全性策略、安全性配置以及白名单和黑名单等)。在一些示例中，组织130可以基于级别为服务指定安全设置。例如，处于高级别的安全设置可以要求用户密码“强”，例如，意味着密码必须包括各种字符(诸如大写和小写)、数字和/或符号。类似地，处于不同级别的安全设置可以将会话不活动定时器设置为较高或较低的时间段，使得，例如，当不活动定时器到期时，用户的会话自动结束。

[0062] 在一些示例中，存储装置122可以包括安全性信息126(“安全性信息”)，该安全性信息126包括由安全性监视和控制系统102执行的安全性分析。在一些示例中，安全性信息126可以包括用于安全性监视和控制系统102的不同客户的分开的条目。在一些示例中，安全性信息126包括可以在需要时进行查询的历史数据：过去分析的结果(例如，自上个月以来、最近三个月、去年或某个其它过去的时间段)。在一些示例中，安全性信息126还可以包括过去的安全性事件的记录、对过去的安全性事件是实际事件还是假阳性的确定、针对过去事件采取的补救动作的记录和/或执行补救动作的结果等数据。在一些示例中，安全性信息126还可以包括网络威胁情报数据，该数据例如是从第三方威胁情报聚合器和分发器获得的。

[0063] 在一些示例中，存储装置122可以包括关于服务提供商110和其它服务提供商的域信息128(“域信息”)。域信息128可以包括例如服务提供商110的网络地址或位置、用于服务提供商110的所有者或运营商的标识信息(例如，拥有和/或运营服务提供商110的个人或组织)、以及用于验证服务提供商110的身份的信息。例如，可以通过查询服务提供商110、从服务提供商110请求证书和/或从服务提供商的ISP或服务提供商的托管服务请求信息来获得域信息128。

[0064] 在一些示例中，存储装置122可以包括应用信息132(“应用信息”)，该应用信息记录服务提供商的应用或服务的用户。应用信息132可以包括例如从组织130收集的数据日志和/或从服务提供商110获得的活动日志。应用信息132可以记录例如在服务提供商110的服务112a-112b的使用期间执行的动作以及执行了这些动作的用户的标识、何时执行了动作的时间戳、当用户执行动作时网络的网络标识和/或用户的地理位置、受动作影响的资源，以及与服务112a-112b的使用相关的其它信息。

[0065] 在各种实施方式中，安全性监视和控制系统102可以耦合到一个或多个数据存储库180或与之通信。可以使用任何类型的持久存储设备来实现数据存储库180，诸如存储器存储设备或其它非暂态计算机可读存储介质。在一些示例中，数据存储库180可以包括或实现一个或多个数据库(例如，文档数据库、关系数据库或其它类型的数据库)、一个或多个文件存储库、一个或多个文件系统或其组合。数据存储库180可以由服务提供商系统提供的服务来实现和/或作为服务提供商系统提供的服务来访问。数据存储库180可以包括用于请求与服务或服务的服务提供商相关的数据的接口。

[0066] 在一些实施方式中，安全性监视和控制系统102可以包括日志收集器系统134，该日志收集器系统134执行用于获得安全性监视和控制系统102可以对其执行分析的数据的

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00403

操作。在各种示例中，安全性监视和控制系统102可以使用不同类型的数据或数据源来进行威胁分析。例如，安全性监视和控制系统102可以使用通过监视客户端设备106a-106c(服务112a-112b是通过客户端设备106a-106c而被使用的)而获得的数据，和/或通过监视组织的网络内的点(例如路由器或防火墙108处的点)获得的数据。在本文中，从客户端设备或组织的网络内获得的数据被称为网络数据。为了获得网络数据，在一些示例中，可以将监视代理放置在客户端设备106a-106c上和/或组织的网络的网络基础设施上。在这些示例中，监视代理可以在用户使用云服务时捕获用户活动。

[0067] 作为另一个示例，安全性监视和控制系统102可以使用来自服务提供商110的数据日志。在各种示例中，当服务提供商110的服务112a-112b被使用时，服务提供商110可以记录用户活动。例如，服务提供商110可以记录用户何时登录服务、用户登录时用户的网络和/或地理位置、用户使用服务时由用户执行的动作、受动作影响的资源，以及与服务的使用相关的其它信息。来自服务提供商的数据在本文中称为活动数据或活动日志。

[0068] 以下示例说明了可以从服务提供商获得的活动数据的一些示例。以下是可以从服务提供商获得的活动数据的第一示例(例如，审核日志记录)：

```
"entries":[
{
"source":{
"type":"user",
```

[0069]

```
"id":"222853877",
"name":"SandraLee",
"login":"sandra@company.com"
},
"created_by":{
```


Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00404

"type":"user",

"id":"222853866",

"name":"MikeSmith",

"login":"mike@company.com"

},

"created_at":"2016-12-02T011:41:31-08:00",

[0070] "event_id":"b9a2393a-20cf-4307-90f5-004110dec233",

"event_type":"ADD_LOGIN_ACTIVITY_DEVICE",

"ip_address":"140.54.34.21",

"type":"event",

"session_id":null,

"additional_details":null

}

[0071] 以下是可以从服务提供商获得的活动数据的第二示例(例如,共享文件审核日志记录)。

"entries":[

{

"type":"event",

[0072] "source":{

"parent":{

"type":"folder",

"id":"0",

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00405

```
                "name":"AllFiles"
            },
            "item_name":"financial2017Q1.doc",
            "item_type":"file",
            "item_id":"159004949136"
        },
        "additional_details":{
            "service_name":"FileSharingApp"
        },
        "event_type":"SHARE",
        "ip_address":"140.191.225.186",
        "event_id":"234d2f55-99d0-4737-9c3b-1a5256fe7c67",
        "created_at":"2016-12-12T20:28:02-07:00",
        "created_by":{
            "type":"user",
            "id":"238746411",
            "name":"JohnSmith",
            "login":"john@company.com"
        }
    }
```


[0073]

[0074] 以下是可以从服务提供商获得的活动数据的第三示例(例如,审核记录)。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00406

```
{
    "Records":[
    {
        "eventVersion":"1.03",
        "userIdentity":{
        "type":"IAMUser",
        "principalId":"111122225533",
        "arn":"arn:aws:iam::111122223333:user/john",
        "accountId":"111122223335",
        "accessKeyId":"AKIAIOSFODNN7JOHN",
        "userName":"john"
```

[0075]

```
    },
    "eventTime":"2016-12-26T20:46:31Z",
    "eventSource":"s3.amazonaws.com",
    "eventName":"PutBucket",
    "awsRegion":"us-west-2",
    "sourceIPAddress":"127.0.0.1",
    "userAgent":"[]",
    "requestParameters":{
        "bucketName":"myprodbucket"
    },
    "responseElements":null,
```


Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00407

"requestID":"47B8E8D397DCE7D6",

"eventID":"cdc4b7ed-e171-4cef-975a-ad829d4123c7",

[0076] "eventType":"AwsApiCall",

"recipientAccountId":"111122223344"

}

[0077] 以下是可以从服务提供商获得的活动数据的第四示例(例如,审核记录)。

[0078] 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47efcde myprodbucket[06/Feb/202017:00:01:57+0000]192.0.2.3 Mary DD6CC733AMARY REST.GET.OBJECT s3-dg.pdf"GET/mybucket/financial2016Q4.pdf HTTP/1.1"200--4406583 4175428"-""S3Console/0.4"-

[0079] 在一些示例中,安全性监视和控制系统102可以使用来自其它数据源的数据,诸如来自除组织的网络或服务提供商110之外的网络源的第三方馈送。其它网络源的示例可以包括网络安全数据聚合器和分发器、社交联网系统、新闻报道或聚合系统、政府系统、声誉系统和其它系统。

[0080] 在各种实施方式中,日志收集器系统134可以执行操作以获得网络数据和/或活动数据。例如,日志收集器系统134可以被配置为与和在企业网络104放置在一起的软件代理进行通信,以使由这些代理记录的数据联网。在这个示例中,网络数据可以包括当用户使用服务提供商110的服务112a-112b或另一个服务提供商的服务时生成的网络流量。在一些示例中,网络数据可以包括其它网络流量,例如当用户使用其它网络资源(诸如组织130的资源或互联网(诸如网站)上的资源)时生成的网络流量。作为另一个示例,日志收集器系统134可以被配置为与服务提供商110通信,以从服务提供商110获得活动日志。服务提供商110可以具有例如使日志收集器系统134能够请求活动日志的API。在这些示例中,日志收集器系统134可以能够使用针对服务提供商110的租户账户的凭证来请求活动日志;即,日志收集器系统134可以扮演服务提供商110的租户并以与租户相同的方式提出请求。

[0081] 在各种实施方式中,由日志收集器系统134获得的数据可以由安全性监视和控制系统102中的数据分析系统136处理。数据分析系统136可以对网络数据和活动数据进行分析以执行操作,诸如正在使用的应用的发现、活动模式学习和识别、异常检测以及网络威胁检测,等等。下面进一步讨论可以由数据分析系统136执行的这些和其它操作。

[0082] 在各种实施方式中,安全性监视和控制系统102的信息处置器系统138管理存储装置122中的数据,包括例如存储数据、定位和检索数据、组织数据以及更新数据,等等。在一些示例中,信息处置器系统138从组织130的用户(诸如管理性用户)接收数据,该用户可以提供信息,诸如组织的用户的列表和关于这些用户的数据。关于用户的数据可以包括例如用户的角色或特权。在这些和其它示例中,信息处置器系统138可以管理用户数据在存储装置122中的适当数据存储库中的存储。

[0083] 在各种实施方式中,映射生成器170对组织130或安全性监视和控制系统102的另一个客户执行安全性分析。例如,映射生成器170可以对数据分析系统136的输出进行操作,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00408

并计算服务、用户、租户或服务、用户和/或租户的组合的安全性的量度。在一些示例中，映射生成器170可以从接口120获得数据以执行安全性分析。安全性分析操作将在下面进一步讨论。

[0084] 在各种实施方式中，安全性监视和控制系统102中的控制管理器172可以代表组织130进行操作，以管理组织的用户对服务提供商110的服务112a-112b的访问。在各种示例中，控制管理器172可以为多个云服务或组织的用户使用的任何云服务提供此服务。在一些示例中，控制管理器172可以使用访问策略来控制用户对服务112a-112b的访问。访问策略可以定义(例如)可以使用或不能使用服务的用户的类型或类别、可以使用或不能使用服务的具体用户、可以使用服务的一天中的时间或一周中的某天、可以向服务传送和从服务传送的数据的量、可以使用的数据带宽的量、在使用服务时可以执行或不能执行的操作、访问策略适用于哪些用户，和/或关于服务的其它限制或权限。在各种示例中，控制管理器172可以为订阅安全性管理和控制系统102的服务的每个组织维护策略集。在一些示例中，控制管理器172可以为预设提供不同的安全性级别，其中选择预设为一个或多个服务选择安全性配置。在这些示例中，组织对预设的选择可以影响组织的一些或全部用户使用服务的能力。

[0085] 在一些示例中，控制管理器172还可以维护用于组织130的安全性策略。安全性策略可以定义动作或动作集，这些动作在被检测到时构成安全性违反或以其他方式需要注意的事件。在一些示例中，可以通过使用一个服务而发生由策略定义为违反安全性的动作，这意味着所有动作都是在使用同一服务时执行的。在一些示例中，动作可以在使用多于一个服务期间发生，其中服务是由一个服务提供商或多个服务提供商提供的。在一些示例中，安全性策略还可以定义要在检测到违反策略时执行的一个或多个补救动作。补救动作可以包括例如向造成了违反的用户、向组织130的网络管理员、向安全性管理和控制系统102的管理员和/或向另一个实体发送通知。

[0086] 在一些示例中，补救可以包括修改对服务或对多个服务的访问。例如，补救动作可以包括阻止特定用户使用服务，或者阻止组织130的所有用户使用服务。作为另一个示例，补救动作可以包括限制可以在使用服务时执行的动作。在一些示例中，执行补救动作可以包括向企业网络104发送指令。在这些示例中，指令可以例如配置客户端设备或网络基础设施设备(诸如防火墙108)。在一些示例中，客户端设备106a-106c和/或网络基础设施设备可以正在执行软件代理，该软件代理使控制管理器172能够与设备通信并且对设备进行修改。修改可以包括例如限制对域或IP地址的访问、阻止所有网络流量、禁用设备或对设备进行某种其它修改。

[0087] 在一些示例中，执行补救动作可以包括向服务提供商110发送指令以修改对服务112a-112b的访问。在这些示例中，补救动作可以包括确定要发送的指令。例如，控制管理器172可以检查服务提供商110和/或服务的API，以识别可以被执行以引起对服务的期望改变的指令。在这个示例中，API可以定义指令，该指令可以例如防止或限制组织130的用户或一组用户对服务的访问。作为另一个示例，API可以定义可以禁用或启用服务的某些功能的指令，其中禁用或启用影响组织130的一个或多个用户。在这些和其它示例中，对服务的修改影响与补救相关联的组织的用户，而不影响不同组织的用户。

[0088] 在各种实施方式中，安全性监视和控制系统102可以包括学习系统178。学习系统178可以将各种机器学习算法应用于由安全性监视和控制系统102收集的数据。然后例如可

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00409

以由数据分析系统136使用关于数据学习到的信息来做出关于在使用由服务提供商110提供的服务时的用户活动的确定。例如，学习系统178可以学习组织的用户的正常或普通行为的模式。在这些和其它示例中，学习系统178可以生成捕获学习系统178已经学习到的模式的模型，该模型可以与用于组织的其它数据一起存储在存储装置122中。

[0089] 为了支持学习系统178，在一些实施方式中，安全性监视和控制系统102包括扫描器174和模式分析器176。在这些实施方式中，扫描器174可以例如扫描数据以获取特定类型的信息。例如，扫描器174可以提取特定用户、用户组、用户类和/或与特定租户相关联的所有用户的活动。作为另一个示例，扫描器174可以提取与特定服务或服务集的使用相关的活动。作为另一个示例，扫描器174可以提取与特定服务提供商相关的活动。在各种实施方式中，模式分析器176可以使用由扫描器174提取的数据来识别数据中的模式。例如，用户和/或组织可以以重复的方式或以循环的方式使用服务。在这些示例中，模式分析器176可以识别重复行为，并且向学习系统178识别这些行为模式。

[0090] 在一些示例中，安全性监视和控制系统102中的数据访问器182可以与服务提供商通信，以从那些服务提供商获得活动数据。活动数据可以是针对用户账户、租户账户、组账户或另一类型的账户。可以针对服务、特定类型的数据(例如，特定属性的数据)、一个或多个用户、或者服务、服务提供商、属性、用户或其它因素的组合来获得活动数据。在一些示例中，数据访问器182可以处理数据以识别与一个或多个准则(诸如一个或多个服务、特定类型的数据(例如，特定属性的数据)、一个或多个用户或其组合)相关的活动。

[0091] 在各种实施方式中，安全性管理和控制系统102提供接口120，安全性管理和控制系统102的客户可以通过该接口使用安全性管理和控制系统102的服务。接口120可以提供例如图形用户界面(GUI)，该图形用户界面可以显示使组织的管理性用户能够配置安全性管理和控制系统102的服务的控制面板或仪表板。图形用户界面还可以使管理性用户能够查看关于服务提供商110的服务112a-112b的用户活动的报告。图形用户界面还可以提供安全性事件的报告并建议补救动作，和/或报告安全性管理和控制系统102自动执行的补救动作的结果。图形用户界面可以例如被实现为可以在组织130的客户端设备106a-106c上执行的软件应用。可替代地或附加地，图形用户界面可以被实现为基于web的界面(例如，网站)。

[0092] 在一些示例中，接口120可以可替代地或附加地提供API，组织130可以通过该API来管理由安全性管理和控制系统102提供的服务。该API可以定义动作，例如，在使用云服务时拉取关于用户活动的报告、拉取关于安全性事件的报告、发布用于补救动作的指令、获得关于用户活动的统计信息和/或与安全性管理和控制系统102的服务相关的其它动作。接口120的API可以使组织能够例如将安全性管理和控制系统102的能力集成到组织130的安全性基础设施中。

[0093] 在各种实施方式中，用于安全性监视和控制的系统可以包括多个部件，这些部件可以位于单个硬件平台上或彼此通信的多个硬件平台上。部件可以包括配置服务器或其它计算设备以执行安全性管理和控制系统102的操作的软件应用和/或模块。

[0094] 图2图示了可以由安全性管理和控制系统实现的示例云安全性系统200的框图。在各种实现方式中，示例云安全性系统200可以为服务提供商230的租户220进行网络威胁分析，并且确定租户220的用户在使用服务提供商230的服务时的动作是否构成网络威胁。在各种实施方式中，云安全性系统200可以包括用于与租户220接口的用户接口部件215和用

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00410

于与服务提供商230接口的提供商接口部件201。在后端，云安全性系统200可以包括用于进行分析的各种应用和用于存储在分析中使用的数据的数据存储库。

[0095] 在图2的示例的上下文中，租户220是服务提供商230的租户，这意味着租户220正在使用服务提供商230的服务。当云安全性系统200被提供为云服务时，租户220也可以是云安全性系统200的租户，因为租户220正在使用云安全性系统200的服务。

[0096] 在各种示例中，用户接口部件215包括管理控制台214和分析可视化控制台216。使用管理控制台214，租户220可以配置用于服务提供商230的服务的安全性控制。安全性控制的配置可以包括例如启用或禁用租户的用户对服务的访问、启用或禁用租户的用户可以使用的服务的特征，以及租户220可用的其它配置。分析可视化控制台216可以被用于查看由云安全性系统200生成的分析。例如，使用分析可视化控制台216，租户220可以查看涉及租户的用户和租户220所订阅的服务的安全性事件的报告。在各种示例中，可以从云安全性系统200的数据存储库获得在管理控制台214和分析可视化控制台216中显示的信息。

[0097] 在各种实施方式中，管理控制台214可以向租户220提供针对多个云服务和/或云服务提供商的控制的规格化视图。管理控制台214可以包括用户界面，该用户界面在同一屏幕上显示用于不同云服务的控制的简化视图。提供给管理控制台214的信息可以是使用基于元数据的模式映射从应用目录数据库208检索的。在一些示例中，管理控制台214可以被用于跨云服务指派一致的访问策略。在这些示例中，管理控制台214可以根据指定的分类器(诸如例如标准、严格或自定义等)来显示控制和/或设置控制。在这个示例中，越高级别的分类与越严格的控制对应。在一些示例中，安全性控制的分类和/或指定符合由组织(诸如美国国家标准技术研究院(NIST)、国际标准化组织(ISO)和/或支付卡行业数据安全标准(PCI DSS))指定的准则，和/或一个这种组织提供的具体证书。在一些示例中，管理控制台214还可以提供用于与SaaS、PaaS和原生应用集成的插件接口。

[0098] 在各种实施方式中，分析可视化控制台216可以以库格式显示安全性指标，其具有用颜色(诸如红色、绿色、黄色)编码的风险因素。可以显示其它统计信息或度量，诸如例如用户登录尝试、具有最新添加的用户的组、已删除的文件、具有最多已删除的文件的用户和/或下载最多文件的用户，等等。一些类型的信息可以特定于特定的服务提供商。例如，对于Salesforce.com，度量可以包括正在下载机会或预算数据、合同或联系人的用户的身份。在一些示例中，分析可视化控制台216提供用于租户的云服务的安全性控制的统一视图。分析可视化控制台216可以显示用于不同云服务的任何或所有安全性控制集而值集，以及当前值相对于与预定策略或配置相关联的值的偏差。

[0099] 在各种示例中，提供商接口部件201可以是云安全性系统200可以用来与服务提供商230进行接口的应用。这些部件可以包括云爬取器(crawler)应用202、云播种器应用204和数据加载器应用206。

[0100] 在各种示例中，云爬取器应用202可以从服务提供商230检索关于安全性控制的信息。可以将由云爬取器应用202检索到的数据输入到应用目录数据库208中。检索信息可以包括例如向服务提供商230发送请求，该请求被制定为向服务提供商230询问可用于服务的安全性控制以及用于安全性控制的任何可用设置。可替代地或附加地，服务提供商230可以包括API，云爬取器应用202可以通过该API获得用于服务的安全性控制。在各种示例中，云爬取器应用202可以从服务提供商230检索软件定义的安全性配置数据。软件定义的安全性

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00411

配置数据可以描述针对特定服务的安全性控制的配置。安全性控制是限制访问云服务提供商所提供(house)的应用和/或数据的机制。例如,软件定义的安全性配置数据可以包括描述为以下各项定义的角色的数据:用户、组和用户的分组;加密密钥;令牌;访问控制;权限;配置;认证策略的类型;移动访问策略;以及许多其它类型的安全性控制。

[0101] 在各种示例中,云爬取器应用202可以连接到服务提供商230,以便获得软件定义的安全性配置数据。服务提供商230可以要求用于访问服务提供商的系统的授权或某种其它形式的同意。授权可以由令牌(诸如使用用于授权的开放授权(OAuth)开放标准)或由凭证(诸如用户名和密码)提供。令牌或凭证可以是租户220的令牌或凭证,或者可以是与云安全性系统200相关联的令牌或凭证。可以使用各种其它技术来授权对云提供商的系统和数据的访问。该连接还可以包括提供服务URL。

[0102] 在各种示例中,可以通过使用服务提供商230的API来收集软件定义的安全性配置数据。API和API类的示例包括代表性状态传输(REST)、Java2平台、企业版(J2EE)、简单对象访问协议(SOAP)和原生编程方法(诸如用于Java的原生应用API),等等。还可以使用其它技术来请求信息,诸如脚本语言(诸如Python和PHP)、部署描述符、日志文件、通过Java数据库连接(JDBC)或REST的数据库连接性以及驻留应用(诸如云信标)。被发送或接收的信息可以以各种格式表示,诸如JavaScript对象表示法(JSON)、可扩展标记语言(XML)或逗号分隔值(CSV)等等。

[0103] 下面的表1提供了由云服务提供商Box和Amazon Web Services(AWS)提供的一些安全性控制的示例。其它云服务提供商可以具有类似的安全性控制或其它安全性控制。

[0104] 表1

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00412

[0105]

安全性控制	Box 中支持	Amazon Web Services (AWS)中支持
用户/组管理	REST(代表性状态传输)API	AWS IAM(身份和访问管理)API
凭证和标识符	N/A	保护和监视账户、令牌、密钥等
登录/登出事件	REST API	AWS 云踪迹-事件 API 和日志文件
客户端的 IP 地址	REST API	AWS 云踪迹-事件 API 和日志文件
客户端使用的设备（iphone、ipad 等）	REST API	AWS 云踪迹-事件 API 和日志文件
密码策略	REST API	AWS IAM 策略
资源访问权限	资源：文件、文件夹 动作：编辑、预览、上传、协调事件	资源：EC2、S3、EBS 动作：创建、访问、重启、终止等。 基于 IP 地址的访问控制

[0106]

限定或限制移动访问	限制用户保存内容以供离线访问	AWS IAM 策略
角色	BOX 具有预定义的管理员角色	角色可以使用预定义的策略来创建

[0107] 表2提供了用于云服务提供商Salesforce.com的安全性控制和受支持的访问中的一些的示例。其它云服务提供商可以使用类似的或其它安全性控制和访问控制。

[0108] 表2

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00413

[0109]

安全性控制	在 Salesforce.com 中支持
用户/组管理	SalesForce 用户/组/简档 API
凭证和标识符	API：设置改变
登录/登出事件	API：审核活动
客户端的 IP 地址	API：审核活动
客户端使用的设备（iphone、ipad 等）	管理设置改变的 API
密码策略	API：设置改变
资源访问权限	使用对象历史进行的 Salesforce 对象监视
限定或限制移动访问	管理设置改变的 API
角色	Salesforce 简档

[0110] 在各种示例中，云爬取器应用202可以根据从服务提供商230获得的软件定义的安全性配置数据来生成安全性控制元数据。安全性控制元数据是规格化的描述符，用于将信息输入到公共数据库(诸如应用目录数据库208)中。可以对安全性控制元数据进行分类(例如，映射到类别中)并索引。分类可以符合由安全性组织指定的标准和/或可以由第三方认证和/或审核。此外，可以围绕特定法规或标准的要求来制定安全性控制元数据和/或元数据的分类。例如，诸如健康保险可移植性和责任法案(HIPAA，Health Insurance Portability and Accountability Act)、萨班斯-奥克斯利法案(Sarbanes-Oxley Act)、联邦风险与授权管理计划(FedRAMP，Federal Risk and Authorization Management Program)和/或支付卡行业数据安全标准(PCIDSS，Payment Card Industry Data Security Standard)之类的法规和标准可以要求报告和审核踪迹。可以以显示法规和标准所需的信息的类型的方式来格式化安全性控制元数据，并促进所需的报告的生成。

[0111] 安全性控制元数据可以存储在应用目录数据库208中。在一些示例中，应用目录数据库208是Apache Cassandra数据库，这是被设计为跨许多商用服务器处置大量数据的开源NoSQL数据库管理系统。在一些示例中，应用目录数据库028是使用适于该应用的数据库的类型来实现的。在一些示例中，可以使用各种数据库来存储应用目录，以供以后检索、报告生成和分析生成。

[0112] 在各种实施方式中，可以使用其它方法来检索软件定义的安全性配置数据并生成

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00414

安全性控制元数据。此外，用于检索软件定义的安全性配置数据的各种类型的控制和机制可以由不同的云服务提供商支持。例如，其它云应用(诸如Office 365、GitHub、Workday和各种Google应用)都使用特定于服务的检索机制。此外，取决于云服务提供商支持什么，用于检索软件定义的安全性配置数据的处理可以被自动化。

[0113] 在各种实施方式中，云播种器应用204可以被用于实现用于租户220的安全性策略。云播种器应用204可以例如为针对服务提供商的服务的租户账户设置安全性控制。安全性控制可以影响例如一个用户账户、多个用户账户或所有用户账户。在一些示例中，云播种器应用204可以在各种情况下设置安全性控制。例如，云播种器应用204可以作为对威胁的补救的一部分，或者在由租户220调用时，设置安全性控制。在各种示例中，云播种器应用204可以被用于协调跨云服务的一致访问策略。在一些示例中，安全性控制可以跨组织所具有的针对不同服务提供商的多个账户来协调。例如，可以定义不同的安全性级别，使得在选择更高或更低级别的安全性时，组织针对不同云服务的账户的安全性控制都被设置为反映该更高或更低级别的安全性。以这种方式，可以强制实施统一的策略和安全性控制配置。可以通过输入到用户界面(诸如上面讨论的控制管理平台)中来定义处于不同的安全性级别的各种安全性控制的值，并且可以将与处于每个安全性级别的安全性控制相关联的值存储在数据库中。可以提供用户界面，以示出组织的针对云服务的账户的安全性控制以及安全性控制值处于一安全性级别的指派。作为示例，处于“严格”安全性级别的安全性控制可以包括针对用户账户的密码要求，诸如最少十个字符、两个数字、一个特殊字符、一个大写字母、不重用最后十个密码，等等。

[0114] 在各种实施方式中，数据加载器应用206可以从服务提供商230检索租户220的活动数据。活动数据可以来自当租户的用户使用服务提供商的服务时由服务提供商230生成的日志。在各种示例中，数据加载器应用206可以通过从服务提供商230请求数据来获得活动数据。可以将由数据加载器应用206检索到的数据输入到着陆(landing)储存库210和/或分析和威胁情报储存库211中。输入到着陆储存库210中的数据可以具有不同的格式和/或具有不同的值范围，例如由于它们是从不同的服务提供商收集的。在一些示例中，来自数据加载器应用206的数据可以在被移至分析和威胁情报储存库211之前被重新格式化和/或结构化，使得例如数据具有统一格式。

[0115] 在各种示例中，数据加载器应用206可以通过连接到服务提供商230并与之通信来获得活动数据。在各种示例中，该连接是通过加密的通信信道进行的。在一些示例中，可以通过令牌或使用登录凭证或另一种认证方法对连接进行认证。在一些示例中，活动数据的收集被调度为周期性地发生(例如，每四个小时、每六个小时或以某个其它时间间隔)。在一些示例中，用于收集的调度可由租户220配置。在一些示例中，数据加载器应用206使用实时计算系统(诸如例如Apache Storm，开源分布式实时计算系统)在事件发生时实时地收集数据。数据加载器应用206可以被配置为将某些事件或活动指定为高风险事件，并且这些事件可以在所调度的检索间隔之外近乎实时地被检索。

[0116] 在各种示例中，活动数据可以包括关于服务提供商的服务的用户的各种类型的信息。例如，与用户账户相关联的活动数据可以包括与服务的用户账户的使用和/或对服务的用户账户所采取的动作相关的信息。在这个示例中，活动数据可以包括信息的源，诸如用户日志和/或审核踪迹。活动数据的更具体类型可以包括例如登录和登出统计信息(包括尝试

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00415

和成功)、文件操作、访问度量、网络下载/上传度量、应用度量(例如,使用、操作、功能等)、用于访问服务的IP地址、用于访问服务的设备,和/或被访问了的云资源(诸如例如文件管理云应用[诸如Box]中的文件和文件夹、人力资源云应用[诸如Workday]中的员工和承包商、和/或客户关系管理云应用[诸如Salesforce]中的联系人和客户)。在各种示例中,活动数据可以包括与事件或统计信息相关联的用户的用户账户或其它用户标识符。在各种示例中,活动数据可以包括关于云系统的系统状况或活动的信息,诸如例如服务器活动、服务器重新引导、服务器所使用的安全密钥以及系统凭证,这种信息是使用经授权的凭证的系统可见的或可访问的。

[0117] 在一些示例中,活动数据还可以包括关于租户账户的安全性配置的信息,可能包括与该租户账户相关联的用户账户的安全性配置。安全性配置可以包括用于租户和/或与该租户相关联的用户账户的安全性控制被设置为的值。

[0118] 在各种示例中,数据加载器应用206可以将检索到的活动数据存储在分析和威胁情报储存库211中。分析和威胁情报储存库211可以是具有查询能力的任何数据库或数据储存库。在一些示例中,虽然可以将任何数据仓库基础设施适当地用于应用,但是分析和威胁情报储存库211在基于NoSQL的基础设施(诸如Apache Cassandra)或另一个分布式数据处理系统中构建。在一些示例中,在将数据移动到分析和威胁情报储存库211之前,首先将该数据输入到着陆储存库210中并重新格式化和/或结构化。

[0119] 在一些示例中,活动数据可能是以由不同服务提供商或服务使用的不同格式而接收的。例如,数据可以以JSON或其它数据交换格式来格式化,或者可以作为日志文件或数据库条目而可用。在一些示例中,数据加载器应用206执行用于规格化数据并将数据重新格式化为用于存储在分析和威胁情报储存库211中并从中检索的通用格式的操作。将数据重新格式化可以包括将数据分类并结构化为通用格式。在一些示例中,数据库适于结构化改变和新值,并且可以运行自动化的处理以检查改变的数据。在一些示例中,云爬取器应用202识别检索出的数据的结构或值的差异,并且可以将改变应用于应用目录数据库208和/或分析和威胁情报储存库211。

[0120] 在一些示例中,数据加载器应用206可以预生成系统报告。系统报告可以由被调度以周期性间隔在数据集上运行的作业(例如,进程)生成。存储在应用目录数据库208和/或分析和威胁情报储存库211中的数据可以被用于生成各种报告。报告的类别可以包括例如认证和授权、网络和设备、系统和改变数据、资源访问和可用性、恶意软件活动以及故障和严重错误,等等。报告可以基于各种属性,诸如例如按应用、按用户、按受保护的资源以及按用于访问的设备,等等。报告可以突出显示最近的改变,诸如云应用中更新后的功能或新修改后的策略。报告可以由所调度的作业(例如,出于性能原因)预先生成或者可以由用户或管理员请求。

[0121] 在一些示例中,报告包括对数据生成的分析。分析可以使用Apache Software Foundation技术(诸如Hadoop、Hive、Spark和Mahout)或所使用的数据存储框架中可用的其它特征。在一些示例中,R编程语言被用于生成分析。在一些示例中,分析的生成包括使用机器学习算法、专有算法和/或来自外部商业源(诸如FireEye和Norse)或公共威胁情报社区(诸如Zeus和Tor)的外部威胁情报。

[0122] 在各种实施方式中,关于访问模式和其它事件统计信息的分析和威胁情报储存库

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00416

211中的活动信息的聚合使系统200能够建立行为的基线。例如，机器学习技术可以应用于检测威胁并提供关于如何对威胁进行响应的推荐。可以开发威胁模型来检测已知或未知或正在出现的威胁。威胁也可以通过将活动数据与外部威胁情报信息(诸如由第三方提供商提供的信息)进行比较来识别，如下面进一步讨论的。在各种示例中，分析和威胁情报储存库211中的数据还可以被用于生成可以经由用户界面可视地呈现给系统管理员的报告，以及生成用于确定威胁级别、检测具体威胁以及预测潜在威胁的分析，等等。

[0123] 在一些示例中，租户220的单个用户可以具有针对服务提供商230和/或针对由服务提供商230提供的服务的多个账户。在各种示例中，一个用户的各个用户账户可以在用户身份储存库209中关联在一起。在一些示例中，用户身份储存库209可以将用户针对多个服务提供商所具有的用户账户分组在一起。在一些示例中，租户220可以具有针对服务提供商230的租户账户。在这些示例中，用户身份储存库209可以将租户220的用户与租户账户相关联，这也可以将用户的用户账户与租户账户相关联。用户账户与租户账户的关联可以通过各种方式被使用，诸如检索关于租户的用户的用户活动的信息。在一些示例中，可以使用租户账户针对服务提供商230的凭证来登录服务提供商230，以检索与该租户账户相关联的用户账户的活动数据。

[0124] 在各种实施方式中，用户身份储存库209还可以被用于促进对用户活动的跟踪和简档的生成，其中简档可以描述特定用户对云服务或多个云服务的使用。在一些示例中，云安全性系统200可以使用用户的简档来采取影响多个云服务的动作。例如，当用户在使用若干云服务的活动可能是安全性风险时，云安全性系统200可以抢先警告系统管理员。可替代地或附加地，作为另一个示例，云安全性系统200可以通过应用补救措施来主动保护用户在其上维护数据的其它服务，诸如向认证添加附加步骤、改变密码、阻止一个或多个特定IP地址、阻止电子邮件或发件人、或锁定账户，等等。

[0125] 在各种实施方式中，云安全性系统200可以包括对由云安全性系统200收集的数据执行分析的应用或软件模块。该应用或软件模块可以存储在易失性或非易失性存储器中，并且在被执行时，可以将处理器配置为执行某些功能或处理。这些应用可以包括描述性分析应用207和预测分析应用212。在一些示例中，描述性分析应用207可以生成分析，诸如关于用户、用户活动和用户使用的资源的统计信息。在一些示例中，威胁检测和预测分析应用212可以使用机器学习和其它算法来生成分析。由预测分析应用212执行的分析可以包括根据活动和行为模型的模式来识别和预测安全性威胁。可以使用存储在分析和威胁情报储存库211中的数据来执行由描述性分析应用207和预测分析应用212执行的分析。

[0126] 在各种实施方式中，云安全性系统200可以包括补救功能，该补救功能提供用于对威胁进行响应的手动和/或自动处理。在一些示例中，分析可以使用从租户系统接收的描述由租户提供的威胁情报的信息。这些源(在示例系统200中被称为租户基线217)可以包括诸如以下信息：要观察或阻止的具体IP地址、要观察或阻止的用户、要观察或阻止的电子邮件地址、要监视的软件弱点、可能易于被滥用的浏览器或浏览器版本，和/或脆弱的移动设备或移动硬件或软件的版本，等等。在一些示例中，分析可以使用从外部第三方馈送218接收的信息。第三方馈送218的源可以是例如威胁情报聚合器或分发器。通过提供关于安全性威胁的外部信息，来自第三方馈送218的信息可以被用于增强云安全性系统200的威胁分析。外部信息可以包括例如被感染的节点的识别、来自特定源IP地址的恶意活动、被恶意软件

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00417

感染的电子邮件消息、脆弱的web浏览器版本，以及对云的已知攻击，等等。

[0127] 在各种实施方式中，事件补救应用213可以被用于响应于检测到的威胁而协调和/或执行补救动作。在一些示例中，当在警报中呈现并选择推荐的补救动作时，可以调用事件补救应用213。事件补救应用213可以执行所选择的补救动作，或者指示另一个应用(诸如云播种器应用204)执行所选择的补救动作。当所选择的补救动作要被手动执行或在安全性系统200外部时，事件补救应用213可以跟踪补救动作的状况以及补救动作是否完成。在一些示例中，事件补救应用213可以被用于存储手动或自动补救动作的结果。在一些示例中，所选择的补救动作将由安全性系统200外部的系统(诸如由第三方的或租户的事件补救系统)执行。在这些示例中，事件补救应用213可以指示或调用第三方的或租户的事件补救系统以执行动作。例如，事件补救应用213可以能够访问第三方或租户220的自动集成处理。

[0128] 图3图示了安全性管理和控制系统的示例分析引擎300的框图。在各种示例中，分析引擎300可以分析各种数据源以识别针对其用户正在使用云服务的组织的网络威胁。在各种示例中，分析引擎300的操作可以被用于检测和/或解决各种威胁场景。

[0129] 威胁场景的一个示例是IP跳跃(IP hopping)。在IP跳跃场景中，攻击者可以在发起攻击之前使用一个或多个代理服务器来隐藏攻击者的真实位置或机器身份。对这种类型的场景的检测可以涉及用于连接到云应用的每个IP连接的地理解析(例如，识别或查找与IP地址相关联的地理位置)。检测还可以包括检测空间数据中的异常特点，并根据这个信息预测威胁。用于检测的度量可以包括例如用户每天使用的唯一IP地址的数量的计数和/或速度，这可以指使用不同IP地址之间的时间差和/或每个IP地址所使用的持续时间。

[0130] 威胁场景的另一个示例是不寻常地理位置场景。不寻常地理位置场景可以指活动源自非预期的位置或已建立的模式之外的位置。这种场景可以包括诸如但不限于从不寻常地理位置成功登录或上传/下载文件之类的活动。

[0131] 威胁场景的另一个示例是暴力攻击。暴力攻击的示例是当攻击者尝试许多密码以便发现正确的密码并破坏用户账户的尝试。检测可以涉及评估失败的登录尝试的速度和事件活动中的模式，以预测暴力攻击。在一些示例中，暴力攻击可以具有不同的速度，诸如缓慢的攻击速度或快速的攻击速度。用于检测的度量可以包括例如对于现有有效账户的不寻常高数量的登录失败和/或利用无效或终止/暂停的用户名的不寻常高数量的登录尝试。

[0132] 威胁场景的另一个示例是内部威胁。内部威胁可以指人从网络内部实施的安全漏洞。例如，组织的已经授权的员工在其被组织雇佣期间可能滥用授权并有意或无意地造成(case)安全漏洞。检测内部威胁可以涉及跟踪用户的正常行为，并在与用户的一个或多个账户相关联的事件或活动偏离正常时生成警报。度量可以包括例如经常性大量使用公司资源，诸如大量下载和/或评级较低的员工下载或共享数量不寻常多的文件/文件夹、从源代码控制系统中删除代码，或下载、删除或修改客户信息，等等。

[0133] 威胁场景的另一个示例是应用滥用。应用滥用是一种场景，其可以包括例如与终止或暂停的员工相关联的事件(例如，使用过期或撤消的用户账户、使用诸如SSH密钥之类的密码密钥，等等)或被恶意软件感染的设备使用有效凭证但不寻常的地理位置或IP地址执行次数不寻常的文件下载/上传。

[0134] 虽然上面讨论了具体的威胁情景和可以被用于辨别这些场景的信息类型，但是本领域技术人员将认识到的是，威胁检测和预测可以利用各种信息和公式中的任何一种。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00418

[0135] 在各种示例中，分析引擎300可以通过检查各种外部和内部数据源来检测以上讨论的威胁场景以及其它威胁场景。外部数据源可以提供从云服务提供商获得的活动数据310。在一些示例中，外部数据可以可选地包括租户基线317和第三方数据318。在一些示例中，内部数据源可以包括由行为分析引擎304确定的数据模型，并且可以可选地包括由安全性管理和控制系统维护的威胁情报数据314。

[0136] 在各种示例中，当用户使用云服务时，云服务可以存储用户活动。例如，云服务可以存储用户登录以使用服务的每次发生、用户在使用服务时执行的动作、受动作影响的资源、被移入、移出或在服务内的数据和/或用户在会话结束时登出，等等。在这些和其它示例中，可以将用户的活动存储在日志文件中，该日志文件在本文中被称为活动日志。活动日志中的条目可以包括例如执行的动作或执行的动作的描述、执行动作的用户的标识、受动作影响的资源、执行动作的时间或动作在什么时间发起和/或完成，和/或执行动作的用户的网络位置或地理位置等信息。在图3的示例中，活动数据310可以包括用于多个服务和/或多个服务提供商的活动日志。在这些和其它示例中，一个活动日志可以包括用于一个服务或由同一服务提供商提供的多个服务的用户活动。

[0137] 在各种示例中，分析引擎300每天一次、隔天一次或在另一个时间间隔周期性地接收更新后的活动数据310。在一些示例中，当某些事件发生时，分析引擎300接收活动数据310，诸如服务指示事件已经发生(例如，服务已被更新、或者服务已检测到网络威胁、或另一个事件在该服务处发起)、组织指示事件已经发生(例如，组织已将用户添加到服务、或网络管理员请求更新后的分析、或另一个事件在该组织处发起)，或安全性管理和控制系统指示事件已经发生(例如，接收到新的威胁情报数据314、或另一个事件在安全性管理和控制系统处发起)。

[0138] 在一些示例中，来自不同云服务的活动日志可以具有不同格式。例如，一个活动日志中的条目可以被提供为逗号分隔的值，而另一个活动日志可以使用JSON语法。在这些和其它示例中，活动日志中的数据可以由分析引擎300规格化，或者在被提供给分析引擎300之前被规格化。规格化活动数据310包括重新格式化活动数据310，使得来自不同服务和/或服务提供商的数据是可比较的、具有相同的含义和/或具有相同的意义和相关性。在规格化之后，行为分析引擎304可以以有意义的方式聚合和比较来自不同云服务的数据。例如，由一个用户对一个云服务进行的一系列失败的登录尝试可以被视为不是威胁。但是，同一用户在多个不同的云服务处的一系列登录失败指示破解用户的密码的协力努力，因此应当发出警报。

[0139] 在各种示例中，可以由行为分析引擎304将活动数据310摄取到分析引擎300中。在各种实施方式中，行为分析引擎304可以从活动数据310收集统计信息，并从活动数据310识别行为特点。统计信息可以包括例如动作(诸如成功的登录尝试或失败的登录尝试)的计数。在一些示例中，统计信息可以与特定服务提供商、特定服务、特定用户、可以在使用服务时执行的特定动作、特定时间范围、其它因素和/或因素组合相关联。

[0140] 在各种实施方式中，行为分析引擎304可以使用从活动数据310生成的统计数据来确定活动简档，该活动简档在本文中也被称为行为简档。例如，行为分析引擎304可以生成活动简档，该活动简档描述特定组织的用户对服务的通用或典型使用模式。作为另一个示例，行为分析引擎304可以为特定用户或用户组生成活动简档。在这个示例中，活动简档可

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00419

以描述用户在使用一个或多个服务时的活动。在各种示例中，为了识别用户跨服务的活动，行为分析引擎304可以能够访问用户身份数据，该用户身份数据可以将特定用户针对不同云服务的账户链接在一起。用户身份数据可以包括例如针对每个云服务的用户名或其它形式的标识。在各种示例中，用户身份数据可以由组织提供。可替代地或附加地，在一些示例中，安全性管理和控制系统可以自动确定不同的用户账户是相关的。例如，安全性管理和控制系统可以假设应当链接具有相同用户名或源自相同IP地址或MAC地址的用户账户。

[0141] 在一些示例中，行为分析引擎304可以在用户的活动简档中包括上下文数据。可以从例如第三方数据318获得上下文数据，其中第三方数据318的源是声誉系统、社交媒体系统、新闻聚合器或提供商，或者可以维护关于用户的信息的另一个系统。上下文数据的示例包括来自旅行应用或电子邮件的旅行位置和行程、来自医疗保健管理系统的员工状态、来自Salesforce应用的敏感财务时间段，和/或来自电子邮件服务器的敏感电子邮件，等等数据。在一些示例中，可以附加地或可替代地从用户使用的客户端设备获得上下文数据。在这些示例中，上下文数据可以包括例如客户端设备的类型的标识、由客户端设备使用的IP地址、由客户端设备的全球定位系统(GPS)接收器计算的地理位置数据，以及关于客户端设备或可以从客户端设备获得的其它信息。

[0142] 在各种示例中，活动简档可以覆盖不同的时间段。在一些示例中，活动简档可以使用固定的移动窗口来覆盖以周为单位测量的时间段。在一些示例中，可以生成“新兴简档”，其捕获相对较近期(诸如上周内或目标日期之前的一周内)的事件。在一些示例中，可以生成“稳定简档”，其包括在上四(或八)周内或目标日期之前的四(或八)周内的事件。在各种示例中，可以生成其它简档或简档类型。

[0143] 在一些示例中，固定的移动窗口可以是不重叠的。即，时间进一步倒退的窗口可以排除时间上较近的窗口中的事件。例如，八周简档不包括四周简档或一周简档中的事件，并且类似地，四周简档不包括一周简档内的事件。每日(或周期性)聚合处理可以在日内或日间运行。

[0144] 下面的表3示出了针对一些用户活动的计算出的示例统计信息。示例用户活动包括四周窗口简档的平均登录计数(“avglogcntday4wk”)、四周窗口简档的平均登录IP地址计数(“avglogipcntday42k”)、一周窗口简档的登录计数的标准偏差(“stdlogcntday1wk”)，以及一周窗口简档的登录IP地址计数的标准偏差(“stdlogipcntday1wk”)。取决于可用数据和/或正被预测的威胁，可以计算类似的和其它统计信息。

[0145] 表3

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00420

[0146]

用户 ID	avglogcntd ay_4wk	avglogipcntd ay_4wk	stdlogcntd ay_1wk	stdlogipcntd ay_1wk
User1	5	4	3	2
User2	6	2	2	1
User3	4	3	2	2
User4	4	4	2	1
User5	5	5	1	1

[0147] 诸如以上所示的那些之类的统计信息可以被组合成特征向量。特征向量可以包括例如登录次数的计数、用于登录的不同IP地址的数量的计数、在24小时时间段内用于登录的任何两个IP地址之间的最大距离、在24小时时间段内用于连接到云应用的不同浏览器的数量的计数，和/或其它量度。可以按云应用和/或按云应用按用户来聚合特征向量。

[0148] 下面的表4示出了示例每日聚合矩阵向量。第一列提供了示例应用提供商，第二列说明了提供商可以支持的向量维度，并且第三列说明了可以指派给每个维度的值。

[0149] 表4

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00421

[0150]

应用	维度	描述
Amazon，Salesforce，Box	登录	(#of count, Avg, Stddev, Max)
Amazon，Salesforce，Box	失败的登录	(#of count, Avg, Stddev, Max)
Amazon，Salesforce，Box	登录 IP	(#of count, Avg, Stddev, Max)
Amazon，Salesforce，Box	失败的登录 IP	(#of count, Avg, Stddev, Max)
Box	下载	(#of count, Avg, Stddev, Max)
Box	下载 IP	(#of count, Avg, Stddev, Max)
Salesforce	浏览器	(#of count, Avg, Stddev, Max)
Salesforce	大量删除、大量传送、数据导出	(#of count, Avg, Stddev, Max)
Salesforce	证书和密钥管理	(#of count, Avg, Stddev,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00422

[0151]

		Max)
Salesforce	网络访问和 IP 白名单改变	(#of count, Avg, Stddev, Max)
Salesforce	管理用户改变	(#of count, Avg, Stddev, Max)
Salesforce	平台	(#of count, Avg, Stddev, Max)
Salesforce	密码策略改变	(#of count, Avg, Stddev, Max)
Salesforce	共享设置改变	(#of count, Avg, Stddev, Max)
Amazon	EC2 实例改变	(#of count, Avg, Stddev, Max)
Amazon	安全组改变	(#of count, Avg, Stddev, Max)
Amazon	SSH 密钥对改变	(#of count, Avg, Stddev, Max)
Amazon	网络 ACL 改变	(#of count, Avg, Stddev, Max)
Amazon	VPN 连接改变	(#of count, Avg, Stddev,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00423

[0152]

		Max)
Amazon	SAML 改变	(#of count, Avg, Stddev, Max)
Amazon	VPC 改变	(#of count, Avg, Stddev, Max)
Amazon	IAM 访问密钥改变	(#of count, Avg, Stddev, Max)

[0153] 下面的表5列出了若干可能的每日聚合矩阵向量的示例值。这里说明的示例向量包括一天中的每日登录计数(“logcntday_1dy”)、一天中的每日失败登录计数(“logfailcntday_1dy”)、一天当中发生失败登录的IP地址每日计数(“logfailipdisday_1dy”)和一天当中用于登录的IP地址的每日计数(“logipdisday_1dy”)。

[0154] 表5

[0155]

用户ID	logcntday_1dy	logfailcntday_1dy	logfailipdisday_1dy	logipdisday_1dy
User1	5	4	3	2
User2	6	2	2	1
User3	4	3	2	2
User4	4	4	2	1
User5	5	5	1	1

[0156] 在各种示例中,由行为分析引擎304确定的活动简档可以由威胁检测引擎302使用,以识别可能对正在使用云服务的组织构成威胁的云服务的使用情况。在一些示例中,威胁检测引擎302应用安全性策略来识别威胁。安全性策略可以描述事件,当该事件发生时,该事件将引起组织和/或安全性管理和控制系统的注意。例如,安全性策略可以指定需要引起组织注意的动作,诸如下载包含信用卡号的文件、复制加密密钥、提升普通用户的特权等。在一些示例中,安全性策略还可以描述在检测到事件时将要采取的动作,诸如阻止对服务的访问或禁用用户账户。

[0157] 在一些示例中,威胁检测引擎302进行异常检测以识别威胁。异常检测可以涉及搜索偏离已建立的规范的统计变化。在一些示例中,威胁检测引擎302的操作可以通过租户基线317来增强,该租户基线可以包括来自组织的威胁情报。在一些示例中,威胁检测引擎302可以可替代地或附加地接收由安全性管理和控制系统维护的威胁情报数据314,和/或接收包括来自例如威胁情报聚合器或分发器的威胁情报的第三方数据318。

[0158] 以下提供了可以被用于异常检测的各种示例算法。提供这些算法作为示例,并且可以使用其它算法。

[0159] 算法1是可以被用于确定登录IP地址变化的算法的一个示例。可以在不同时间段内为登录IP地址特征向量计算Z得分。下面的示例使用一周、四周和八周作为不同时间段的

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00424

示例，得出三个Z得分：

[0160] $$L1\ Z得分 = \frac{过去\ 24\ 小时登录IP - 1\ 周Avg登录\ IP}{1\ 周Stddev\ 登录\ IP}$$

[0161] $$L2\ Z得分 = \frac{过去\ 24\ 小时登录IP - 4\ 周Avg登录\ IP}{4\ 周Stddev\ 登录\ IP}$$

[0162] $$L3\ Z得分 = \frac{过去\ 24\ 小时登录IP - 8\ 周Avg登录\ IP}{8\ 周Stddev登录\ IP}$$

[0163] Z得分可以与指派给每个得分的权重(w1...w3)组合：

[0164] L组合＝(w1×L1Z得分)+(w2×L2Z得分)+(w3×L3Z得分)

[0165] 在一些示例中，权重之和为一。取决于何时执行计算，可以动态地计算所应用的权重。例如，在第一天，可以使用基于现有数据计算的值来应用默认基线，包括默认Avg(平均值)和默认Stddev(标准偏差)。作为另一个示例，对于从第二天开始的第一周，L1 Z得分可用，使得权重可以被设置为w1＝1、w2＝0、w3＝0。继续该示例，在五周之后，L1和L2 Z得分可用，并且权重可以被设置为w1＝0.4、w2＝0.6、w3＝0。在14周之后，L1、L2和L3 Z得分可用，使得权重可以被设置为w1＝0.2、w2＝0.3、w3＝0.5。登录IP地址的变化中的异常条件可以被定义为L_Combined(L组合)>T，其中T是阈值。该阈值可以根据先前的数据确定和/或可以随时间被修改。

[0166] 算法2是可以被用于检测失败的登录IP地址变化的算法的示例。可以针对不同时间段内的登录IP地址特征向量计算Z得分，这里以一周、四周和八周作为示例来进行说明：

[0167] $$L1\ Z得分 = \frac{过去\ 24\ 小时失败的登录IP - 1\ 周Avg失败的登录IP}{1\ 周Stddev失败的登录IP}$$

[0168] $$L2\ Z得分 = \frac{过去\ 24\ 小时失败的登录IP - 4\ 周Avg失败的登录IP}{4\ 周Stddev失败的登录IP}$$

[0169] $$L3\ Z得分 = \frac{过去\ 24\ 小时失败的登录IP - 8\ 周wk\ Avg失败的登录IP}{8\ 周Stddev\ 失败的登录IP}$$

[0170] 失败的登录IP地址的Z得分可以与指派给每个得分的权重(w1...w3)组合：

[0171] L组合＝(w1×L1Z得分)+(w2×L2Z得分)+(w3×L3Z得分)

[0172] 在各种示例中，可以取决于何时执行计算来动态地计算所应用的权重。例如，在第一天，可以使用基于现有数据计算的值来应用默认基准，包括默认Avg(平均值)和默认Stddev(标准偏差)。在这个示例中，权重可以随着周的进展而变化，如算法1的示例中所提供的那样。登录IP地址的变化中的异常条件可以被定义为L_Combined(L组合)>T，其中T是阈值。该阈值可以根据先前的数据确定和/或可以随时间被修改。

[0173] 在各种示例中，针对一个云服务的用户检测到的异常活动可以被威胁检测引擎302用于计算或重新计算在使用另一个云服务时的威胁的可能性。以这种方式，可以主动地筛选在使用一个云服务期间发生的新事件，以检测和/或预测在使用另一个云服务时的威胁。在各种示例中，可以将跨不同云服务的多个数据点关联起来，以提高威胁得分的准确性。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00425

[0174] 算法3提供了可以被用于分析多个应用行为的算法的示例。在算法3中，将与各种云服务活动(诸如登录)相关联的用户IP地址解析为地理位置坐标IP1(纬度1，经度1)、IP2(纬度2，经度2)、IP3(纬度3，经度3)等。如果用户对于不同的云服务具有不同的用户名，那么可以将与该用户相关联的各个用户名映射到唯一的特定于用户的身份，该身份跨服务识别该用户。可以使用各种距离量度和/或公式中的任何一个计算针对任何数量的云服务(例如，Amazon Web Services、Box、Salesforce等)的用于登录(例如，登录尝试、成功登录和/或失败的登录)的任何两个IP地址之间的距离。在一些示例中，距离d是使用Haversine距离公式计算的，如下所示：

[0175] $Diff_{经度}$=经度2-经度1

[0176] $Diff_{纬度}$=纬度2-纬度1

[0177] $$a=(sin\left(\frac{Diff_{纬度}}{2}\right))^2+(cos(纬度1)\times cos(纬度2)\times(sin\left(\frac{Diff_{经度}}{2}\right))^2$$

[0178] $$c=2\times atan2(\sqrt{a},\sqrt{1-a})$$

[0179] d=R×c

[0180] 在用于d的等式中，R是地球的半径。

[0181] 可以使用如上计算的最大距离来计算Z得分以确定不同时间段内用户行为的偏差。作为示例，示出了1周、4周和8周的时间段：

[0182] $$L1\ Z得分=\frac{过去\ 24\ 小时最大距离IP登录-1\ 周Avg(每天最大距离IP登录)}{1\ 周Stddev(每天IP登录之间的最大距离)}$$

[0183] $$L2\ Z得分=\frac{过去\ 24\ 小时最大距离IP登录-4\ 周Avg(每天最大距离IP登录)}{4\ 周Stddev(每天IP登录之间的最大距离)}$$

[0184] $$L3\ Z得分=\frac{过去\ 24\ 小时最大距离IP登录-8\ 周Avg(每天最大距离IP登录)}{8\ 周Stddev(每天IP登录之间的最大距离)}$$

[0185] Z得分可以与指派给每个得分的权重(w1...w3)组合，如下：

[0186] L组合=(w1×L1Z得分)+(w2×L2Z得分)+(w3×L3Z得分)

[0187] 在各种示例中，可以取决于何时执行计算来动态地计算所应用的权重。例如，在第一天，可以使用基于现有数据计算的值来应用默认基线，包括默认Avg(平均值)和默认Stddev(标准偏差)。在这个示例中，如上面所提供的，权重可以随时间的进展而变化。登录IP地址的变化中的异常条件可以被定义为L_Combined(L组合)>T，其中T是阈值。该阈值可以从先前的数据确定和/或可以随时间被修改。

[0188] 算法4提供了用于确定访问云应用时使用的浏览器或操作系统(OS)的变化的算法的示例。可以使用特征向量来计算Z得分，该特征向量表示当访问云应用时使用的不同浏览器或操作系统的数量的计数，其中访问发生在各个时间段。作为示例，下面使用一周、四周和八周：

[0189] $$L1\ Z得分=\frac{\{浏览器,\ OS\}过去\ 24\ 小时计数-1\ 周Avg(\{浏览器,OS\}每天计数)}{1\ 周Stddev(\{浏览器,OS\}每天计数)}$$

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00426

[0190] $$L2\,Z得分 = \frac{\{浏览器，OS\}过去\,24\,小时计数 - 4\,周Avg\,(\{浏览器,OS\}每天计数)}{4\,周Stddev\,(\{浏览器,OS\}每天计数)}$$

[0191] $$L3\,Z得分 = \frac{\{浏览器，OS\}过去\,24\,小时计数 - 8\,周Avg\,(\{浏览器,OS\}每天计数)}{8\,周Stddev\,(\{浏览器,OS\}每天计数)}$$

[0192] 可以将Z得分与指派给每个得分的权重(w1...w3)组合，如下：

[0193] L组合＝(w1×L1Z得分)+(w2×L2Z得分)+(w3×L3Z得分)

[0194] 在各种示例中，可以初始地应用默认基线，并且随着时间的进展，权重可以随着更多数据变得可用而变化。登录IP地址的变化中的异常条件可以被定义为L_Combined(L组合)>T，其中T是阈值。该阈值可以根据先前的数据确定和/或可以随时间被修改。

[0195] 算法5提供了用于确定从云应用下载的数量的变化的算法的示例。可以使用特征向量来计算Z得分，该特征向量表示用户账户在不同时间段内的下载次数计数，诸如以下示例中提供的一周、四周和八周：

[0196] $$L1\,Z得分 = \frac{过去\,24\,小时的下载计数 - 1\,周Avg\,(每天下载计数)}{1\,周Stddev\,(每天下载计数)}$$

[0197] $$L2\,Z得分 = \frac{过去\,24\,小时的下载计数 - 4\,周Avg\,(每天下载计数)}{4\,周Stddev\,(每天下载计数)}$$

[0198] $$L3\,Z得分 = \frac{过去\,24\,小时的下载计数 - 8\,周Avg\,(每天下载计数)}{8\,周Stddev\,(每天下载计数)}$$

[0199] 可以将Z得分与指派给每个得分的权重(w1...w3)组合，如下：

[0200] L组合＝(w1×L1Z得分)+(w2×L2Z得分)+(w3×L3Z得分)

[0201] 在各种示例中，可以初始地应用默认基线，并且随着时间的进展，权重可以随着更多数据变得可用而变化。登录IP地址的变化中的异常条件可以被定义为L_Combined(L组合)>T，其中T是阈值。该阈值可以根据先前的数据确定和/或可以随时间被修改。

[0202] 在各种示例中，可以使用诸如上面说明的得分之类的得分，以及其它指标，来计算风险得分，该风险得分在本文中也称为安全性的量度。在各种示例中，威胁检测引擎302可以计算用户、用户组或类别、服务和/或服务提供商的风险得分。风险得分可以指示安全性风险的程度。例如，可以定义从一到五的标度，其中越高的值指示用户或服务对组织构成越高的安全性风险。

[0203] 用于计算风险得分的指标也可以以得分的形式提供特定的风险因素。例如，异常检测的结果可以包括得分形式的指标，该指标指示与规范的偏离程度和/或异常对组织造成的风险程度。在一些示例中，与同一用户或同一服务相关联的每个异常可以被用作单独的指标。在各种示例中，可以被用于计算风险得分的其它指标可以与用户、服务、服务提供商、用户看上去所在的地理位置、用户看上去所在的域、一天中的时间或一周中的某天或一年中的时间或另一个因素相关联。例如，可以从与用户相关联的组织、从声誉站点、从社交媒体站点、从新闻组织或从其它源获得针对用户的指标。可以例如从威胁情报聚合器或分发器获得针对服务或服务提供商的指标，威胁情报聚合器或分发器可以跟踪服务或服务提供商的声誉。其它指标可以由内部威胁情报数据314提供。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00427

[0204] 在各种示例中，可以将风险得分计算为可用指标的加权和。例如，给定指标“I_1，I_2，...，I_n”，可以使用以下公式计算风险得分：

[0205] $$\text{风险得分} = \frac{I_1W_1 + I_2W_2 + \cdots + I_nW_n}{W_1 + W_2 + \cdots + W_n}$$

[0206] 在前面的等式中，“W_1，W_2，…W_n”是权重。在各种示例中，权重值可以指示指标的相对重要性，越不重要的指标接收越低的权重值。

[0207] 在一些示例中，分析引擎300可以获得关于风险得分的有效性和/或准确性的反馈。作为示例，组织的网络管理员可以提供反馈。作为另一个示例，安全性管理和控制系统的管理员可以提供反馈。可替代地或附加地，在一些示例中，可以使用自动机器学习算法(诸如决策树和神经网络)来获得反馈。在一些示例中，分析引擎300可以使用反馈来调整权重、指标和/或源，包括可能去除源或指标。在这些和其它示例中，威胁检测引擎302可以用经调整的指标和权重来计算新的风险得分。

[0208] 在各种示例中，威胁检测引擎302可以对用于计算风险得分的每个指标和/或风险得分执行回归分析。回归分析可以包括构建和更新线性回归模型。线性回归模型可以提供输出，诸如$S = c_1(I_1) + c_2(I_2) + \cdots + c_n(I_n)$。由回归模型计算出的系数$c_i$可以是新的或经修改的权重，它们将代替初始权重来计算风险得分。随着更多反馈和更多数据被收集，该模型可以提供更高的准确性。

[0209] 在各种示例中，分析引擎300可以对从服务提供商获得的活动数据310执行各种其它分析306。在一些示例中，各种类型的算法对于分析数据可以特别有用。决策树、时间序列、朴素贝叶斯分析和用于构建用户行为简档的技术是可以被用于基于可疑活动和/或外部数据馈送的模式来生成预测的机器学习技术的示例。诸如聚类之类的技术可以被用于检测离群值和异常活动。例如，可以基于帐户访问一个或多个文件或使来自(由第三方馈送或以其它方式)标记为恶意的IP地址的一系列登录尝试失败而识别威胁。以类似的方式，威胁也可以(可能随时间)基于一个云应用或跨多个云应用的不同活动模式。

[0210] 可以被生成的一类分析是描述性分析或统计分析。可以使用预定义的系统查询集(诸如例如MapReduce作业以及Spark和Apache Hive查询等等)来生成统计数据。可以使用关联技术为单个服务或者跨多个服务生成描述性分析。可以被生成的报告的示例包括例如登录统计信息(例如，登录失败最多的用户、基于IP地址的登录历史(包括对IP声誉、地理位置和其它因素的考虑))、用户统计信息(例如，具有最多资源[文件、EC2机器等]的用户、跨云的权利、改变的密码的数量等)、活动统计信息(例如，用户跨云的活动)、密钥轮换的统计信息(例如，是否在最近三十天内已轮换安全外壳(SSH)密钥，等等)，以及资源统计信息(例如，文件夹的数量、由用户下载的文件、由漫游或移动用户下载的文件，等等)，等等。可以使用各种统计分析技术，诸如均值、标准偏差、回归、样本尺寸确定和假设测试，等等。可以识别趋势，诸如在一定时间段内的登录活动、基于密码相关的支持问题的过去历史的此类问题，或识别在一定时间段内看到最多活动的移动设备的类型。报告中的数据可以显示在作为事件查看器的用户界面上，该事件查看器示出事件的“墙”，以及用户可以响应于或为了补救事件而采取的动作。可以基于预定义的规则构造警报，这些规则可以包括具体的事件和阈值。

[0211] 可以被生成的另一类分析是预测和启发式分析。这些可以结合机器学习算法以生

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00428

成威胁模型，诸如例如与基线期望的偏差、罕见和不频繁的事件、以及用于导出用户的可疑行为的行为分析，等等。可以训练算法和简档，以智能地预测不寻常行为是否为安全性风险。可以集成来自诸如但不限于MaxMind、FireEye、Qualys、Mandiant、AlienVault和Norse STIX之类的提供商的第三方馈送，以增强威胁情报。这些第三方馈送可以提供关于潜在安全性威胁并与之相关的外部信息，诸如例如IP地址声誉、恶意软件、被感染的节点的标识、脆弱的web浏览器版本、用户对代理或虚拟专用网络(VPN)服务器的使用，以及对云的已知攻击。在一些示例中，威胁信息以结构化威胁信息表达(STIX)数据格式表示。例如，一个或多个服务可以贡献有关特定IP地址的信息，诸如声誉(例如，已知具有软件漏洞、恶意软件的主机，或攻击源)和/或与IP地址相关联的地理位置。可以将这个信息与检索到的涉及IP地址的活动数据(诸如什么时间从那个IP地址尝试登录)以及从活动数据导出的信息(诸如登录尝试距离多远)组合。这些因素可以被用于确定“登录速度”度量。可以为其它活动(诸如文件访问、销售交易或虚拟机的实例)确定度量。

[0212] 聚类和回归算法可以被用于对数据进行分类并找到常见的模式。例如，聚类算法可以通过聚合从移动设备登录的用户的所有条目而将数据放入集群中。预测分析还可以包括基于活动来识别威胁，其中活动是诸如用户在几个月内没有访问特定的云应用并然后在下个月表现出高活动性，或者用户在过去几周中每周下载一个文件，这表明潜在的高级持续威胁(APT)场景。在一些示例中，随时间收集的数据被用于构建正常行为的模型(例如，事件和活动的模式)，并将偏离正常的行为标记为异常行为。在将一个或多个标记的事件或活动表征为真阳性或假阳性(例如，通过用户反馈)之后，可以将该信息提供回一个或多个机器学习算法，以自动修改系统的参数。因此，可以至少以上述方式利用机器学习算法来做出推荐并减少假警报(假阳性)。在一段时间内从各种参数收集的活动数据可以与机器学习算法一起使用，以生成被称为用户行为简档的模式。活动数据可以包括上下文信息，诸如IP地址和地理位置。

[0213] 在各种实施方式中，算法可以使用先前获取的用户活动数据来模拟正常用户活动。例如，租户基线317可以包括用户过去使用云服务的记录。该模拟可以被用于训练其它机器学习算法以学习组织的用户的正常行为。一般而言，特定的安全性问题可能不总是重复，并且因此可能无法通过纯受监督算法被检测到。但是，诸如离群值检测之类的技术可以建立对于检测异常活动有用的基线。此类异常活动以及上下文威胁情报可以以较低的预测错误提供对威胁的更准确的预测。

[0214] 在各种实施方式中，其它分析306可以包括对安全性控制漂移的检测，安全性控制漂移可以是指以看似任意的方式改变一个或多个安全性控制，这会增加安全性风险。在一些示例中，可以响应于云服务的安全性控制的变化以及与风险事件相关联的可动作情报来生成风险事件。威胁可以包括对于应用的使用而言是异常或不合规的活动、事件或安全性控制。作为示例，云应用中租户的密码策略可能已经被改变为施加更少的要求(例如，字符的类型和/或数量)。这可以生成风险事件和警报，以推荐将密码策略改变回原始密码策略。

[0215] 在各种实施方式中，分析引擎300可以包括推荐引擎308，推荐引擎308接收威胁检测引擎302、行为分析引擎304和其它分析306的输出。在各种示例中，推荐引擎308可以引发警报322、做出推荐324、自动执行动作326以及提供可视化328，组织可以使用该可视化328来理解组织对云服务的使用、检测到的安全性风险以及安全性风险的补救，等等。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00429

[0216] 在各种示例中，可以在可视化328中提供警报322，可视化328可以使用组织可访问的用户界面来查看。可替代地或附加地，可以通过其它通信信道来提供警报322，诸如电子邮件、文本消息、短消息服务(SMS)消息、语音邮件或另一种通信方法。在一些示例中，警报322可以作为安全消息被传送(例如，通过安全通信信道或要求密钥或登录凭证以查看)。

[0217] 警报可以包括关于检测到的事件的信息，诸如例如事件标识符、日期、时间、风险级别、事件类别、用户账户和/或与事件相关联的安全性控制、与事件相关联的服务、事件的描述、补救类型(例如，手动或自动)和/或事件状况(例如，打开、关闭)，等等信息。警报中关于每个风险事件的信息可以包括例如受影响的云服务的标识符或类别的实例、优先级、日期和时间、描述、推荐的补救类型和/或状况等信息。风险事件还可以具有用户可选择的动作，诸如编辑、删除、标记状况完成和/或执行补救动作。补救动作的选择可以调用诸如事件补救应用和/或云播种器应用之类的应用以执行所选择的补救。可以将有关识别出的威胁的警报和/或其它信息发送到安全性监视和控制系统外部的实体。

[0218] 在一些示例中，可以随时间提供不同事件类别中的事件的计数作为图形可视化，诸如图表。该图表可以显示例如颜色编码的类别中的每一个类别的按日期的事件(诸如在不寻常时间的活动、下班后的下载、失败的登录等)的计数。事件类别的视觉表示(例如，线)可以被打开和关闭。在一些示例中，威胁也可以显示在摘要视图中。

[0219] 在一些示例中，当组织的网络管理员接收到警报322时，网络管理员可以从组织的网络内采取补救动作。在这些示例中，安全性管理和控制系统可以将警报保持在“打开”状态，直到网络管理员报告可以关闭警报。

[0220] 在各种示例中，当威胁检测引擎302或其它分析306标记需要关注的事件时，推荐引擎308还可以确定推荐324。推荐可以包括可以被采取以进一步调查可疑事件或补救可疑事件(例如，对其采取纠正动作)的动作。在各种示例中，可以在用户界面中呈现的可视化328中向组织的网络管理员呈现推荐324。可替代地或附加地，推荐324可以通过其它形式的通信来呈现，诸如电子邮件、文本消息、短消息服务(SMS)消息、语音邮件等等。在各种示例中，组织的网络管理员可以选择激活推荐的动作，这可以使安全性管理和控制系统执行动作。

[0221] 在各种示例中，推荐引擎308可以使用关联规则学习来生成推荐。在一些示例中，推荐引擎308可以使用简档链接算法以通过找到跨服务相关性来链接跨多个云应用的活动。可以使用一个或多个属性或标识因素(诸如在云中普遍使用的主要用户标识符或单一登录(SSO)认证机制(例如，活动目录、Okta等))跨多个云服务识别单个用户。跨应用的活动的相关性的示例发现用户从不同的IP地址同时登录到两个云服务中、发现执行若干次失败登录尝试并随后改变用户的密码的用户，以及经常对于两个或多个云服务有多次失败登录的用户，等等示例。

[0222] 在各种示例中，推荐引擎308还可以确定动作326，包括安全性管理和控制系统将自动执行的补救动作。在各种示例中，组织可以被配置为在分析引擎300检测到某些安全性事件时自动执行补救动作。补救动作的示例尤其包括停用账户、重置密码或设置更强的安全性控制，等等。在这些和其它示例中，补救动作可以包括修改受安全性事件影响的服务或另一个服务的安全设置。在后一种情况下，当检测到安全性事件时，分析引擎300可以确定该另一个服务可能受到影响，或者应该以其它方式得到保护。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00430

[0223] 在一些示例中，组织可以使用第三方事件管理自动化系统，诸如例如ServiceNow或IBM QRadar等等。在这些示例中，安全性管理和控制系统可以能够与第三方事件管理系统接口以补救安全性事件。例如，事件管理系统可以具有API，通过该API，安全性管理和控制系统可以与事件管理系统进行交互。在这个和其它示例中，由推荐引擎308确定的动作可以包括向事件管理系统发送关于安全性事件的警报和/或其它信息，事件管理系统可以跟踪并且还可能补救该事件。事件管理系统可以将状况(例如，完成或未完成)返回给安全性管理和控制系统。以这种方式，可以将补救委派给外部系统，并将结果报告回安全性管理和控制系统以“闭合环路”。例如，如果期望为用户账户重置密码，那么动作可以包括向组织的管理用户账户的内部IT系统发送警报或消息。管理员或系统可以完成密码重置操作，并将完成状况报告回云安全性系统。

[0224] 在各种示例中，云服务可以定义将服务的管理性用户与普通用户区分开的特权。通过拥有更大或更多特权，管理性用户可以为组织配置和管理云服务，而普通用户只能使用云服务。

[0225] 如以上所讨论的，识别云服务的特权用户可能不是直接的。例如，每个云服务可以具有特权的不同定义，这些定义将一个用户限定为管理性用户而另一个用户是普通用户。作为另一个示例，云服务的管理性用户可以随时间而改变，新的管理性用户是以不受组织控制的方式和时间创建的。作为另一个示例，当组织的员工离职或改变工作职责时，关于哪些用户账户具有特权的知识可能会丢失。

[0226] 在各种实施方式中，安全性管理和控制系统可以实施用于识别云服务的特权用户的技术。然后，可以使用诸如上述方法来跟踪特权用户的活动，这些方法包括行为分析、异常检测和机器学习技术，同时应用了与如果特权账户被滥用则会造成的更大伤害相称的更高的审查程度。

[0227] 在各种实施方式中，特权用户的识别可以是安全性管理和控制系统的分析引擎的部件。图4图示了行为分析引擎404的框图，该行为分析引擎404是可以实现特权用户的识别的分析引擎的一个部件的示例。在各种示例中，行为分析引擎404可以接收活动数据410，该活动数据410可以包括针对一个云服务或多个云服务的用户活动的记录。活动数据410可以包括例如执行的动作的列表、执行动作的用户、受(一个或多个)动作影响的一个或多个对象、诸如时间戳之类的上下文数据和/或用户执行动作的网络位置，等等。

[0228] 在各种示例中，行为分析引擎404可以包括统计信息分析引擎432，该统计信息分析引擎432可以对活动数据410进行统计分析。在各种示例中，统计信息分析引擎432可以输出行为模型442，该行为模型442可以描述组织的用户使用云服务或多个云服务的方式。例如，统计信息分析引擎432可以输出描述特定用户对云服务的使用、由一组用户对云服务的使用和/或由组织中的所有用户对云服务的使用的模型。作为另一个示例，统计信息分析引擎432可以输出描述组织的用户使用云服务的方式的模型和/或描述组织的一个或多个用户使用多个云服务的方式的模型。在各种示例中，统计信息分析引擎432可以输出描述用户、用户在使用云服务时的动作和/或使用云服务的方式的其它模型。

[0229] 在各种示例中，行为分析引擎404还可以包括特权用户识别引擎434，该特权用户识别引擎434可以生成云服务的特权用户444的列表。在各种示例中，特权用户识别引擎434可以使用可能结合使用的不同技术来识别云服务的特权用户。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00431

[0230] 作为第一示例，特权用户识别引擎434可以从租户基线417学习特权用户的身份，该租户基线可以包括作为云服务的管理性用户的用户的用户名的列表。在各种示例中，组织可以具有云服务的管理性用户的列表。例如，当组织注册诸如Salesforce之类的服务时，组织可以向Salesforce指定哪些用户将成为管理性用户。在一些情况下，组织可以改变服务的管理性用户，并且可以向安全性管理和控制系统提供管理性用户的更新后的列表。但是，在许多情况下，组织可能没有良好的流程(process)来跟踪云服务的管理性用户，因此可能无法在租户基线417中提供准确的列表。

[0231] 作为第二示例，特权用户识别引擎434可以从自服务提供商获得的服务提供商数据418中学习特权用户的身份。服务提供商具有可以包括使安全性管理和控制系统能够请求服务的特权用户列表的功能的API。在各种示例中，安全性管理和控制系统可以周期性地做出这个请求，以应对服务的特权用户发生改变的情况。但是，许多云服务不能提供这个功能。可替代地或附加地，云服务自己对特权用户的定义可能太狭窄，并且可能排除了会影响该服务的其他用户或该服务的操作的操作。

[0232] 在各种实施方式中，除了依赖租户基线417和/或服务提供商数据418，或作为其替代，特权用户识别引擎434可以使用学习技术从来自云服务的活动数据410中确定特权用户。在这些实施方式中，特权用户识别引擎434可以包括受监督学习引擎436和/或无监督学习引擎438。

[0233] 在各种实施方式中，受监督学习引擎436可以从组织和/或从安全性管理和控制系统的操作者接收训练数据。在各种示例中，训练数据可以包括可以对云服务执行的动作的一些或全部，以及用于指示动作是否具有特权的动作的标签。在一些示例中，标签可以指示特权的不同类别。例如，特权的集合可以是管理性特权，而特权的不同集合是共同管理性特权。在各种示例中，受监督学习引擎436可以使用训练数据来训练分类模型，诸如神经网络或随机森林或另一种分类模型。

[0234] 在各种示例中，受监督学习引擎436可以将分类模型应用于来自服务的活动数据410。一般而言，分类模型是从与来自同一云服务的活动数据对应的训练数据生成的(例如，训练数据是从可以使用该云服务执行的动作开发的)。使用分类模型，受监督学习引擎436可以例如在活动日志中提取涉及特权动作的事件。因为活动日志可以列出每个事件的用户，所以受监督学习引擎436可以确定发起了事件的用户可能是特权用户。在各种示例中，除了动作之外或作为其补充的准则可以被用于确定事件是由特权用户实施的。例如，事件的特权性质可以通过受动作影响的资源、动作影响资源的方式、执行动作的次数、执行动作的时间、发起事件的IP地址以及其它准则来指示。

[0235] 在各种示例中，可以通过获得来自组织和/或来自安全性管理和控制系统的操作者的反馈来验证并提高由受监督学习引擎436做出的确定的准确性。例如，行为分析引擎404可以输出由受监督学习引擎436识别的特权用户444的列表，以及特权用户执行的将这些用户识别为具有特权的动作。审阅者可以向安全性管理和控制系统指示用户和/或将用户识别为具有特权的动作是否正确。这个信息可以被反馈回受监督学习引擎436，该受监督学习引擎436可以更新适当的分类模型。

[0236] 在各种实施方式中，无监督学习引擎438可以尝试学习具有特权的动作，并从这种学习中识别特权用户。无监督学习引擎438可以例如解析活动数据410中的事件并将事件的

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00432

组成部分存储为特征集。例如，无监督学习引擎438可以使用具有诸如“动作”、“动作类型”、“用户ID”、“时间戳”、“受影响的资源”和针对事件的其它组成部分的其它字段之类的字段的数据结构来表示特征集。作为另一个示例，无监督学习引擎438可以每天、每周或针对不同的时间段聚合事件，并且以这种方式搜集用于云服务的大量历史事件数据。在各种示例中，无监督学习引擎438可以将各种无监督学习技术(诸如异常检测和神经网络等)应用于累积特征集。

[0237] 作为示例，K均值集群可以被应用于特征集以识别活动数据410中的集群，其中集群可以将特权用户和普通用户分组在一起。在一些示例中，无监督学习引擎438可以被提供有具有特权的动作的列表，或者应当仅由特权用户访问的资源的列表，或者无监督学习引擎438可以随后用来识别特权用户的集群的其它准则。可替代地或附加地，无监督学习引擎438可以被呈现给安全性管理和控制系统的操作者，该操作者然后可以识别特权用户的集群，并且从这些集群中定义描述特权用户的特点的模型。

[0238] 下面的表6说明了由一组用户每天执行的动作的平均计数的示例。在这个示例中，动作包括文件共享以及文件共享应用中的创建用户动作。

[0239] 表6

[0240]

用户	文件共享事件#	创建用户事件#
John	3	0
Mary	6	0
Bill	2	0
Kumar	4	0
Chi	0	0
Jose	11	0
Colin	9	0
Sean	20	1
Mike	19	1
Steve	17	1
Alexi	18	3
Samantha	20	3
Josh	16	2
Jenny	21	2

[0241] 图5图示了表6中的数据的曲线图500的示例。横轴502映射文件共享事件的计数，而纵轴映射创建用户事件的计数。根据表6的数据，一组用户仅执行文件共享事件，因此用于这些用户的数据点都沿着一条线落在零创建用户事件处。第二组用户同时执行文件共享事件和创建用户事件二者，因此用于这些用户的数据点在水平和垂直方向上都大于零。

[0242] 如图5的示例中所示，两个用户集群是明显的。第一个集群(包括用户Sean、Mike、Steve、Alexi、Samantha、Josh和Jenny)由于执行了相同的动作而形成一组用户。第二个集群(包括用户John、Mary、Bill、Kumar、Chi、Jose和Colin)也由于执行了相同的动作而形成第二组用户。在这个示例中，由于执行了创建用户事件，因此可以将第一组中的用户识别为特权用户。在各种示例中，可以从被认为有特权的活动的描述中和/或由负责分析聚类工具

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00433

的输出的操作者来确定将第一集群识别为一组特权用户。

[0243] 在各种示例中，仅动作可能不指示执行了该动作的用户是否是特权用户。例如，在文件共享服务中，一些文件夹可以用诸如“机密”之类的关键词标记，而其它文件夹则具有其它标签或没有标签。在这个示例中，涉及被标记为“机密”的文件夹的一些或所有事件可以是特权事件。例如，读取标记为“机密”的文件夹中的文件的任何用户都可以被视为特权用户。作为另一个示例，只有可以在标记为“机密”的文件夹中添加新文件或删除文件的用户才可以被视为特权用户。作为另一个示例，只曾经访问具有不同标签或没有标签的文件夹的用户可以被视为普通的非特权用户。

[0244] 图6包括图示用于确定云服务的特权用户并管理特权用户的活动可能造成的安全性风险的处理600的示例的流程图。在各种示例中，示例处理600可以由安全性管理系统的计算系统执行。在各种示例中，处理600可以被实施为存储在非暂态计算机可读介质上的指令，该指令可以由计算系统的一个或多个处理器执行以执行处理600的步骤。

[0245] 在步骤602处，处理600包括从服务提供商系统获得活动数据，其中活动数据描述在使用云服务期间执行的动作，其中动作由与租户相关联的一个或多个用户执行，其中服务提供商系统向租户提供租户账户，并且其中租户账户使一个或多个用户能够访问云服务。在一些示例中，租户可以是将人和资源聚集在一起以实现共同目的的组织。组织的示例包括公司、大学、医院、政府机构以及其它人和资源的组。在一些示例中，租户可以是使用网络服务的个人。在各种示例中，组织或个人可以订阅服务提供商的服务，并且可以通过订阅而获得由服务提供商提供的服务的使用。云服务的示例包括基础设施、平台、网络和软件应用等。

[0246] 在各种示例中，在订阅后，向租户提供使该租户能够访问该服务的租户账户。在一些示例中，可以向与租户相关联的用户给予针对服务的各个用户账户，这使每个用户都能够使用该服务。在这些和其它示例中，随着用户使用服务，服务提供商可以记录由用户执行的动作。动作可以包括例如登录服务、登出、上传数据、下载数据、修改数据、启动可执行文件以及其它操作。在各种示例中，这些动作可以与信息(诸如执行动作的用户的标识、发起和/或完成动作的时间、受动作影响的资源，和/或用户从其执行动作的网络位置或地理位置，等等)一起记录在活动数据中。

[0247] 在步骤604处，处理600包括在活动数据中识别相对于云服务具有特权的一个或多个动作。特权动作是使用户能够以不是所有用户都被允许执行的方式来影响资源的动作。例如，特权动作可以是在由第一用户执行时可以以影响其他用户对云服务的使用的方式来修改云服务的动作。作为另一个示例，特权动作可以是在由第一用户执行时可以影响云服务的其他用户的用户账户的动作。

[0248] 在一些示例中，可以使用与云服务相关联的动作的列表来识别具有特权的一个或多个动作，其中该列表中的动作相对于云服务被分类为具有特权。例如，对于一个云服务，用于创建用户账户和删除用户账户的动作可以被分类为具有特权。作为另一个示例，对于另一个云服务，可以将用于修改某些文件夹的动作分类为具有特权。

[0249] 在一些示例中，可以使用管理性动作的列表来识别具有特权的一个或多个动作。在这些示例中，管理性动作的列表可以包括已知要求更高特权的动作，诸如创建或删除用户账户、修改访问控制或安全性设置、修改凭证等等。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00434

[0250] 在步骤606处，处理600包括使用活动数据识别执行了该一个或多个动作的用户集，其中该用户集是从与租户相关联的该一个或多个用户确定的。

[0251] 在一些示例中，识别用户集可以包括使用一个或多个动作和过去的活动数据来生成模型。在这些示例中，模型可以描述云服务的、相对于云服务具有特权的使用模式。例如，受监督或无监督学习可以被用于训练神经网络，以识别相对于特定云服务具有特权的动作或一系列动作。在受监督学习示例中，可以为神经网络提供带标签的训练数据，该数据识别具有特权或不具有特权的动作。在无监督学习示例中，神经网络可以被配置为最小化成本函数，其中成本函数对云服务的改变进行建模。在这些和其它示例中，该模型可以被用于识别用户集。

[0252] 可替代地或附加地，识别用户集可以包括对在使用云服务期间执行的动作进行分组，以及识别包括具有特权的动作的一组动作。例如，可以使用K均值聚类技术绘制活动数据中的动作以及执行了动作的用户，以识别执行类似动作的用户。从图中可以识别出用户组。

[0253] 在步骤608处，处理600包括将用户集分类为具有特权。在各种示例中，由于执行了被识别为具有特权的一个或多个动作，用户集是具有特权的。

[0254] 在步骤610处，处理600包括使用活动数据确定一个或多个用户的一个或多个风险得分。在各种示例中，风险得分指示用户在使用云服务时执行的动作对租户造成的安全性风险的程度。可以为单个用户、用户组、单个服务、多个服务、来自不同服务提供商的服务和/或用户和服务的组合计算风险得分。在一些示例中，风险得分被计算为风险指标的加权和。风险指标可以与执行的动作、受动作影响的资源、特定用户、一组用户、服务、服务提供商、用户所在的网络位置、服务提供商或服务所在的网络位置、地理位置、一天中的时间或一周中的某天或一年中的某月、另一个因素或因素的组合相关联。可替代地或附加地，风险指标可以来自从网络威胁情报的外部聚合器和/或分发器获得的威胁情报。在一些示例中，越重要的风险指标被赋予越高的权重值。

[0255] 在各种示例中，与非特权用户的风险得分相比，以更大的权重来计算被分类为具有特权的用户的风险得分。因此，例如，当特权用户执行动作时，特权用户的风险得分可以比非特权用户执行相同动作时的风险得分更高。可替代地或附加地，与普通用户相比，可以使用更多或不同的指标来确定特权用户的风险得分。可替代地或附加地，与非特权用户的风险得分相比，可以更频繁地计算特权用户的风险得分，或者以其它方式给予更高的审查程度。

[0256] 在步骤612处，处理600包括确定用户集中的用户的风险得分大于阈值。在各种示例中，阈值可以指示在超过阈值时构成对租户的安全性风险的活动。在各种示例中，阈值可以与特定租户、特定用户、一组用户、特定服务或服务提供商、一天中的时间或一周中的某天、另一因素、或因素的组合相关联。

[0257] 在步骤614处，处理600包括确定用于服务提供商系统的安全性控制，其中该安全性控制被服务提供商系统用来配置对云服务的访问。安全性控制可以取决于诸如以下因素之类的因素：使风险得分超过阈值的执行了的一个或多个动作、受动作影响的资源、威胁情报和/或与租户相关联的配置设置，等等因素。可以被确定的安全性控制的示例包括阻止用户或用户组使用服务、禁用针对云服务的特定用户账户、使服务在某些用户登录和/或执行

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00435

某些动作时发送警报，以及阻止数据上传到服务或从服务下载，等等。

[0258] 在步骤616处，处理600包括确定要发送到服务提供商系统的一个或多个指令。在各种示例中，可以通过查询服务提供商以请求适当的指令以完成期望的配置来确定这些指令。在这些示例中，计算系统可以包括用于执行查询的自动化程序。作为另一个示例，可以从云服务的API确定指令，其中该API可以包括使计算系统能够修改云服务的配置的操作。

[0259] 在步骤618处，处理600包括将一个或多个指令发送到服务提供商系统，其中一个或多个指令使安全性控制相对于用户改变，其中用户对云服务的访问由于对安全性控制的改变而被修改。在各种示例中，发送指令可以包括使用指派给租户的授权(例如，密码、令牌或其它形式的凭证)，这可以使安全性管理系统能够代表租户配置云服务。在一些示例中，指令被发送到服务提供商系统以供租户或服务提供商激活。一旦被执行，就可以监视、减轻和/或停止由特权用户造成的安全性风险。

[0260] 图7绘出了在其中可以实现上面讨论的各种示例的分布式系统700的简化图。在所示示例中，分布式系统700包括一个或多个客户端计算设备702、704、706和708，其被配置为通过一个或多个网络710执行和操作客户端应用，诸如web浏览器、专有客户端(例如，Oracle Forms)等。服务器712可以经由网络710与远程客户端计算设备702、704、706、708通信地耦合。

[0261] 在各种示例中，服务器712可以适于运行一个或多个服务或软件应用。在一些示例中，服务器712还可以提供其它服务或软件应用，并且可以包括非虚拟和虚拟环境。在一些示例中，这些服务可以作为基于web的服务或云服务被提供，或者在软件即服务(SaaS)模型下被提供给客户端计算设备702、704、706、708的用户。操作客户端计算设备702、704、706、708的用户又可以利用一个或多个客户端应用来与服务器712交互以利用由这些部件提供的服务。

[0262] 在图7中绘出的配置中，系统700的软件部件718、720和722被示出为在服务器712上实现。在其它示例中，系统700的一个或多个部件和/或由这些部件提供的服务也可以由客户端计算设备702、704、706、708中的一个或多个来实现。然后，操作客户端计算设备的用户可以利用一个或多个客户端应用来使用由这些部件提供的服务。这些部件可以用硬件、固件、软件或其组合来实现。应该认识到的是，各种不同的系统配置是可能的，其可能与示例分布式系统700不同。图7中所示的示例因此是用于实现任意系统的分布式系统的一个示例，而不是要进行限制。

[0263] 客户端计算设备702、704、706、708可以包括各种类型的计算设备。例如，客户端计算设备可以包括便携式手持设备(例如，**iPhone®**、蜂窝电话、**iPad®**、计算平板电脑、个人数字助理(PDA))或可穿戴设备(例如，Google **Glass®**头戴式显示器)，运行诸如Microsoft Windows **Mobile®**和/或各种移动操作系统(诸如iOS、Windows Phone、Android、BlackBerry 10、Palm OS等)的软件。设备可以支持各种软件应用，诸如各种与互联网相关的应用、电子邮件、短消息服务(SMS)应用，并且可以使用各种其它通信协议。客户端计算设备还可以包括通用个人计算机，作为示例，包括运行各种版本的Microsoft **Windows®**、Apple **Macintosh®**和/或Linux操作系统的个人计算机和/或膝上型计算机。客户端计算设备可以是运行任何各种可商业获得的**UNIX®**或类UNIX操作系统(包

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00436

括但不限于各种GNU/Linux操作系统，诸如例如Google Chrome OS)的工作站计算机。作为示例，客户端计算设备还可以包括能够通过(一个或多个)网络710通信的任何其它电子设备，诸如瘦客户端计算机、启用互联网的游戏系统(例如，具有或不具有**Kinect®**手势输入设备的微软Xbox游戏控制台)和/或个人消息传送设备。

[0264] 虽然图7中示出的分布式系统700被示出为具有四个客户端计算设备，但是可以支持任何数量的客户端计算设备。其它设备(诸如具有传感器的设备等)可以与服务器712交互。

[0265] 分布式系统700中的(一个或多个)网络710可以是本领域技术人员熟悉的、可以利用任何各种可用的协议支持数据通信的任何类型的网络，其中协议包括但不限于TCP/IP(传输控制协议/网际协议)、SNA(系统网络体系架构)、IPX(互联网报文交换)、AppleTalk，等等。仅仅作为示例，(一个或多个)网络710可以是局域网(LAN)，基于以太网、令牌环的LAN、广域网、互联网、虚拟网络、虚拟专用网(VPN)、内联网、外联网、公共交换电话网(PSTN)、红外网络、无线网络(例如，依据电子电气学会(IEEE)802.11协议套件、**Bluetooth®**和/或任何其它无线协议当中任意一种操作的网络)；和/或这些和/或其它网络的任意组合。

[0266] 服务器712可以由一个或多个通用计算机、专用服务器计算机(作为示例，包括PC(个人计算机)服务器、**UNIX®**服务器、中档服务器、大型计算机、机架安装的服务器等)、服务器场、服务器集群或任何其它适当的布置和/或组合组成。服务器712可以包括运行虚拟操作系统的一个或多个虚拟机，或涉及虚拟化的其它计算体系架构。一个或多个灵活的逻辑存储设备池可以被虚拟化，以维护用于服务器的虚拟存储设备。虚拟网络可以由服务器712使用软件定义的联网来控制。在各种示例中，服务器712可以适于运行在前述公开中所描述的一个或多个服务或软件应用。例如，服务器712可以与用于执行以上描述的处理的服务器对应。

[0267] 服务器712可以运行包括以上讨论的操作系统当中任意一种的操作系统，以及任何可商业获得的服务器操作系统。服务器712还可以运行任何各种附加的服务器应用和/或中间层应用，包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(公共网关接口)服务器、**JAVA®**服务器、数据库服务器，等等。示例性数据库服务器包括但不限于从Oracle、Microsoft、Sybase、IBM(国际商业机器)等可商业获得的那些数据库服务器。

[0268] 在一些实现方式中，服务器712可以包括一个或多个应用，以分析和整合从客户端计算设备702、704、706、708的用户接收到的数据馈送和/或事件更新。作为示例，数据馈送和/或事件更新可以包括，但不限于，**Twitter®**馈送、**Facebook®**更新或者从一个或多个第三方信息源接收到的实时更新和连续数据流，其可以包括与传感器数据应用、金融报价机、网络性能测量工具(例如，网络监视和流量管理应用)、点击流分析工具、汽车交通监视等相关的实时事件。服务器712还可以包括一个或多个应用，以经由客户端计算设备702、704、706和708的一个或多个显示设备显示数据馈送和/或实时事件。

[0269] 分布式系统700还可以包括一个或多个数据库714、716。这些数据库可以提供用于存储信息的机制，其中信息诸如用户交互信息、使用模式信息、适应规则信息以及由上面讨论的各种示例使用的其它信息。数据库714、716可以驻留在各种位置。举例来说，数据库

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00437

714、716中的一个或多个可以驻留在服务器712本地(和/或驻留在服务器712中)的非暂态存储介质上。可替代地,数据库714、716可以远离服务器712并且经由基于网络的或专用的连接与服务器712通信。在一些示例中,数据库714、716可以驻留在存储区域网络(SAN)中。类似地,用于执行归属于服务器712的功能的任何必要文件可以适当地本地存储在服务器712上和/或远程存储。在一些示例中,数据库714、716可以包括关系数据库,诸如由Oracle提供的适于响应于SQL格式化的命令而存储、更新和检索数据的数据库。

[0270] 在一些示例中,云环境可以提供一个或多个服务。图8是系统环境800的一个或多个部件的简化框图,其中服务可以被提供为云。在图8所示的示例中,系统环境800包括一个或多个客户端计算设备804、806、808,用户可以使用它们来与提供云服务的云基础设施系统802进行交互。云基础设施系统802可以包括一个或多个计算机和/或服务器,这些计算机和/或服务器可以包括以上针对图7的服务器712描述的那些。

[0271] 应当认识到的是,图8中描绘的云基础设施系统802可以具有不同于所描绘的部件的其它部件。另外,图8中所示的示例仅仅是云基础设施系统的一个示例,该云基础设施系统可以结合上面讨论的各种示例。在一些示例中,云基础设施系统802可以具有比图中所示更多或更少的部件、可以组合两个或更多个部件,或者可以具有部件的不同配置或布置。

[0272] 客户端计算设备804、806、808可以是与以上针对客户端计算设备702、704、706、708描述的那些设备类似的设备。客户端计算设备804、806、808可以被配置为操作客户端应用,诸如web浏览器、专有客户端应用(例如,Oracle Forms)或可以被客户端计算设备的用户使用以与云基础设施系统802交互来使用由云基础设施系统802提供的服务的一些其它应用。虽然示例系统环境800被示为具有三个客户端计算设备,但是可以支持任何数量的客户端计算设备。诸如具有传感器的设备等的其它设备可以与云基础设施系统802交互。

[0273] (一个或多个)网络810可以促进客户端计算设备804、806、808与云基础设施系统802之间的通信和数据交换。每个网络可以是对本领域技术人员熟悉的可以利用任何各种商用的协议支持数据通信的任何类型的网络,其中协议包括以上针对图7的(一个或多个)网络710所描述的协议。

[0274] 在一些示例中,由图8的云基础设施系统802提供的服务可以包括按需对云基础设施系统的用户可用的服务的主机。还可以提供各种其它服务,包括但不限于在线数据存储和备份解决方案、基于Web的电子邮件服务、托管的办公套件和文档协作服务、数据库处理、受管理的技术支持服务等。由云基础设施系统提供的服务可以动态扩展,以满足其用户的需求。

[0275] 在一些示例中,由云基础设施系统802提供的服务的具体实例化在本文中可以被称为“服务实例”。一般而言,经由通信网络(诸如互联网)从云服务提供者的系统使得对用户可用的任何服务被称为“云服务”。通常,在公共云环境中,构成云服务提供者的系统的服务器和系统与消费者自己的本地服务器和系统不同。例如,云服务提供者的系统可以托管应用,并且用户可以经由诸如互联网的通信网络按需订购和使用应用。

[0276] 在一些示例中,计算机网络云基础设施中的服务可以包括对存储装置、托管的数据库、托管的web服务器、软件应用或者由云供应商向用户提供的其它服务的受保护的计算机网络访问,或者如本领域中另外已知的。例如,服务可以包括通过互联网对云上的远程存储的受密码保护的访问。作为另一个示例,服务可以包括基于web服务的托管的关系数据库

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00438

和脚本语言中间件引擎，用于由联网的开发人员私人使用。作为另一个示例，服务可以包括对在云供应商的网站上托管的电子邮件软件应用的访问。

[0277] 在一些示例中，云基础设施系统802可以包括以自助服务、基于订阅、弹性可扩展、可靠、高度可用和安全的方式交付给消费者的应用套件、中间件和数据库服务产品。这种云基础设施系统的示例是由本受让人提供的Oracle Public Cloud(Oracle公共云)。

[0278] 云基础设施系统802还可以提供与“大数据”相关的计算和分析服务。术语“大数据”一般用来指可由分析员和研究者存储和操纵以可视化大量数据、检测趋势和/或以其它方式与数据交互的极大数据集。这种大数据和相关应用可以在许多级别和不同规模上由基础设施系统托管和/或操纵。并行链接的数十个、数百个或数千个处理器可以作用于这种数据，以便呈现其或者模拟对数据或其所表示的内容的外力。这些数据集可以涉及结构化数据，诸如在数据库中组织或以其它方式根据结构化模型组织的数据，和/或者非结构化数据(例如，电子邮件、图像、数据blob(二进制大对象)、网页、复杂事件处理)。通过充分利用相对快速地将更多(或更少)的计算资源聚焦在目标上的能力，云基础设施系统可以更好地用于基于来自企业、政府机构、研究组织、私人个人、一群志同道合的个人或组织或其它实体的需求在大数据集上执行任务。

[0279] 在各种示例中，云基础设施系统802可以适于自动地供应、管理和跟踪消费者对由云基础设施系统802提供的服务的订阅。云基础设施系统802可以经由不同的部署模型提供云服务。例如，服务可以在公共云模型下提供，其中云基础设施系统802由销售云服务的组织拥有(例如，由Oracle公司拥有)并且使服务对一般公众或不同的工业企业可用。作为另一个示例，服务可以在私有云模型下提供，其中云基础设施系统802仅针对单个组织操作，并且可以为组织内的一个或多个实体提供服务。云服务还可以在社区云模型下提供，其中云基础设施系统802和由云基础设施系统802提供的服务由相关社区中的若干个组织共享。云服务还可以在混合云模型下提供，混合云模型是两个或更多个不同模型的组合。

[0280] 在一些示例中，由云基础设施系统802提供的服务可以包括在软件即服务(SaaS)类别、平台即服务(PaaS)类别、基础设施即服务(IaaS)类别、或包括混合服务的服务的其它类别下提供的一个或多个服务。消费者经由订阅订单可以订购由云基础设施系统802提供的一个或多个服务。云基础设施系统802然后执行处理，以提供消费者的订阅订单中的服务。

[0281] 在一些示例中，由云基础设施系统802提供的服务可以包括但不限于应用服务、平台服务和基础设施服务。在一些示例中，应用服务可以由云基础设施系统经由SaaS平台提供。SaaS平台可以被配置为提供属于SaaS类别的云服务。例如，SaaS平台可以提供在集成的开发和部署平台上构建和交付点播应用套件的能力。SaaS平台可以管理和控制用于提供SaaS服务的底层软件和基础设施。通过利用由SaaS平台提供的服务，消费者可以使用在云基础设施系统上执行的应用。消费者可以获取应用服务，而无需消费者单独购买许可证和支持。可以提供各种不同的SaaS服务。示例包括但不限于为大型组织提供用于销售绩效管理、企业集成和业务灵活性的解决方案的服务。

[0282] 在一些示例中，平台服务可以由云基础设施系统802经由PaaS平台提供。PaaS平台可以被配置为提供属于PaaS类别的云服务。平台服务的示例可以包括但不限于使组织(诸如Oracle)能够在共享的共同体系架构上整合现有应用的服务，以及利用由平台提供的共

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00439

享服务构建新应用的能力。PaaS平台可以管理和控制用于提供PaaS服务的底层软件和基础设施。消费者可以获取由云基础设施系统802提供的PaaS服务，而无需消费者购买单独的许可证和支持。平台服务的示例包括但不限于Oracle Java云服务(JCS)、Oracle数据库云服务(DBCS)以及其它。

[0283] 通过利用由PaaS平台提供的服务，消费者可以采用由云基础设施系统支持的编程语言和工具，并且还控制所部署的服务。在一些示例中，由云基础设施系统提供的平台服务可以包括数据库云服务、中间件云服务(例如，Oracle Fusion Middleware服务)和Java云服务。在一个示例中，数据库云服务可以支持共享服务部署模型，其使得组织能够汇集数据库资源并且以数据库云的形式向消费者提供数据库即服务。中间件云服务可以为消费者提供开发和部署各种业务应用的平台，以及Java云服务可以在云基础设施系统中为消费者提供部署Java应用的平台。

[0284] 可以由云基础设施系统中的IaaS平台提供各种不同的基础设施服务。基础设施服务促进底层计算资源(诸如存储装置、网络和其它基本计算资源)的管理和控制，以便消费者利用由SaaS平台和PaaS平台提供的服务。

[0285] 在一些示例中，云基础设施系统802还可以包括基础设施资源830，用于提供用来向云基础设施系统的消费者提供各种服务的资源。在一个示例中，基础设施资源830可以包括执行由PaaS平台和SaaS平台提供的服务的硬件(诸如服务器、存储装置和联网资源)的预先集成和优化的组合，以及其它资源。

[0286] 在一些示例中，云基础设施系统802中的资源可以由多个用户共享并且按需动态地重新分配。此外，资源可以分配给在不同时区中的用户。例如，云基础设施系统802可以使第一时区内的第一用户集合能够利用云基础设施系统的资源达到指定的小时数，然后使得能够将相同资源重新分配给位于不同时区中的另一用户集合，从而最大化资源的利用率。

[0287] 在一些示例中，可以提供由云基础设施系统802的不同部件或模块共享，以使得能够由云基础设施系统802供应服务的多个内部共享服务832。这些内部共享服务可以包括但不限于，安全和身份服务、集成服务、企业储存库服务、企业管理器服务、病毒扫描和白名单服务、高可用性、备份和恢复服务、用于启用云支持的服务、电子邮件服务、通知服务、文件传输服务等。

[0288] 在一些示例中，云基础设施系统802可以在云基础设施系统中提供云服务(例如，SaaS、PaaS和IaaS服务)的综合管理。在一个示例中，云管理功能可以包括用于供应、管理和跟踪由云基础设施系统802等接收到的消费者的订阅的能力。

[0289] 在一个示例中，如图8中所绘出的，云管理功能可以由诸如订单管理模块820、订单编排模块828、订单供应模块824、订单管理和监视模块826以及身份管理模块828的一个或多个模块提供。这些模块可以包括或可以使用一个或多个计算机和/或服务器提供，该一个或多个计算机和/或服务器可以是通用计算机、专用服务器计算机、服务器场，服务器集群或任何其它适当的布置和/或组合。

[0290] 在示例操作中，在步骤834处，使用客户端设备(诸如客户端计算设备804、806、808)的消费者可以通过请求由云基础设施系统802提供的一个或多个服务并且对由云基础设施系统802提供的一个或多个服务的订阅下订单来与云基础设施系统802交互。在一些示例中，消费者可以访问诸如第一云用户界面(UI)812、第二云UI 814和/或第三云UI 822的

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00440

云用户界面(UI)并经由这些UI下订阅订单。响应于消费者下订单而由云基础设施系统802接收到的订单信息可以包括识别消费者和消费者打算订阅的由云基础设施系统802提供的一个或多个服务的信息。

[0291] 在步骤836处，从消费者接收到的订单信息可以存储在订单数据库818中。如果这是新的订单，那么可以为该订单创建新的记录。在一个示例中，订单数据库818可以是由云基础设施系统818操作以及与其它系统元素结合操作的若干数据库当中的一个。

[0292] 在步骤838处，订单信息可以被转发到订单管理模块820，订单管理模块820可以被配置为执行与订单相关的计费和记帐功能，诸如验证订单，并且在通过验证后，预订订单。

[0293] 在步骤840处，关于订单的信息可以被传送到订单编排模块822，订单编排模块822被配置为编排用于由消费者下的订单的服务和资源的供应。在一些情况下，订单编排模块822可以使用订单供应模块824的服务用于供应。在一些示例中，订单编排模块822使得能够管理与每个订单相关联的业务处理，并且应用业务逻辑来确定订单是否应当继续供应。

[0294] 如图8中绘出的示例中所示，在842处，在接收到新订阅的订单后，订单编排模块822向订单供应模块824发送分配资源和配置履行订购订单所需的资源的请求。订单供应模块824使得能够为由消费者订购的服务分配资源。订单供应模块824提供由云基础设施系统800提供的云服务和用来供应用于提供所请求的服务的资源的物理实施方式层之间的抽象级别。这使得订单编排模块822能够与实施方式细节隔离，诸如服务和资源是实际上实时供应，还是预先供应并且仅在请求时才进行分配/指定。

[0295] 在步骤844处，一旦供应了服务和资源，就可以向订阅的消费者发送指示所请求的服务现在已准备好用于使用的通知。在一些情况下，可以向消费者发送使得消费者能够开始使用所请求的服务的信息(例如，链接)。

[0296] 在步骤846处，可以由订单管理和监视模块826来管理和跟踪消费者的订阅订单。在一些情况下，订单管理和监视模块826可以被配置为收集关于消费者使用所订阅的服务的使用统计。例如，可以针对所使用的存储量、所传送的数据量、用户的数量以及系统启动时间和系统停机时间的量等来收集统计数据。

[0297] 在一些示例中，云基础设施系统800可以包括身份管理模块828，其被配置为提供身份服务，诸如云基础设施系统800中的访问管理和授权服务。在一些示例中，身份管理模块828可以控制关于希望使用由云基础设施系统802提供的服务的消费者的信息。这种信息可以包括认证这些消费者的身份的信息和描述那些消费者被授权相对于各种系统资源(例如，文件、目录、应用、通信端口、存储器段等)执行的动作的信息。身份管理模块828还可以包括关于每个消费者的描述性信息以及关于如何和由谁来访问和修改描述性信息的管理。

[0298] 图9图示了可以被用于实现上面讨论的各种示例的计算机系统900的示例。在一些示例中，计算机系统900可以被用于实现上述各种服务器和计算机系统中的任何一个。如图9中所示，计算机系统900包括各种子系统，其包括处理子系统904，处理子系统904经由总线子系统902与多个外围子系统通信。这些外围子系统可以包括处理加速单元906、I/O子系统908、存储子系统918和通信子系统924。存储子系统918可以包括有形的计算机可读存储介质922和系统存储器910。

[0299] 总线子系统902提供用于使计算机系统900的各种部件和子系统按照期望彼此通信的机制。虽然总线子系统902被示意性地示为单个总线，但是总线子系统的替代示例可以

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00441

利用多个总线。总线子系统902可以是若干种类型的总线结构中的任何一种，包括存储器总线或存储器控制器、外围总线和利用任何各种总线体系架构的局部总线。例如，此类体系架构可以包括工业标准体系架构(ISA)总线、微通道体系架构(MCA)总线、增强型ISA(EISA)总线、视频电子标准协会(VESA)局部总线和外围部件互连(PCI)总线，其可以实现为根据IEEE P1386.1标准制造的夹层(Mezzanine)总线，等等。

[0300] 处理子系统904控制计算机系统900的操作并且可以包括一个或多个处理单元932、934。处理单元可以包括一个或多个处理器，包括单核或多核处理器、处理器的一个或多个核、或其组合。在一些示例中，处理子系统904可以包括一个或多个专用协处理器，诸如图形处理器、数字信号处理器(DSP)等。在一些示例中，处理子系统904的处理单元中的一些或全部可以利用定制电路来实现，诸如专用集成电路(ASIC)或现场可编程门阵列(FPGA)。

[0301] 在一些示例中，处理子系统904中的处理单元可以执行存储在系统存储器910中或计算机可读存储介质922上的指令。在各种示例中，处理单元可以执行各种程序或代码指令，并且可以维护多个并发执行的程序或进程。在任何给定的时间，要执行的程序代码中的一些或全部可以驻留在系统存储器910中和/或计算机可读存储介质922上，潜在地包括在一个或多个存储设备上。通过适当的编程，处理子系统904可以提供各种功能。

[0302] 在一些示例中，可以提供处理加速单元906，用于执行定制的处理或用于卸载由处理子系统904执行的一些处理，以便加速由计算机系统900执行的整体处理。

[0303] I/O子系统908可以包括用于向计算机系统900输入信息和/或用于从或经由计算机系统900输出信息的设备和机制。一般而言，术语“输入设备”的使用旨在包括用于向计算机系统900输入信息的所有可能类型的设备和机制。用户接口输入设备可以包括，例如，键盘、诸如鼠标或轨迹球的指示设备、结合到显示器中的触摸板或触摸屏、滚轮、点拨轮、拨盘、按钮、开关、键板、具有语音命令识别系统的音频输入设备、麦克风以及其它类型的输入设备。用户接口输入设备也可以包括使用户能够控制输入设备并与其交互的诸如Microsoft **Kinect®**运动传感器的运动感测和/或姿势识别设备、Microsoft **Xbox®**360游戏控制器、提供用于接收利用姿势和口语命令的输入的接口的设备。用户接口输入设备也可以包括眼睛姿势识别设备，诸如从用户检测眼睛活动(例如，当拍摄图片和/或进行菜单选择时的“眨眼”)并将眼睛姿势转换为到输入设备(例如，Google **Glass®**)中的输入的Google **Glass®**眨眼检测器。此外，用户接口输入设备可以包括使用户能够通过语音命令与语音识别系统(例如，**Siri®**导航器)交互的语音识别感测设备。

[0304] 用户接口输入设备的其它示例包括但不限于，三维(3D)鼠标、操纵杆或指示杆、游戏板和图形平板、以及音频/视频设备，诸如扬声器、数字相机、数字摄像机、便携式媒体播放器、网络摄像机、图像扫描仪、指纹扫描仪、条形码读取器3D扫描仪、3D打印机、激光测距仪、以及眼睛注视跟踪设备。此外，用户接口输入设备可以包括，例如，医疗成像输入设备，诸如计算机断层摄影、磁共振成像、位置发射断层摄影、医疗超声检查设备。用户接口输入设备也可以包括，例如，音频输入设备，诸如MIDI键盘、数字乐器等。

[0305] 用户接口输出设备可以包括显示子系统、指示器灯或诸如音频输出设备的非可视显示器等。显示子系统可以是阴极射线管(CRT)、诸如利用液晶显示器(LCD)或等离子体显示器的平板设备、投影设备、触摸屏等。一般而言，术语“输出设备”的使用旨在包括用于从

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00442

计算机系统900向用户或其它计算机输出信息的所有可能类型的设备和机制。例如，用户接口输出设备可以包括但不限于，可视地传达文本、图形和音频/视频信息的各种显示设备，诸如监视器、打印机、扬声器、耳机、汽车导航系统、绘图仪、语音输出设备和调制解调器。

[0306] 存储子系统918提供用于存储由计算机系统900使用的信息的储存库或数据存储。存储子系统918提供有形非瞬态计算机可读存储介质，用于存储提供一些示例的功能的基本编程和数据结构。当由处理子系统904执行时提供上述功能的软件(程序、代码模块、指令)可以存储在存储子系统918中。软件可以由处理子系统904的一个或多个处理单元执行。存储子系统918也可以提供用于存储根据本公开使用的数据的储存库。

[0307] 存储子系统918可以包括一个或多个非暂态存储器设备，包括易失性和非易失性存储器设备。如图9中所示，存储子系统918包括系统存储器910和计算机可读存储介质922。系统存储器910可以包括多个存储器，包括用于在程序执行期间存储指令和数据的易失性主随机存取存储器(RAM)和其中存储固定指令的非易失性只读存储器(ROM)或闪存存储器。在一些实施方式中，包含帮助在诸如启动期间在计算机系统900内的元件之间传送信息的基本例程的基本输入/输出系统(BIOS)通常可以存储在ROM中。RAM通常包含当前由处理子系统904操作和执行的数据和/或程序模块。在一些实施方式中，系统存储器910可以包括多个不同类型的存储器，诸如静态随机存取存储器(SRAM)或动态随机存取存储器(DRAM)。

[0308] 作为示例而非限制，如在图9中所绘出的，系统存储器910可以存储应用程序912，其可以包括客户端应用、Web浏览器、中间层应用、关系数据库管理系统(RDBMS)等、程序数据923和操作系统916。作为示例，操作系统916可以包括各种版本的Microsoft **Windows®**、Apple **Macintosh®**和/或Linux操作系统、各种商用**UNIX®**或类UNIX操作系统(包括但不限于各种GNU/Linux操作系统、Google **Chrome®** OS等)和/或诸如iOS、**Windows®**Phone、**Android®** OS、**BlackBerry®** 10OS和**Palm®**OS操作系统的移动操作系统。

[0309] 计算机可读存储介质922可以存储提供一些示例的功能的编程和数据结构。当由处理子系统904执行时使处理器提供上述功能的软件(程序、代码模块、指令)可以存储在存储子系统918中。作为示例，计算机可读存储介质922可以包括非易失性存储器，诸如硬盘驱动器、磁盘驱动器、诸如CD ROM、DVD、**Blu-Ray®**(蓝光)盘或其它光学介质的光盘驱动器。计算机可读存储介质922可以包括但不限于，**Zip®**驱动器、闪存存储器卡、通用串行总线(USB)闪存驱动器、安全数字(SD)卡、DVD盘、数字视频带等。计算机可读存储介质922也可以包括基于非易失性存储器的固态驱动器(SSD)(诸如基于闪存存储器的SSD、企业闪存驱动器、固态ROM等)、基于易失性存储器的SSD(诸如基于固态RAM、动态RAM、静态RAM、DRAM的SSD、磁阻RAM(MRAM)SSD)，以及使用基于DRAM和基于闪存存储器的SSD的组合的混合SSD。计算机可读存储介质922可以为计算机系统900提供计算机可读指令、数据结构、程序模块和其它数据的存储。

[0310] 在一些示例中，存储子系统918也可以包括计算机可读存储介质读取器920，其可以进一步连接到计算机可读存储介质922。与系统存储器910一起并可选地与其组合，计算机可读存储介质922可以全面地表示远程、本地、固定和/或可移动存储设备加上用于存储计算机可读信息的存储介质。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00443

[0311] 在一些示例中，计算机系统900可以提供对执行一个或多个虚拟机的支持。计算机系统900可以执行诸如管理程序之类的程序，以便促进虚拟机的配置和管理。每个虚拟机可以被分配存储器、计算(例如，处理器、内核)、I/O和联网资源。每个虚拟机通常运行其自己的操作系统，其可以与由计算机系统900执行的其它虚拟机执行的操作系统相同或不同。相应地，多个操作系统可以潜在地由计算机系统900并发地运行。每个虚拟机一般独立于其它虚拟机运行。

[0312] 通信子系统924提供到其它计算机系统和网络的接口。通信子系统924用作用于从计算机系统900的其它系统接收数据和向其传输数据的接口。例如，通信子系统924可以使计算机系统900能够经由互联网建立到一个或多个客户端计算设备的通信信道，用于从客户端计算设备接收信息和发送信息到客户端计算设备。

[0313] 通信子系统924可以支持有线和/或无线通信协议两者。例如，在一些示例中，通信子系统924可以包括用于(例如，使用蜂窝电话技术、高级数据网络技术(诸如3G、4G或EDGE(全球演进的增强数据速率)、WiFi(IEEE 802.11族标准)、或其它移动通信技术、或其任意组合)接入无线语音和/或数据网络的射频(RF)收发器部件、全球定位系统(GPS)接收器部件和/或其它部件。在一些示例中，作为无线接口的附加或替代，通信子系统924可以提供有线网络连接(例如，以太网)。

[0314] 通信子系统924可以以各种形式接收和传输数据。例如，在一些示例中，通信子系统924可以以结构化和/或非结构化的数据馈送926、事件流928、事件更新930等形式接收输入通信。例如，通信子系统924可以被配置为实时地从社交媒体网络的用户和/或诸如**Twitter®**馈送、**Facebook®**更新、诸如丰富站点摘要(RSS)馈送的web馈送的其它通信服务接收(或发送)数据馈送926，和/或来自一个或多个第三方信息源的实时更新。

[0315] 在一些示例中，通信子系统924可以被配置为以连续数据流的形式接收本质上可能是连续的或无界的没有明确结束的数据，其中连续数据流可以包括实时事件的事件流928和/或事件更新930。生成连续数据的应用的示例可以包括例如传感器数据应用、金融报价机、网络性能测量工具(例如网络监视和流量管理应用)、点击流分析工具、汽车流量监视等。

[0316] 通信子系统924还可以被配置为向一个或多个数据库输出结构化和/或非结构化的数据馈送926、事件流928、事件更新930等，其中所述一个或多个数据库可以与耦合到计算机系统900的一个或多个流数据源计算机通信。

[0317] 计算机系统900可以是各种类型中的一种，包括手持便携式设备(例如，**iPhone®**蜂窝电话、**iPad®**计算平板、PDA)、可穿戴设备(例如，Google**Glass®**头戴式显示器)、个人计算机、工作站、大型机、信息站、服务器机架或任何其它数据处理系统。

[0318] 由于计算机和网络不断变化的性质，对图9中绘出的计算机系统900的描述旨在仅仅作为具体示例。具有比图9中所绘出的系统更多或更少部件的许多其它配置是可能的。基于本文所提供的公开内容和教导，本领域普通技术人员将理解实现各种示例的其它方式和/或方法。

[0319] 虽然已经描述了具体实施方式，但是各种修改、更改、替代构造和等效物也包含在本公开的范围之内。修改包括所公开的特征的任何相关组合。本公开中描述的实施方式不限于在某些具体数据处理环境内的操作，而是可以在多个数据处理环境内自由操作。此外，

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00444

虽然已经使用特定序列的事务和步骤描述了本公开中描述的实施方式，但是，对本领域技术人员应当清楚的是，本公开的范围不限于所描述系列的事务和步骤。上述实施方式的各种特征和方面可以被单独或结合使用。

[0320] 另外，虽然已经利用硬件和软件的特定组合描述了本公开中描述的实施方式，但是应当认识到的是，硬件和软件的其它组合也在本公开的范围之内。本公开中描述的实施方式可以只用硬件、或只用软件、或利用其组合来实现。本文描述的各种处理可以在同一处理器或以任何组合的不同处理器上实现。因而，在部件或模块被描述为被配置为执行某些操作的情况下，这种配置可以例如通过设计电子电路来执行操作、通过对可编程电子电路(诸如微处理器)进行编程来执行操作、或其任意组合来实现。进程可以使用各种技术来通信，包括但不限于用于进程间通信的常规技术，并且不同的进程对可以使用不同的技术，或者同一对进程可以在不同时间使用不同的技术。

[0321] 因而，说明书和附图应当在说明性而不是限制性的意义上考虑。但是，将清楚的是，在不背离权利要求中阐述的更广泛精神和范围的情况下，可以对其进行添加、减少、删除和其它修改和改变。因此，虽然已描述了具体的实施方式，但是这些实施方式不旨在进行限制。各种修改和等效物都在以下权利要求的范围之内。

[0322] 如以下所使用的，对一系列示例的任何引用都应被理解为分离地对这些示例中的每一个的引用(例如，“示例1-4”应被理解为“示例1、2、3或4”)。

[0323] 示例1是一种计算机实现的方法，包括在安全性管理系统的计算机系统处执行的步骤。这些步骤包括从服务提供商系统获得活动数据，其中活动数据描述在使用云服务期间执行的动作，其中动作由与租户相关联的一个或多个用户执行，其中服务提供商系统向租户提供租户账户，并且其中租户账户使该一个或多个用户能够访问云服务。步骤还包括在活动数据中识别相对于云服务具有特权的一个或多个动作。步骤还包括使用活动数据来识别执行了该一个或多个动作的用户集，其中用户集是从与租户相关联的该一个或多个用户中确定的。步骤还包括将用户集分类为具有特权。步骤还包括使用活动数据为一个或多个用户确定一个或多个风险得分。步骤还包括确定用户集中的用户的风险得分大于阈值。步骤还包括确定用于服务提供商系统的安全性控制，其中安全性控制被服务提供商系统用于配置对云服务的访问。步骤还包括确定要发送到服务提供商系统的一个或多个指令。步骤还包括将一个或多个指令发送到服务提供商系统，其中一个或多个指令使安全性控制相对于用户被改变，其中用户对云服务的访问由于对安全性控制的改变而被修改。

[0324] 示例2是示例1的方法，其中使用与云服务相关联的动作的列表来识别该一个或多个动作，其中动作的列表中的动作相对于云服务被分类为具有特权。

[0325] 示例3是示例1-2的方法，其中该一个或多个动作是使用管理性动作的列表来识别的。

[0326] 示例4是示例1-3的方法，其中步骤还包括使用该一个或多个动作和过去的活动数据来生成模型，该模型描述相对于云服务具有特权的云服务的使用模式。步骤还包括使用模型来识别用户集。

[0327] 示例5是示例1-4的方法，其中步骤还包括对在云服务的使用期间执行的动作进行分组。步骤还包括识别包括具有特权的动作的一组动作，其中用户集是使用该组动作来识别的。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00445

[0328] 示例6是示例1-5的方法，其中风险得分指示用户在使用云服务时所执行的动作对租户的安全性风险的程度。

[0329] 示例7是示例1-6的方法，其中将风险得分计算为风险指标的加权和。

[0330] 示例8是示例1-7的方法，其中，与非特权用户的风险得分相比，以更大的权重来计算被分类为具有特权的用户的风险得分。

[0331] 示例9是示例1-8的方法，其中特权动作是在由第一用户执行时可以以影响其他用户对云服务的使用的方式来修改云服务的动作。

[0332] 示例10是示例1-9的方法，其中特权动作是当由第一用户执行时可以影响云服务的其他用户的用户账户的动作。

[0333] 示例11是一种安全性管理系统的计算系统，其包括一个或多个处理器以及耦合到一个或多个处理器并可由其读取的存储器。存储器包括指令，指令在由一个或多个处理器执行时使一个或多个处理器执行根据示例1-10的(一个或多个)方法的操作。

[0334] 示例12是一种非暂态机器可读介质，该介质上存储有指令，指令在由安全性管理系统的计算系统的一个或多个处理器执行时使一个或多个处理器执行根据示例1-10的(一个或多个)方法的步骤。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00446

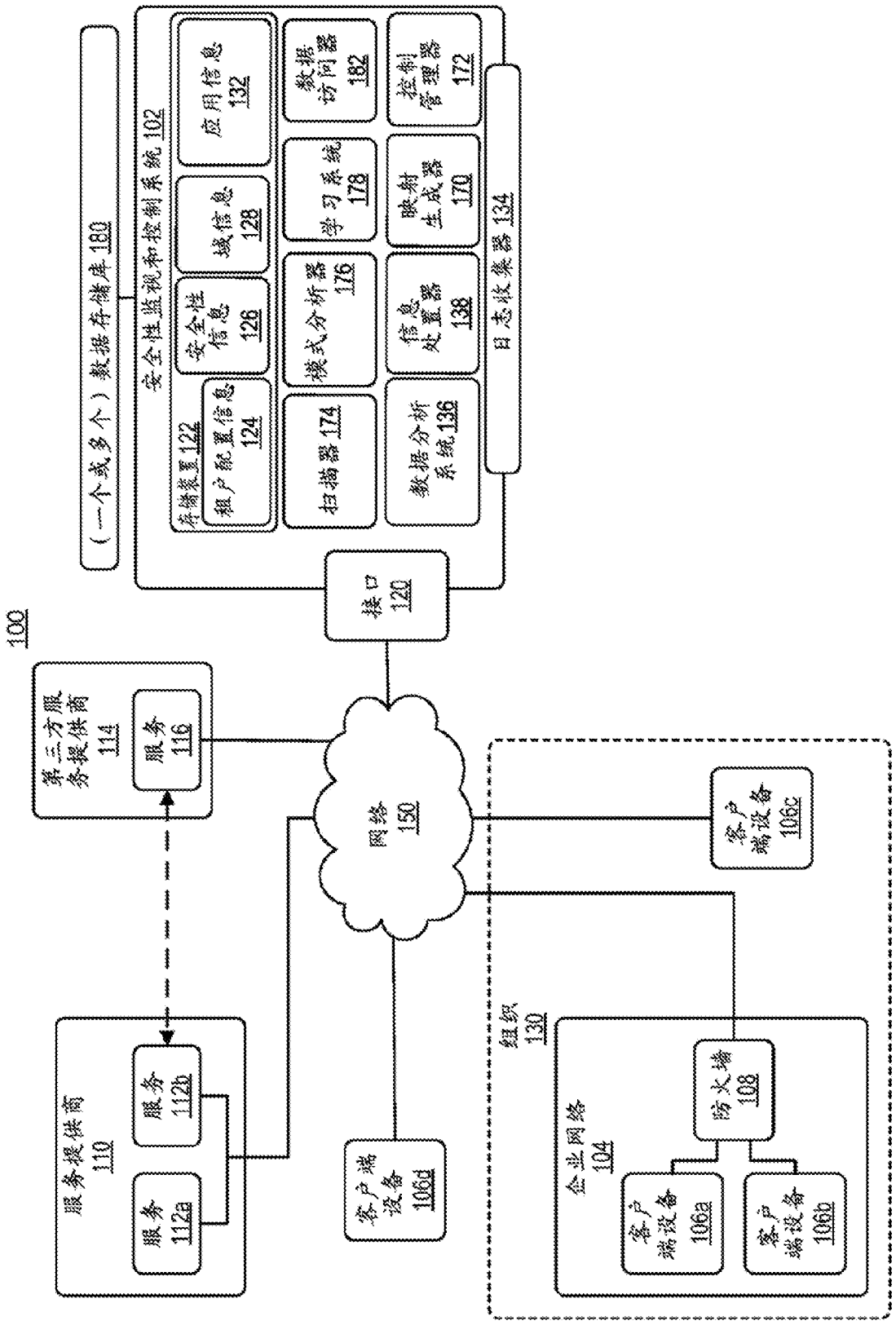


图1

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00447

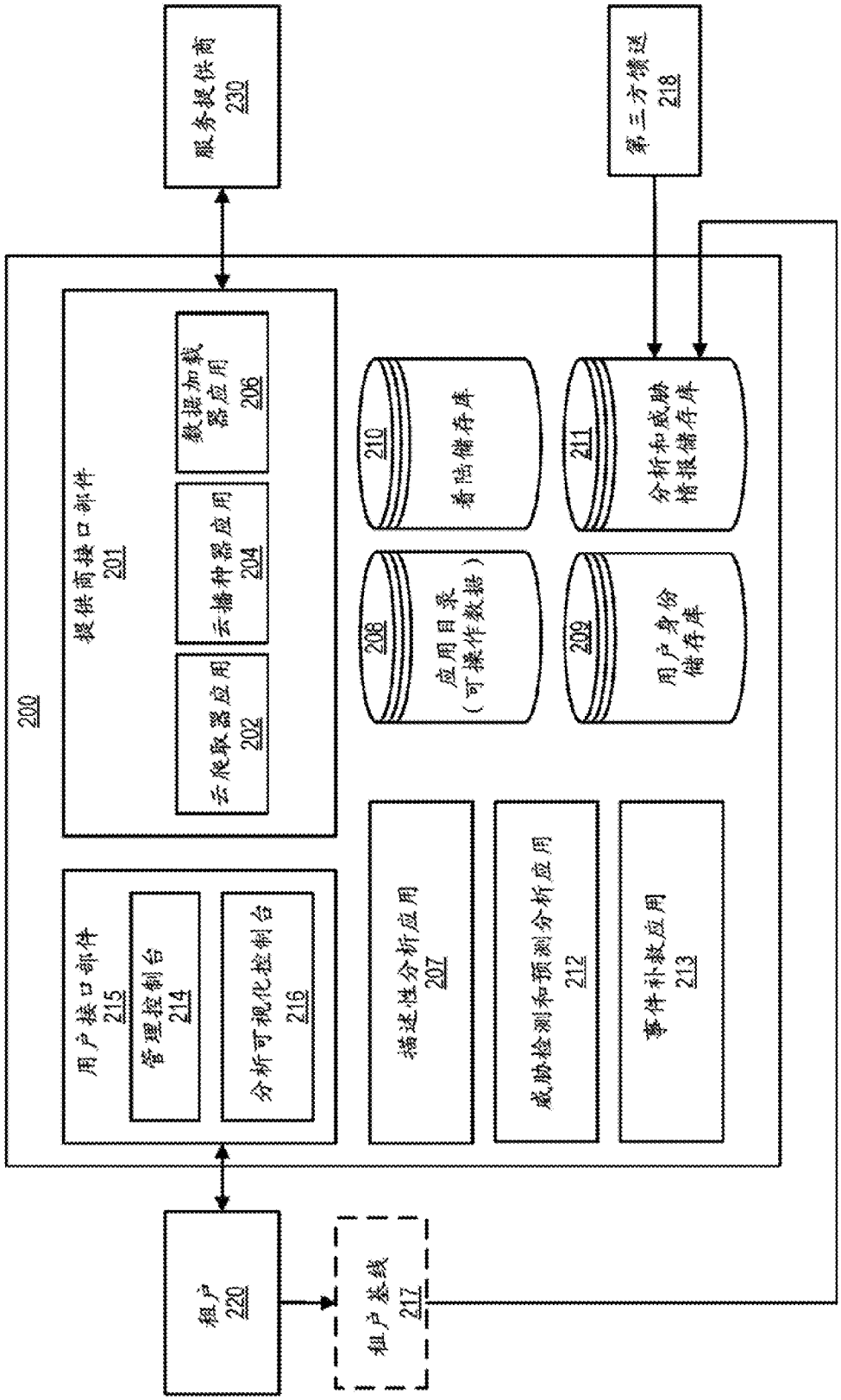


图2

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00448

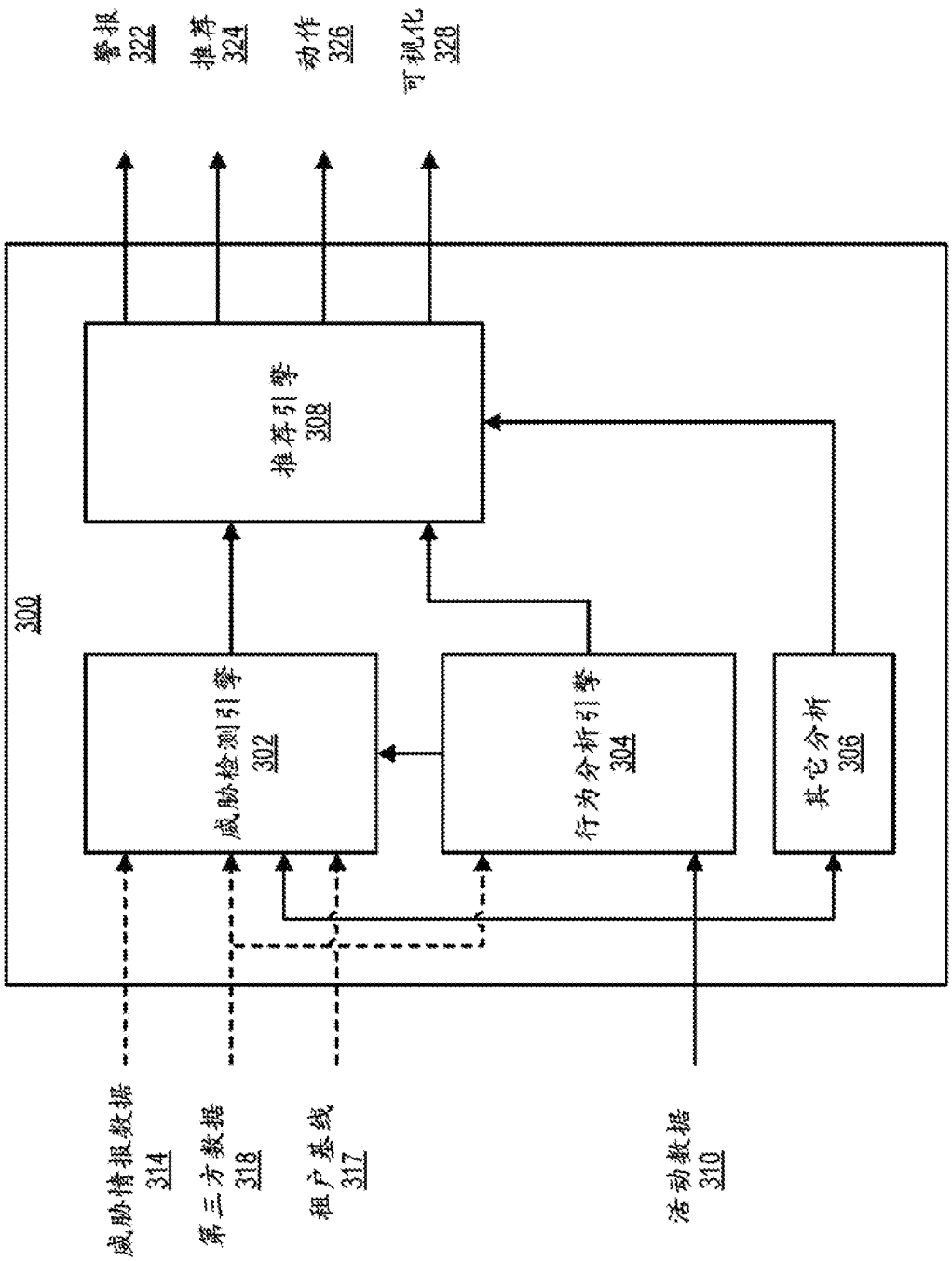


图3

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00449

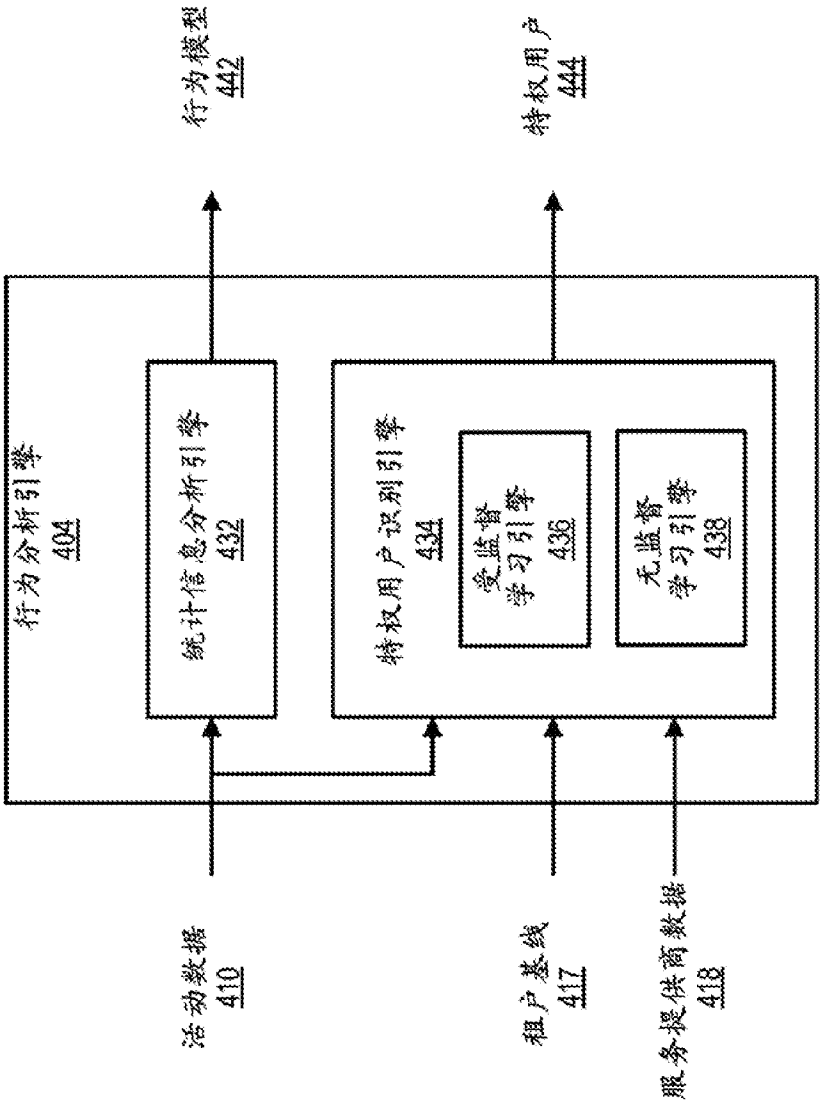


图4

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00450

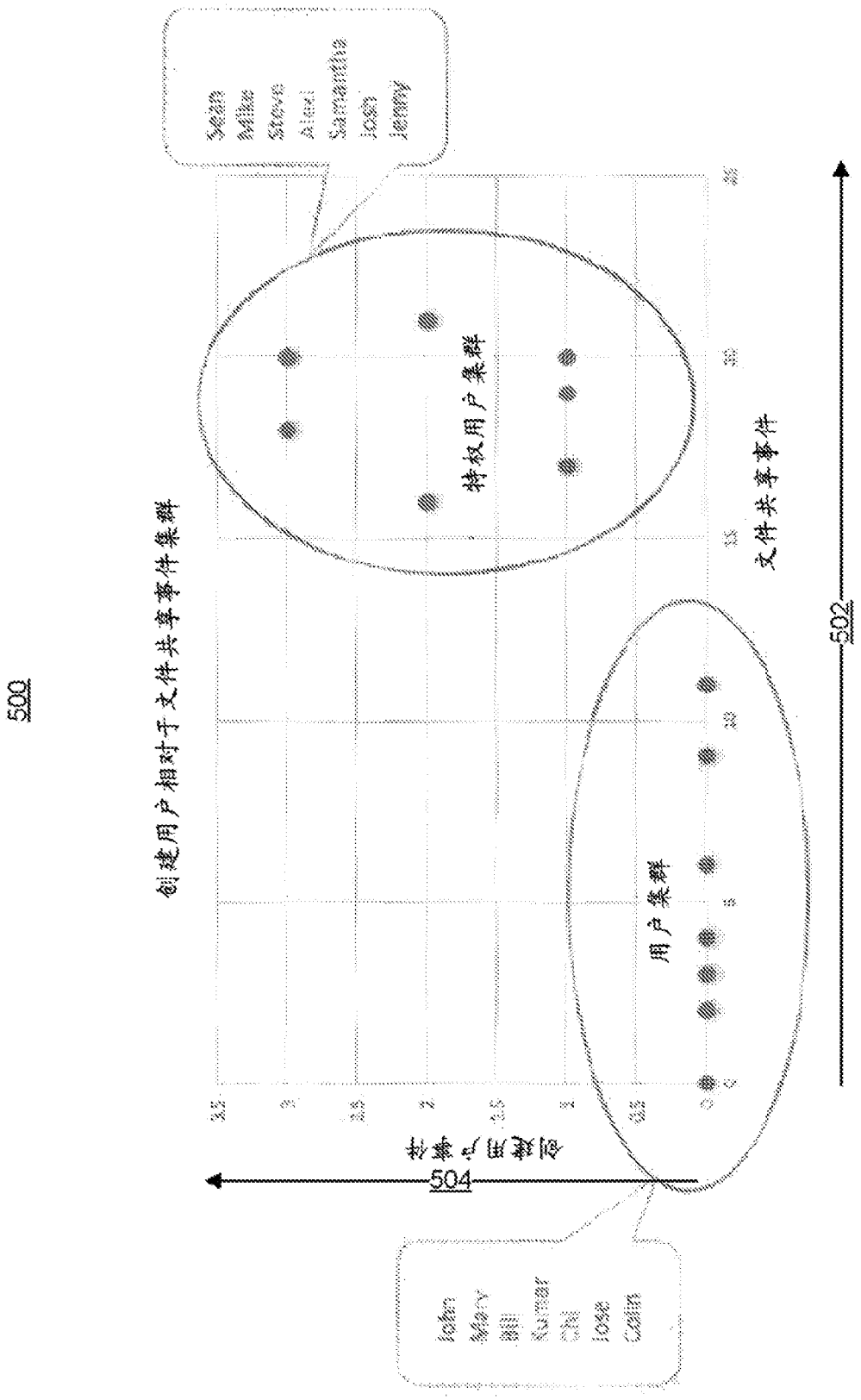


图5

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00451

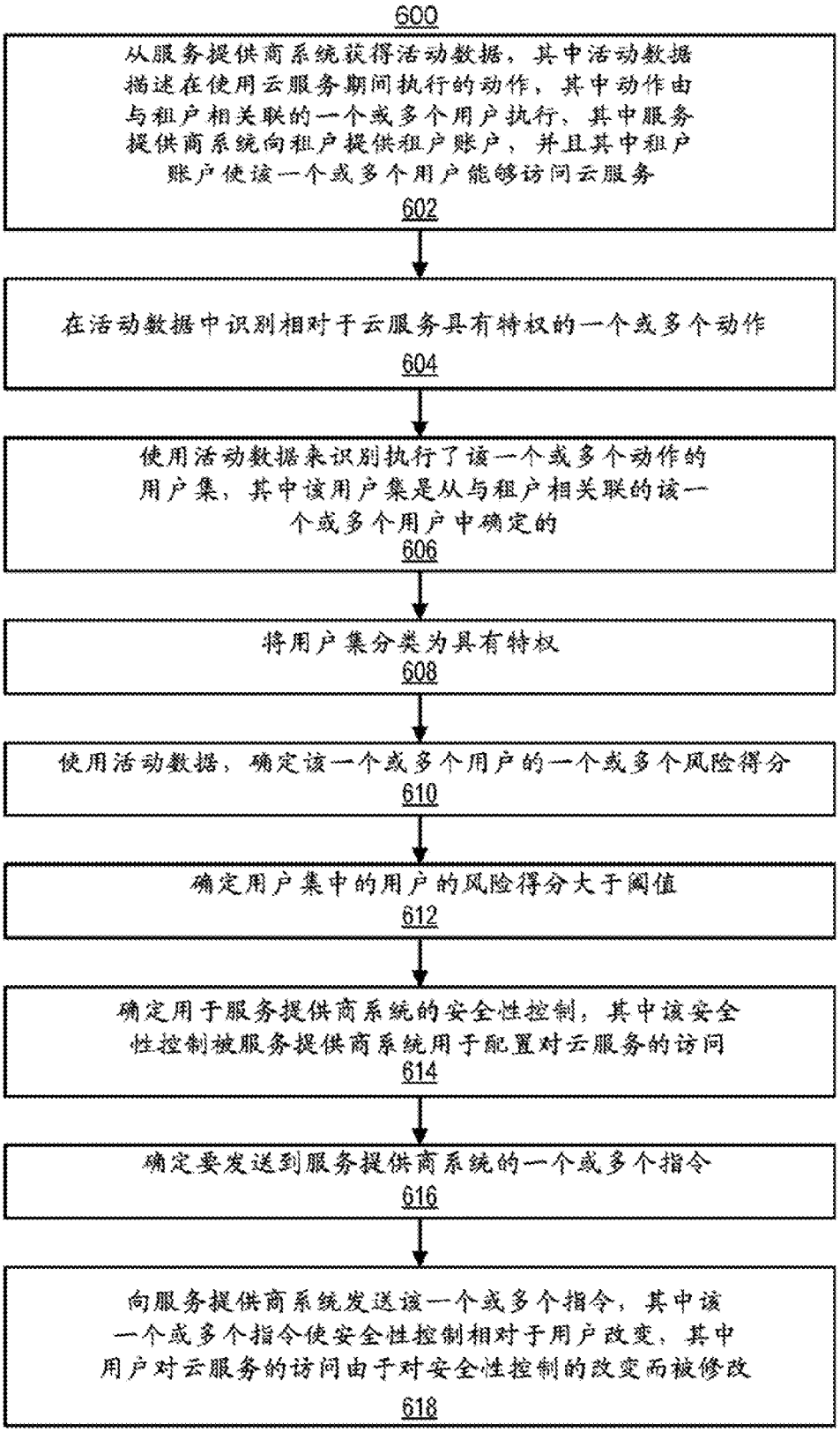


图6

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00452

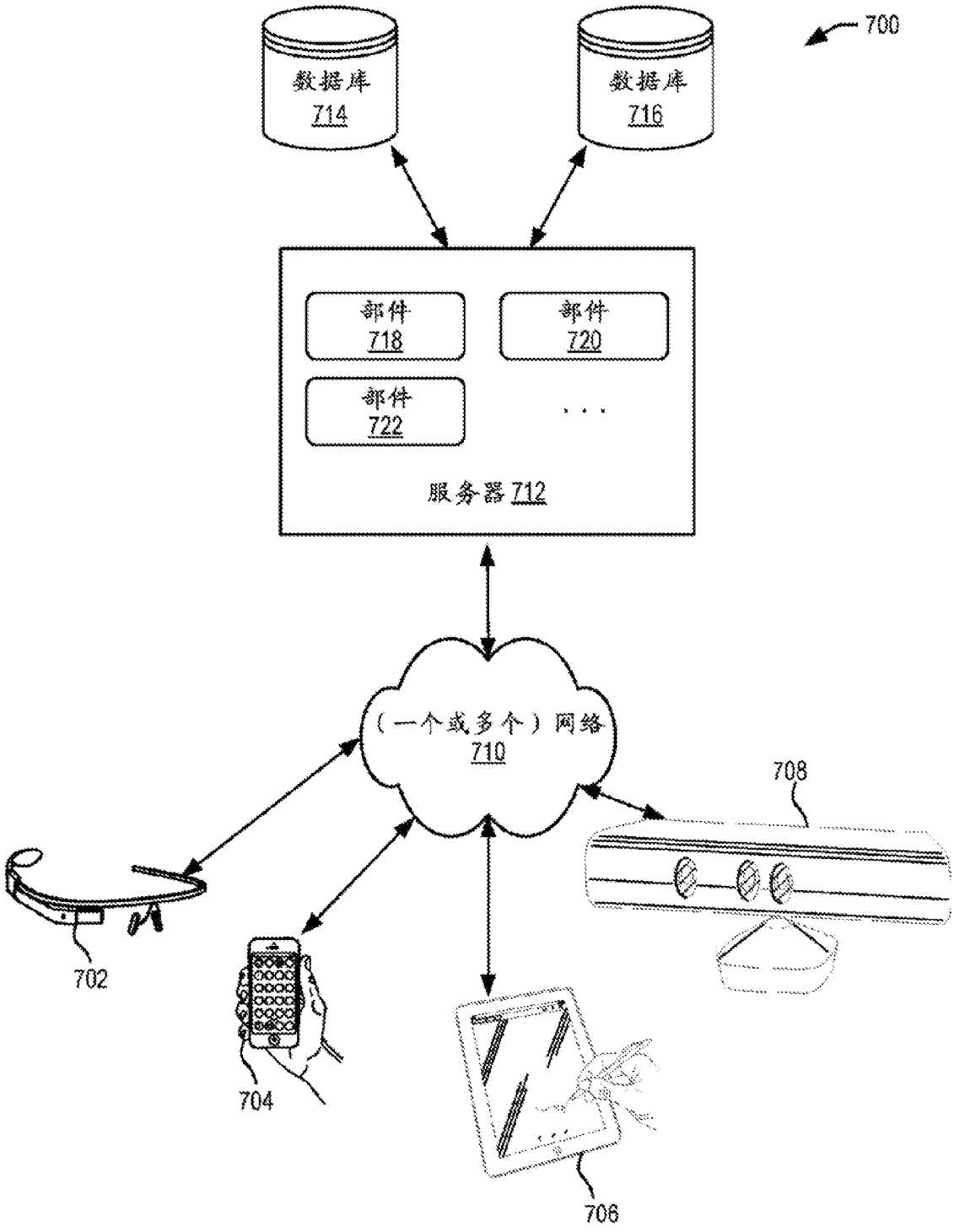


图7

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00453

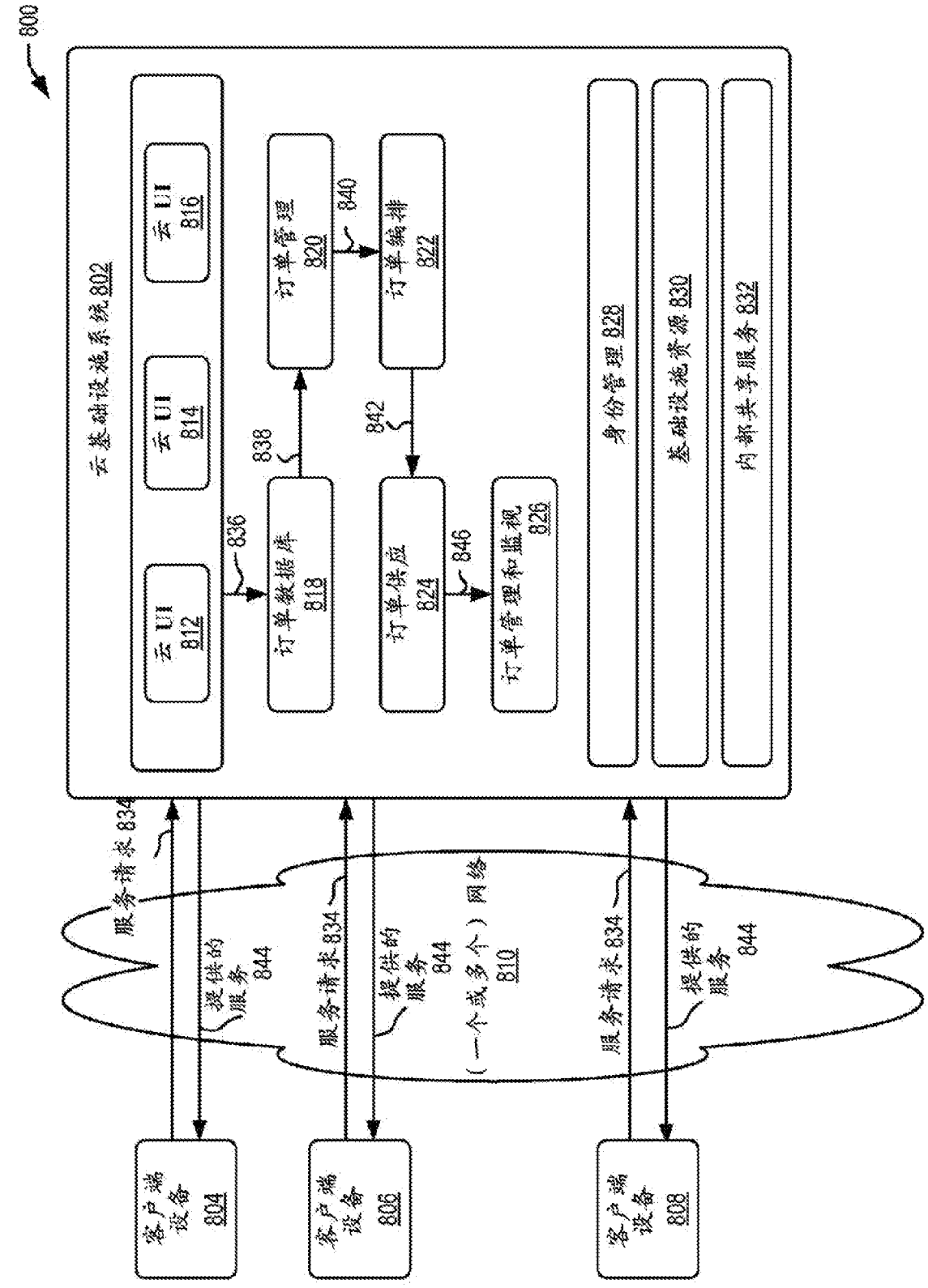


图8

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00454

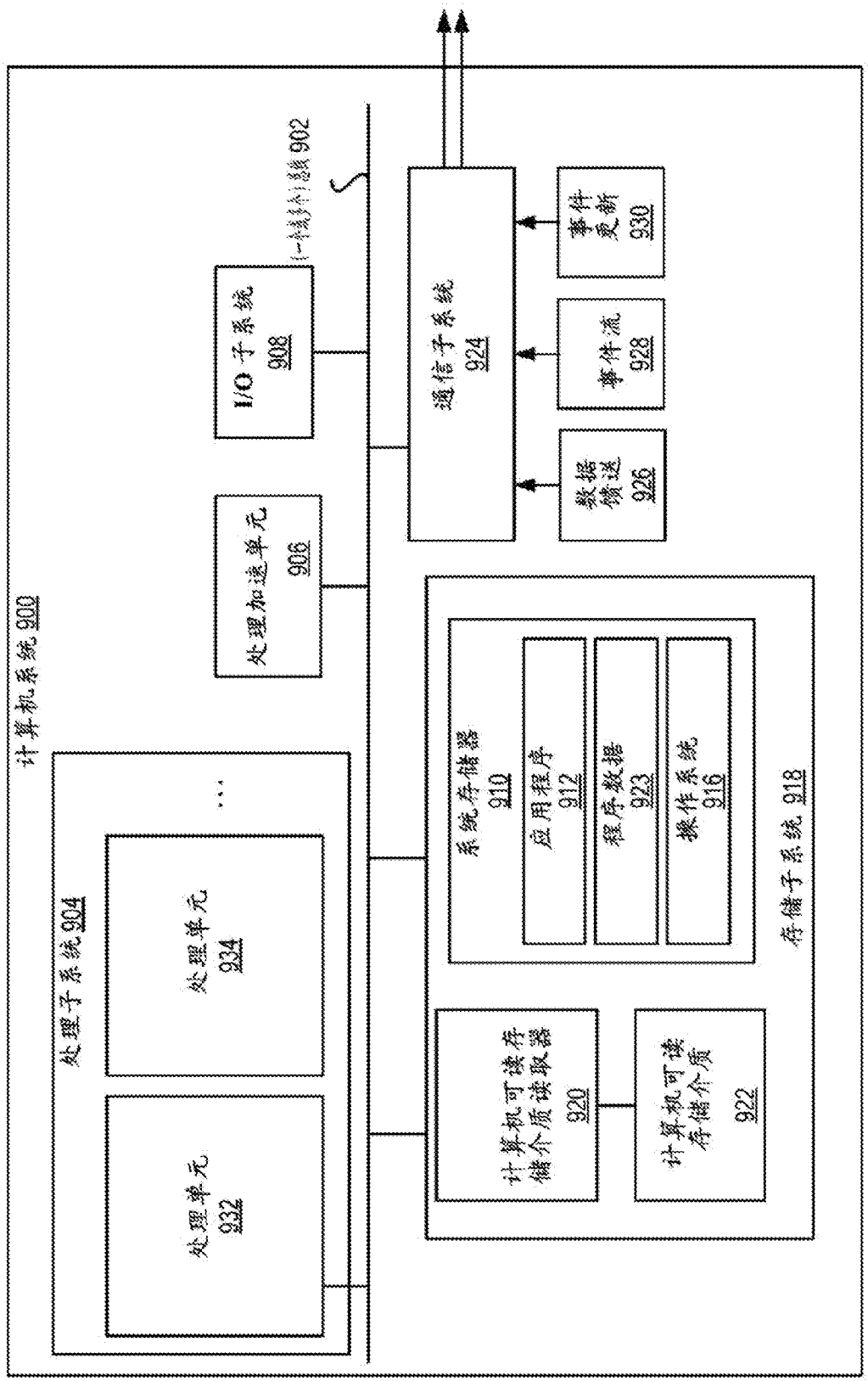


图9

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00455

Google cloud deployment

All Images News Videos Shopping More Tools

About 421,000,000 results (0.58 seconds)

Ad · https://azure.microsoft.com/services/cloud_services

Microsoft Azure Cloud Services - Get Your Free Account

Focus on Apps, Not Hardware. Autoscale to Meet Demand and Save Money.

Cloud deployment is the process of deploying an application through one or more hosting models---software as a service (SaaS), platform as a service (PaaS) and/or infrastructure as a service (IaaS)—that leverage the cloud. This includes architecting, planning, implementing and operating workloads on cloud.

https://www.cognizant.com › glossary › cloud-deployment

What is cloud deployment? - Cognizant

About featured snippets • Feedback

Things to know

- Aws three types — What are the three types of cloud deployment ...
- Advantages — Advantages of cloud deployments
- Role — What does cloud operations do
- Requirements — Cloud requirements

More

Feedback

People also ask

What are 3 types of cloud deployment models?

What are the four cloud deployment types?

What is AWS cloud deployment?

What is an example of a cloud deployment model?

Feedback

https://bejamas.io › discovery › cloud-deployment

What does it mean when we say Cloud Deployment? - Bejamas

Jul 6, 2021 --- Cloud Deployment is the enablement of service models that may be accessed on demand by end-users depending on the cloud model, a type of cloud ...
Related Posts · Cloud Computing Service... · Cloud Computing Deployment...

https://www.geeksforgeeks.org › cloud-deployment-mo...

Cloud Deployment Models - GeeksforGeeks

Nov 29, 2022 --- The cloud deployment model identifies the specific type of cloud environment based on ownership, scale, and access, as well as the cloud's ...

https://apprenda.com › Cloud Computing Overview

Deployment to the Cloud - Apprenda

Cloud deployment refers to the enablement of SaaS (software as a service), PaaS (platform as a service) or IaaS (infrastructure as a service) solutions that ...

https://www.sciencedirect.com › topics › computer-science

Cloud Deployment Model - an overview | ScienceDirect Topics

There are four cloud deployment models: public, private, community, and hybrid. Each deployment model is defined according to where the infrastructure for the ...

https://cloud.google.com › deploy

Google Cloud Deploy - Fully Managed Continuous Delivery

Google Cloud Deploy makes continuous delivery to GKE and Anthos easy and powerful. Define releases and progress them through environments such as test, stage, ...

Connect the tools you love: Google Cloud Dep... Easy rollout / rollback: Rollout and rollback ...
Pipeline visualization: Visualize the path to del... Auditing and traceability: Google Cloud De...

https://www.suntecgroup.com › blogs › what-is-cloud-d...

What is Cloud Deployment? - SunTec Group

What is Cloud Deployment? ... The integration of software-as-a-service (SaaS), platform-as-a-service (PaaS), and infrastructure-as-a-service (IaaS) into the cloud ...

https://launchdarkly.com › blog › cloud-deployment-mod...

5 Cloud Deployment Models: Which One is Right for You?

Oct 4, 2022 — A cloud deployment model defines the cloud services you are consuming and the responsibility model for who manages them.
What Is A Cloud Deployment... · Cloud Service Models · Deploying To The Cloud

https://www.virtasant.com › blog › aws-cloud-services-...

AWS Cloud Services and Deployment Models - Virtasant

Sep 30, 2021 — Choosing cloud deployment and service models can be a complicated decision. Our overview provides clarity so you can make an informed choice ...
Cloud Computing Models · Software As A Service (saas) · Cloud Computing Deployment...

https://www.virtusa.com › Digital Themes Glossary

What is Cloud Deployment? - Virtusa

Cloud deployment is the utilization of cloud environments to run applications through the use of different models, such as software-as-a-service (SaaS), ...

Related searches

Deployment models of cloud computing

Public cloud	Software as a service	Virtual private cloud	Platform as a service	Multicloud	Infrastructu...

See more

Cloud service deployment tools

Feedback

- cloud deployment **aws**
- cloud deployment **gcp**
- cloud deployment **services**
- cloud deployment **platforms**
- cloud deployment **models**
- **community** cloud deployment **model**
- cloud deployment **examples**
- cloud deployment **models tutorialspoint**

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00457

Fairfax County, Virginia - From your IP address - Update location

Help Send feedback Privacy Terms

cloud deploy* learn* similar*

Everything

ADVANCED SEA

1 ARTICLE

An evaluation of multi-probe locality sensitive hashing for computing similarities over web-scale query logs

Cormode, Graham ; Dasgupta, Anirban ; Goyal, Amit ; Lee, Chi Hoon; Wang, Yeng-Tseng

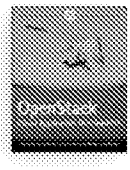
PloS one, 2018, Vol.13 (1), p.e0191175-e0191175

Many modern applications of AI such as web search, mobile browsing, image processing, and natural language processing rely on finding similar items from a large database of complex objects...

PEER REVIEWED OPEN ACCESS

Available Online

2 BOOK



Openstack cloud application development

Adkins, Scott, author.; Adkins, Scott, author.

2016

Available Online

3 ARTICLE

Development and clinical deployment of a smartphone-based visual field deep learning system for glaucoma detection

Li, Fei ; Song, Diping ; Chen, Han ; Xiong, Jian ; Li, Xingyi ; Zhong, Hua ; Tang, Guangxian ; Fan, Sujie ; Lam, Dennis S C ; Pan, Weihua ; Zheng, Yajuan ; Li, Ying ; Qu, Guoxiang ; He, Junjun ; Wang, Zhe ; Jin, Ling ; Zhou, Rouxi ; Song, Yunhe ; Sun, Yi ; Cheng, Weijing ; Yang, Chunman ; Fan, Yazhi ; Li, Yingjie ; Zhang, Hengli ; Yuan, Ye ; Xu, Yang ; Xiong, Yunfan ; Jin, Lingfei ; Lv, Aiguo ; Niu, Lingzhi ; Liu, Yuho…

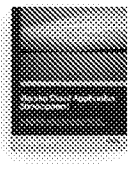
NPJ digital medicine, 2020, Vol.3 (1), p.123-123

...). Herein, we develop and evaluate the performance of 'iGlaucoma', a smartphone application-based deep learning system (DLS...

PEER REVIEWED OPEN ACCESS

Available Online

4 BOOK



Heroku cloud application development : a comprehensive guide to help you build, deploy, and troubleshoot cloud applications seamlessly using Heroku

Hanjura, Anubhav, author.; Shiriskar, Sagar, cover designer.

2014

Available Online



De, Debashis.; Mukherjee, Anwesha.; Buyya, Rajkumar.
2022
Available Online >

6



BOOK
Docker in the Cloud
Goasguen, Sébastien, author.
2016
Available Online >

7



VIDEO
Deploying Elastic Cloud Compute (EC2) Instances
Suehring, Steve, author.
2017
Available Online >

8



ARTICLE
Semantic tracking and recommendation using fourfold similarity measure from large scale data using hadoop distributed framework in cloud
R, Priyadarshini ; Tamilselvan, Latha ; N, Rajendran
International journal of intelligent unmanned systems, 2019, Vol.7 (4), p.189-208
Purpose The purpose of this paper is to propose a fourfold semantic similarity that results in more accuracy compared to the existing literature...
PEER REVIEWED
Available Online >

9



ARTICLE
RAFALE: Rethinking the provisioning of virtuAl network services using a Fast and scAlable machine LEarning approach
Suwi, Hanan ; Lahiou, Laaziz ; Kara, Nadjia ; Edstrom, Claes
The Journal of supercomputing, 2022, Vol.78 (14), p.15786-15819
.... To reap the benefits of this new technology, there is a need for novel mechanisms that help cloud and network/service providers deploy the increasingly complex virtual network services seamlessly...
PEER REVIEWED
Available Online >

10



ARTICLE
A Graph Representation Learning Algorithm for Low-Order Proximity Feature Extraction to Enhance Unsupervised IDS Preprocessing
Hao, Yiran ; Sheng, Yiqiang ; Wang, Jinlin
Applied sciences, 2019, Vol.9 (20), p.4473
.... According to the homophily hypothesis, the local proximity structure in the similarity relational graph has similar embedding after preprocessing...

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00460



1 2 3 4 5 >

Results Per Page 10 25 50

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2021 年 3 月 25 日 (25.03.2021)

WIPO | PCT

(10) 国际公布号
WO 2021/052379 A1

(51) 国际专利分类号：
H04L 12/851 (2013.01)

(21) 国际申请号： PCT/CN2020/115693

(22) 国际申请日： 2020 年 9 月 16 日 (16.09.2020)

(25) 申请语言： 中文

(26) 公布语言： 中文

(30) 优先权：
201910872990.0 2019年9月16日 (16.09.2019) CN

(71) 申请人：华为技术有限公司 (**HUAWEI TECHNOLOGIES CO., LTD.**) [CN/CN]；中国广东省深圳市龙岗区坂田华为总部办公楼，**Guangdong** 518129 (**CN**)。

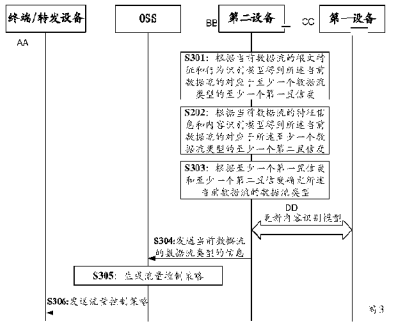
(72) 发明人：吴俊 (**WU, Jim**)；中国广东省深圳市龙岗区坂田华为总部办公楼，**Guangdong** 518129 (**CN**)。胡新宇 (**HU, Xinyu**)；中国广东省深圳市龙岗区坂田华为总部办公楼，**Guangdong** 518129 (**CN**)。王雅莉 (**WANG, Yali**)；中国广东省深圳市龙岗区坂田华为总部办公楼，**Guangdong** 518129 (**CN**)。司晓云 (**SI, Xiaoyun**)；中国广东省深圳市龙岗区坂田华为总部办公楼，**Guangdong** 518129 (**CN**)。

(81) 指定国（除另有指明，要求每一种可提供的国家保护）：AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国（除另有指明，要求每一种可提供的地区保护）：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,

(54) **Title** :DATA STREAM TYPE IDENTIFICATION METHOD AND RELATED DEVICES

(54) 发明名称：一种数据流类型识别方法及相关设备



S301 Obtain at least one firs: confidence level of a current data stream corresponding to at least one data stream type according to message characteristics of the current data stream and a behavior recognition mode
S302 Obtain at least one second confidence level of the current data stream corresponding to at least one data stream type according to the characteristic information of the current data stream and a content recognition model
S303 Determine the data stream type of the current data stream according to the at least one first confidence level and the at least one second confidence level
S304 Send information of the data stream type of the current data stream
S305 Generate a traffic control policy
S306 Send the traffic control policy
AA Terminal/forwarding device
BB Second device
CC First device
DD Update the content to the model

(57) **Abstract** : A data stream type identification method and related devices. The method comprises: obtaining at least one first confidence level of a current data stream corresponding to at least one data stream type according to message characteristics of the current data stream and a behavior recognition model, the behavior recognition model being a model obtained on the basis of a plurality of data stream samples; obtaining at least one second confidence level of the current data stream corresponding to at least one data stream type according to the characteristic information of the current data stream and a content recognition model, the content recognition model being a model obtained on the basis of one or more historical data streams; and determining the data stream type of the current data stream according to the at least one first confidence level and the at least one second confidence level. The data stream type can be more accurately identified using the present method.

(57) 摘要：一种数据流类型识别方法及相关设备。该方法根据当前数据流的报文特征和行为识别模型得到当前数据流的对应于至少一个数据流类型的至少一个第一置信度，行为识别模型为根据多个数据流样本得到的模型；根据当前数据流的特征信息和内容识别模型得到当前数据流的对应于至少一个数据流类型的至少一个第二置信度，内容识别模型为根据一条或多条历史数据流得到的模型；根据至少一个第一置信度和至少一个第二置信度确定当前数据流的数据流类型。采用该方法能够更准确地识别出数据流类型。

WO 2021/052379 A1

RO, RS, SE, SI, SK, SM, TR),OAPI （BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布：

- 包括国际检索报告（条约第21条（3））。

一种数据流类型识别方法及相关设备

本申请要求于2019年9月16日提交的申请号为201910872990.0、发明名称为“一种数据流类型识别方法及相关设备”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本发明涉及计算机技术领域和通信领域，进一步涉及人工智能(Artificial Intelligence, AI)在计算机技术领域和通信领域的应用，尤其涉及一种数据流类型识别方法及相关设备。

背景技术

随着计算机技术的迅猛发展，越来越多的企业使用私有的办公类应用进行办公，如桌面云、语音会议、视频会议等均属于私有的办公类应用。为了合理安排各业务的流量提高业务的可靠性，通常需要合理配置QoS优先级、实时选路等。而QoS优先级、实时选路等的前提是需要获知当前的办公应用属于哪种类型的应用(即哪种数据流类型)。

目前主要采用如下两种方式来获知办公应用的类型：第一种，基于人工识别的方式，主要是手工配置规则来匹配网络流量中的关键字(包括报头数据、载荷数据等)，从而识别出应用类型。然而，人工识别的方式耗时耗力，而且随着企业办公应用云化带来的动态性，出于安全通常会进行传输加密，这就导致人工难以设定出应用类型的识别规则，从而很难识别出应用类型。第二种，采用离线学习的方式，主要是预先采集样本数据，然后通过人工或使用第三方工具对样本数据进行标注，然后对标注过的样本数据使用机器学习或者神经网络的算法进行模型的离线训练。将离线训练好的模型，用于推理现网流量的应用类型。但是，企业应用每次进行改动或更新，都可能训练出的模型无法使用；另外，私有应用的实现没有规范，不同企业的同类型应用，适用的模型也不一样。即以上训练出的模型在用于改动、更新后的应用或另一个企业的应用的类型识别时，都可能出现识别准确率不高的问题。

如何快速准确地识别出办公应用的应用类型(也称数据流类型)是本领域的技术人员正在研究的技术问题。

发明内容

本发明实施例公开了一种数据流类型识别方法及相关设备，能够更准确地识别出数据流类型。

第一方面，本申请实施例提供一种数据流类型识别方法，该方法包括：

根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度，其中，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项；

根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度，其中，所述特征信息包括目的地址和协议类型，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据所述行为识别模型得到的；

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00464

根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

上述方法中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

结合第一方面，在第一方面的第一种可能的实现方式中，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，包括：

根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

在上述方法中，预先根据内容识别模型和行为识别模型各自对最终识别结果的影响程度，分别为两种模型配置置信度的权重（为行为识别模型配置的权重为第一置信度的权重，为内容识别模型配置的权重为第二置信度的权重），因此基于这两个置信度的权重计算出的综合置信度更能体现当前数据流的实际类型。另外，引入第一预设阈值来衡量对应的数据流类型是否可取，能够提高确定数据流类型的效率和准确性。

结合第一方面，或者第一方面的上述任一种可能的实现方式，在第一方面的第二种可能的实现方式中，所述方法还包括：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则向设备发送所述当前数据流的特征信息和所述第一数据流类型的信息，所述第二预设阈值大于所述第一预设阈值；

接收所述设备发送的第一信息，所述第一信息是所述设备根据所述当前数据流的特征信息和所述第一数据流类型的标识信息得到的；

根据所述第一信息更新所述内容识别模型，以得到新的内容识别模型。

结合第一方面，或者第一方面的上述任一种可能的实现方式，在第一方面的第三种可能的实现方式中，还包括：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，所述第二预设阈值大于所述第一预设阈值。

上述方法中，本申请发明人利用当前数据流的数据流类型的认定结果对内容识别模型进行校正。具体的，引入了第二预设阈值，在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值时，将当前数据流的相关信息发送给设备进行训练，以用于获得新的内容识别模型，以使得下一次的认定结果更加准确。

结合第一方面，或者第一方面的上述任一种可能的实现方式，在第一方面的第四种可能

Netskope Exhibit 1005
Netskope v. Fortinet IPR2026-00515
Page 00465

的实现方式中，所述根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，包括：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同，则将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

也即是说，如果在已经识别过的记录中，存在历史数据流的特征信息与当前数据流的特征信息相同但数据流类型不同的情况，则将记录中的数据流类型更新为当前数据流的第一数据流类型，这主要是为了适应云资源的弹性部署，例如，同样的云资源在前一段时间用于视频会议，在下一段时间用于桌面云；采用上述方法能够在下一段时间内及时将数据流类型更新为桌面云，从而在弹性部署云资源的情况下仍然可以准确识别出当前数据流的数据流类型。

结合第一方面，或者第一方面的上述任一种可能的实现方式，在第一方面的第五种可能的实现方式中，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型之后，还包括：

向运维支持系统 OSS 发送所述当前数据流的数据流类型的信息，所述当前数据流的数据流类型的所述信息用于所述 OSS 生成针对所述当前数据流的流量控制策略。

也即是说，在确定出当前数据流的数据流类型之后，将当前数据流类型的相关信息通知给 OSS 系统，这样 OSS 系统就可以基于当前数据流的数据流类型来生成针对所述当前数据流的流量控制策略，例如当前数据流的第一数据流类型为视频会议的视频流时，将其对应的流量控制策略定义为优先传输的策略，即当有多个数据流待传输时，优先传输该当前数据流。

结合第一方面，或者第一方面的上述任一种可能的实现方式，在第一方面的第六种可能的实现方式中，所述报文长度包括报文中以太帧长度、IP 长度、传输协议长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议 TCP 和/或用户数据报协议 UDP。

第二方面，本申请实施例提供一种数据流类型识别方法，该方法包括：

接收设备发送的当前数据流的特征信息和数据流类型的信息；

根据所述当前数据流的特征信息和数据流类型的信息生成第一信息；

向所述设备发送所述第一信息，以用于更新内容识别模型，所述内容识别模型用于得到对应于至少一个数据流类型的至少一个第二置信度，其中，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据行为识别模型得到的，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项。

上述方法中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度。

Netskope Ex.,1005
Netskope v. Fortinet IPR2026-00515
Page 00466

也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

第三方面，本申请实施例提供一种数据流类型识别设备，包括存储器和处理器，其中，所述存储器用于存储计算机程序，所述处理器调用所述计算机程序，用于执行如下操作：

根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度，其中，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项；

根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度，其中，所述特征信息包括目的地址和协议类型，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据所述行为识别模型得到的；

根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

上述方法中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本(或说典型)数据流的数据流类型，同样类型的数据流的某些特征(例如报文长度、报文传输速度等)可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面(如目的地址、协议类型等)对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

结合第三方面，在第三方面的第一种可能的实现方式中，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，具体为：

根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

在上述操作中，预先根据内容识别模型和行为识别模型各自对最终识别结果的影响程度，分别为两种模型配置置信度的权重(为行为识别模型配置的权重为第一置信度的权重，为内容识别模型配置的权重为第二置信度的权重)，因此基于这两个置信度的权重计算出的综合置信度更能体现当前数据流的实际类型。另外，引入第一预设阈值来衡量对应的数据流类型是否可取，能够提高确定数据流类型的效率和准确性。

结合第三方面，或者第三方面的上述任一种可能的实现方式，在第三方面的第二种可能的实现方式中，所述设备还包括收发器，所述处理器还用于：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则通过所述收发器向其他设备发送所述当前数据流的特征信息和所述第一数据流类型的信息，所述第二预设阈值大于所述第一预设阈值；

通过所述收发器接收所述其他设备发送的第一信息，所述第一信息是所述其他设备根据

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00467

所述当前数据流的特征信息和所述第一数据流类型的标识信息得到的；

根据所述第一信息更新所述内容识别模型，以得到新的内容识别模型。

结合第三方面，或者第三方面的上述任一种可能的实现方式，在第三方面的第三种可能的实现方式中，所述处理器还用于：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，所述第二预设阈值大于所述第一预设阈值。

上述方法中，本申请发明人利用当前数据流的数据流类型的认定结果对内容识别模型进行校正。具体的，引入了第二预设阈值，在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值时，将当前数据流的相关信息发送给设备进行训练，以用于获得新的内容识别模型，以使得下一次的认定结果更加准确。

结合第三方面，或者第三方面的上述任一种可能的实现方式，在第三方面的第四种可能的实现方式中，所述根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，具体为：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同，则将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

也即是说，如果在已经识别过的记录中，存在历史数据流的特征信息与当前数据流的特征信息相同但数据流类型不同的情况，则将记录中的数据流类型更新为当前数据流的第一数据流类型，这主要是为了适应云资源的弹性部署，例如，同样的云资源在前一段时间用于视频会议，在下一段时间用于桌面云；采用上述方法能够在下一段时间内及时将数据流类型更新为桌面云，从而在弹性部署云资源的情况下仍然可以准确识别出当前数据流的数据流类型。

结合第三方面，或者第三方面的上述任一种可能的实现方式，在第三方面的第五种可能的实现方式中，所述设备还包括收发器，所述处理器还用于，在所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型之后，通过所述收发器向运维支持系统 OSS 发送所述当前数据流的数据流类型的信息，所述当前数据流的数据流类型的所述信息用于所述 OSS 生成针对所述当前数据流的流量控制策略。

也即是说，在确定出当前数据流的数据流类型之后，将当前数据流类型的相关信息通知给 OSS 系统，这样 OSS 系统就可以基于当前数据流的数据流类型来生成针对所述当前数据流的流量控制策略，例如当前数据流的第一数据流类型为视频会议的视频流时，将其对应的流量控制策略定义为优先传输的策略，即当有多个数据流待传输时，优先传输该当前数据流。

结合第三方面，或者第三方面的上述任一种可能的实现方式，在第三方面的第六种可能的实现方式中，所述报文长度包括报文中以太帧长度、IP 长度、传输协议长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议 TCP 和/或用户数据报协议 UDP。

第四方面，本申请实施例提供一种数据流类型识别设备，包括存储器、处理器和收发器，其中，存储器用于存储计算机程序，所述处理器调用所述计算机程序，用于执行如下操作：

通过所述收发器接收其他设备发送的当前数据流的特征信息和数据流类型的信息；

根据所述当前数据流的特征信息和数据流类型的信息生成第一信息；

通过所述收发器向所述其他设备发送所述第一信息，以用于更新内容识别模型，所述

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00468

容识别模型用于得到对应于至少一个数据流类型的至少一个第二置信度，其中，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据行为识别模型得到的，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项。

上述操作中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本(或说典型)数据流的数据流类型，同样类型的数据流的某些特征(例如报文长度、报文传输速度等)可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面(如目的地址、协议类型等)对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

第五方面，本申请实施例提供一种数据流类型识别设备，该设备包括：

第一识别单元，用于根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度，其中，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项；

第二识别单元，用于根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度，其中，所述特征信息包括目的地址和协议类型，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据所述行为识别模型得到的；

确定单元，用于根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

上述设备中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本(或说典型)数据流的数据流类型，同样类型的数据流的某些特征(例如报文长度、报文传输速度等)可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面(如目的地址、协议类型等)对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

结合第五方面，在第五方面的第一种可能的实现方式中，所述确定单元，用于根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，具体为：

用于根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00469

据流的数据流类型为所述第一数据流类型。

在上述设备中,预先根据内容识别模型和行为识别模型各自对最终识别结果的影响程度,分别为两种模型配置置信度的权重(为行为识别模型配置的权重为第一置信度的权重,为内容识别模型配置的权重为第二置信度的权重),因此基于这两个置信度的权重计算出的综合置信度更能体现当前数据流的实际类型。另外,引入第一预设阈值来衡量对应的数据流类型是否可取,能够提高确定数据流类型的效率和准确性。

结合第五方面,或者第五方面的上述任一种可能的实现方式,在第五方面的第二种可能的实现方式中,所述设备还包括:

第一发送单元,用于在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值的情况下,向其他设备发送所述当前数据流的特征信息和所述第一数据流类型的信息,所述第二预设阈值大于所述第一预设阈值;

接收单元,用于接收所述其他设备发送的第一信息,所述第一信息是所述其他设备根据所述当前数据流的特征信息和所述第一数据流类型的标识信息得到的;

更新单元,用于根据所述第一信息更新所述内容识别模型,以得到新的内容识别模型。

结合第五方面,或者第五方面的上述任一种可能的实现方式,在第五方面的第三种可能的实现方式中,还包括:

更新单元,用于在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值的情况下,根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型,以得到新的内容识别模型,所述第二预设阈值大于所述第一预设阈值。

上述方法中,本申请发明人利用当前数据流的数据流类型的认定结果对内容识别模型进行校正。具体的,引入了第二预设阈值,在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值时,将当前数据流的相关信息发送给设备进行训练,以用于获得新的内容识别模型,以使得下一次的认定结果更加准确。

结合第五方面,或者第五方面的上述任一种可能的实现方式,在第五方面的第四种可能的实现方式中,所述根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型,以得到新的内容识别模型,具体为:

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同,则将所述第一记录的数据流类型更新为所述第一数据流类型,以获得第二记录;所述多条记录中的每一条记录包括特征信息和数据流类型;

对包括所述第二记录的多条记录进行训练,以得到新的内容识别模型。

也即是说,如果在已经识别过的记录中,存在历史数据流的特征信息与当前数据流的特征信息相同但数据流类型不同的情况,则将记录中的数据流类型更新为当前数据流的第一数据流类型,这主要是为了适应云资源的弹性部署,例如,同样的云资源在前一段时间用于视频会议,在下一段时间用于桌面云;采用上述方法能够在下一段时间内及时将数据流类型更新为桌面云,从而在弹性部署云资源的情况下仍然可以准确识别出当前数据流的数据流类型。

结合第五方面,或者第五方面的上述任一种可能的实现方式,在第五方面的第五种可能的实现方式中,所述设备还包括:

第二发送单元,用于在所述确定单元;f艮据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型之后,向运维支持系统 OSS 发送所述当前数据流的数据流类型的信息,所述当前数据流的数据流类型的所述信息用于所述 OSS 生成针对所述当前数据流的流量控制策略。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00470

也即是说，在确定出当前数据流的数据流类型之后，将当前数据流类型的相关信息通知给 OSS 系统，这样 OSS 系统就可以基于当前数据流的数据流类型来生成针对所述当前数据流的流量控制策略，例如当前数据流的第一数据流类型为视频会议的视频流时，将其对应的流量控制策略定义为优先传输的策略，即当有多个数据流待传输时，优先传输该当前数据流。

结合第五方面，或者第五方面的上述任一种可能的实现方式，在第五方面的第六种可能的实现方式中，所述报文长度包括报文中以太帧长度、IP 长度、传输协议长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议 TCP 和/或用户数据报协议 UDP。

第六方面，本申请实施例提供一种数据流类型识别设备，该设备包括：

接收单元，用于接收其他设备发送的当前数据流的特征信息和数据流类型的信息；

生成单元，用于根据所述当前数据流的特征信息和数据流类型的信息生成第一信息；

发送单元，用于向所述其他设备发送所述第一信息，以用于更新内容识别模型，所述内容识别模型用于得到对应于至少一个数据流类型的至少一个第二置信度，其中，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据行为识别模型得到的，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项。

上述设备中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

第七方面，本申请实施例提供一种计算机可读存储介质，所述计算机可读存储介质中存储有计算机程序，当其在处理器上运行时，实现第一方面或者第一方面的任意可能的实现方式所描述的方法。

第八方面，本申请实施例提供一种计算机程序产品，所述计算机程序产品存储在存储器上，当所述计算机程序产品在处理器上运行时，实现第一方面或者第一方面的任意可能的实现方式所描述的方法。

第九方面，本申请实施例提供了一种数据流类型识别系统，该系统包括第一设备和第二设备，其中，所述第二设备为上述第三方面，或者第三方面的任一可能的实现方式，或者第五方面，或者第五方面的任一可能的实现方式所描述的数据流类型识别设备；所述第一设备为上述第四方面，或者第四方面的任一可能的实现方式，或者第六方面，或者第六方面的任一可能的实现方式所描述的数据流类型识别设备。

本申请实施例中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00471

型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

附图说明

以下对本发明实施例用到的附图进行介绍。

图 1 是本发明实施例提供的一种数据流类型的识别系统的架构示意图；

图 2A 是本发明实施例提供的内容识别模型和行为识别模型的场景示意图；

图 2B 是本发明实施例提供的一种分类模型的结构示意图；

图 2C 是本发明实施例提供的一种分类模型的结构示意图；

图 2D 是本发明实施例提供的一种分类模型的结构示意图；

图 3 是本发明实施例提供的一种数据流类型识别方法的流程示意图；

图 4 是本发明实施例提供的数据流 a 和数据流 b 的场景示意图；

图 5 是本发明实施例提供的数据流记录的样例示意图；

图 6 是本发明实施例提供的数据流记录的样例示意图；

图 7 是本发明实施例提供的一种第二设备的结构示意图；

图 8 是本发明实施例提供的一种第一设备的结构示意图；

图 9 是本发明实施例提供的又一种第二设备的结构示意图；

图 10 是本发明实施例提供的又一种第一设备的结构示意图。

具体实施方式

下面结合本发明实施例中的附图对本发明实施例进行描述。

请参见图 1，图 1 是本发明实施例提供的一种数据流类型的识别系统的架构示意图，该系统包括运维支持系统（operational system support，OSS）101、服务器 102、转发设备 103、终端 104，其中，终端 104 的数量可以为多个，图中仅以 6 个为例进行了示意，终端 104 上用于运行各种应用，例如，视频会议应用、语音会议应用、桌面云应用等等，不同的应用所产生的数据流的数据流类型（也称应用类型）往往不同。本申请实施例中，终端 104 所产生的数据流需要先通过转发设备 103 发往目的设备，其中，转发设备可以包括路由器、交换机等，转发设备 103 的数量可以是一个也可以是多个，例如，存在一个路由器和三个交换机；再如，仅存在一个交换机；再如，存在三个交换机，等等。上述服务器 102 可以为一个服务器或者多个服务器组成的服务器集群。

其中，终端 104 所产生的数据流在该终端 104 上该如何发送，以及在转发设备 103 上该如何转发，都可以按照 OSS 101 生成的流量控制策略来进行，例如，当流量控制策略规定视频会议应用所产生的数据流具有最高优先级时，假若在终端 104，或者转发设备 103 上有包括视频会议应用所产生的数据流在内的多种数据流需要发送时，会优先传输视频会议应用所产生的数据流。需要说明的是，该流量控制策略由 OSS 101 基于当前数据流的数据流类型来生成的。本申请实施例中，OSS 101 生成流量控制策略所用到的当前数据流的数据流类型是通过第二设备来确定的。

如图 2A 所示，第二设备在确定当前数据流的数据流类型的时候需要用到行为识别模型

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00472

和内容识别模型，其中，模型的参数可以包括但不限于置信度权重向量（w1，w2）、数据流类型的第一预设阈值01、第二预设阈值02，其中，第一预设阈值也可以称为分类阈值，用于衡量是否将数据流类型划分到某一类；第二预设阈值也称为模型更新阈值，用于衡量何时对内容识别模型进行更新。这些模型参数将在后续的方法流程中做更具体的阐述。其中，内容识别模型在对当前数据流进行数据流类型识别时的输入可以包括特征信息（如目的IP、目的端口、协议类型等），行为识别模型在对当前数据流进行数据流类型识别时的输入可以包括报文信息（如报文长度、报文传输速度、报文间隔时间、报文方向等）。第二设备对内容识别模型得到的置信度和行为识别模型得到的置信度基于置信度权重向量（w1，w2）得到最终的数据流类型；这个过程中，如果通过第二预设阈值02确定内容识别模型需要进行更新，则获取更新内容识别模型所需的参数。

可选的，更新所需要用到的参数可以是第二设备通过对一些数据流的相关信息进行训练得到，也可以是第一设备通过对一些数据流的相关信息进行训练得到，然后发送给第二设备。

本申请实施例中，该第二设备可以是上述OSS 101，也可以是上述服务器102，也可以是上述转发设备103；另外，该第一设备可以是上述OSS 101，也可以是上述服务器102，也可以是上述转发设备103。结合以上描述，该第一设备与该第二设备可以是相同的设备，也可以是不同的设备。并且，当第一设备和第二设备都不是上述服务器102时，可以认为图1所示的架构的中不存在服务器102。

需要说明的是，上述内容识别模型实质上是一个分类模型，如图2B所示，分类模型可以是树模型，如图2C所示，分类模型还可以是神经网络模型，如图2D所示，分类模型还可以是支持向量机（support vector machine，SVM）模型，当分类模型还可以是其他形式的模型。可选的，由于上述内容识别模型是通过提取输入的向量（如目的互联网协议（internet protocol，IP）地址、目的端口号、协议类型等特征信息）的特征来分类得到的，因此该内容识别模型可以将同一目的IP地址，同一协议类型，相同端口号的数据流识别为相同的数据流类型；将同一网段，同一协议类型，相近端口号的数据流识别为相同的数据流类型；将目的端口号是20（文件传输协议（file transfer protocol，FTP）的知名端口号）的传输控制协议（transmission control protocol，TCP）流量识别为下载的数据流类型。

另外，该第二设备在识别当前数据流的数据流类型时需要用到的一些关于当前数据流的信息，都可以由终端104或者其他设备发送给第二设备，或者第二设备自己采集得到。

请参见图3，图3是本发明实施例提供的一种数据流类型识别方法，该方法可以基于图1所示的架构来实现，该方法包括但不限于如下步骤：

步骤S301：第二设备根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度。

具体地，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；可选的，该多个数据流样本可以为离线样本，即该行为识别模型可以为离线训练得到的模型。该多个数据流样本还可以为预先挑选的比较典型（或说有代表性）的样本，例如，视频会议应用的数据流的报文大多时候的报文长度比较长，但是也会偶尔出现报文长度比较短的时候，相比而言，报文长度比较长更能反映当前数据流为视频会议应用的数据流，因此，在选择关于视频会议应用的数据流时，尽量选择报文长度比较长的这种有代表性的作为数据流样本。可选的，该多个数据流样本的数据流类型可以是认为确定的，即人工打标签。由于所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，因此该行为识别模型

Netskope Exhibit 1005
Netskope v. Fortinet IPR2026-00515
Page 00473

能够反映一个数据流中的报文特征与数据流类型之间的一些关系，因此，当向该行为识别模型输入当前数据流的报文特征时，其能够一定程度上预测该当前数据流属于某个或者某些数据流类型的倾向（或者说概率），体现倾向（或者说概率）的参数也可以称为置信度。

本申请实施例中，所述报文特征可以包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项，可选的，所述报文长度包括报文中以太帧长度、IP 长度、传输协议长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议 TCP 和/或用户数据报协议 UDP。当然，报文特征除了包括这里举例的特征外，还可以包括其他特征，例如，报文长度、报文传输速度、报文间隔时间和报文方向中的最大值、最小值、均值、方差、分位数等。该报文特征可以是以向量的形式输入到行为识别模型中，例如可以是（报文长度，报文传输速度，报文间隔时间）这种形式。另外，本申请实施例中的数据流类型还可以称之为应用类型。

第一种可能情况中，数据流类型可能有 N 个，N 大于或者等于 1，本申请实施例可以估计（或者说预测）出当前数据流属于该 N 个数据流类型中每个类型的置信度，即得到所述当前数据流的对应于 N 个数据流类型的 N 个第一置信度，举例来说，假若该 N 个数据流类型指的是视频会议的数据流类型、语音会议的数据流类型、桌面云的数据流类型，那么，需要通过行为识别模型估计当前数据流属于视频会议的数据流类型的第一置信度、属于语音会议的数据流类型的第一置信度、属于桌面云的数据流类型的第一置信度。假若 N 个数据流类型指的是视频会议的数据流类型，则需要通过行为识别模型估计当前数据流属于视频会议的数据流类型的第一置信度。

第二种可能情况，数据流类型可能有多个，但本申请实施例重点关注其中一个数据流类型，因此本申请实施例只估计（或者说预测）当前数据流属于该重点关注的数据流类型的置信度，即得到所述当前数据流的对应于一个数据流类型的一个第一置信度，举例来说，假若该多个数据流类型指的是视频会议的数据流类型、语音会议的数据流类型、桌面云的数据流类型，但本申请实施例只关注视频会议的数据流类型，因此只需通过行为识别模型估计当前数据流属于视频会议的数据流类型的第一置信度即可。

步骤 S202：第二设备根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度。

具体地，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型。可选的，该一条或者多条历史数据流可以为在线数据流，即在此之前的一段时间内持续产生的一条或者多条数据流，所述历史数据流的数据流类型是由上述行为识别模型识别得到的，即该内容识别模型可以为在线训练得到的模型。由于所述内容识别模型为根据一条或者多条历史数据流的特征信息和数据流类型得到的模型，因此该内容识别模型能够反映一个数据流中的特征信息与数据流类型之间的一些关系，因此，当向该内容识别模型输入当前数据流的特征信息时，其能够一定程度上预测该当前数据流属于某个或者某些数据流类型的倾向（或者说概率），体现倾向（或者说概率）的参数也可以称为置信度。

本申请实施例中，所述特征信息可以包括目的地址、协议类型、端口号中的一项或者多项，其中，目的地址可以为 IP 地址，也可以为目的 MAC 地址，还可以为其他形式的地址；当然，特征信息除了包括这里举例的特征外，还可以包括其他特征。进一步地，这里的特征信息可以是针对目标的信息，例如，目标 IP，目标端口等。该特征信息可以是以向量的形式输入到内容识别模型中，可以是（ip，port，protocol）这种形式，例如可以为（10.29.74.5，8443，6）。也可以是（mac，port，protocol）这种形式，例如可以为（05FA 1525 EEFF，843...

Netskope Ex.,1005
Netskope v. Fortinet IPR2026-00515
Page 00474

6)。当然还可以是其他形式，此处不再一一举例。

第一种可能情况中，数据流类型可能有 N 个，本申请实施例可以估计 (或者说预测) 出当前数据流属于该 N 个数据流类型中每个类型的置信度，即得到所述当前数据流的对应于 N 个数据流类型的 N 个第二置信度，举例来说，假若该 N 个数据流类型指的是视频会议的数据流类型、语音会议的数据流类型、桌面云的数据流类型，那么，需要通过内容识别模型估计当前数据流属于视频会议的数据流类型的第二置信度、属于语音会议的数据流类型的第二置信度、属于桌面云的数据流类型的第二置信度。假若 N 个数据流类型指的是视频会议的数据流类型，则需要通过内容识别模型估计当前数据流属于视频会议的数据流类型的第二置信度。

第二种可能情况，数据流类型可能有多个，但本申请实施例重点关注其中一个数据流类型，因此本申请实施例只估计 (或者说预测) 当前数据流属于该重点关注的数据流类型的置信度，即得到所述当前数据流的对应于一个数据流类型的一个第二置信度，举例来说，假若该多个数据流类型指的是视频会议的数据流类型、语音会议的数据流类型、桌面云的数据流类型，但本申请实施例只关注视频会议的数据流类型，因此只需通过内容识别模型估计当前数据流属于视频会议的数据流类型的第二置信度即可。

步骤 S303：第二设备根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

具体地，因为至少一个第一置信度能够在一定程度上表征当前数据流的数据类型倾向，至少一个第二置信度也可以在一定程度上表征当前数据流的数据流类型倾向，因此对两者进行综合考虑可以获得更准确可信的数据流类型倾向，从而得出当前数据流的数据流类型。

一种可选的方案中，得到的哪种数据流类型的综合置信度最大，则将该数据流类型确定为当前数据流的数据流类型，例如，根据视频会议的数据流类型的第一置信度和视频会议的数据流类型的第二置信度确定出的视频会议的数据流类型的综合置信度为 0.7；根据语音会议的数据流类型的第一置信度和语音会议的数据流类型的第二置信度确定出的语音会议的数据流类型的综合置信度为 0.2；根据桌面云的数据流类型的第一置信度和桌面云的数据流类型的第二置信度确定出的桌面云的数据流类型的综合置信度为 0.1；由于视频会议的数据流类型的综合置信度最大，因此将当前数据流的数据流类型确定为视频会议的数据流类型。

又一种可选的方案中，所述根据至少一个第一置信度和至少一个第二置信度确定所述当前数据流的数据流类型，可以具体为：根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个，也即是说，该至少一个数据流类型中的每个数据流类型均满足这里的第一数据流类型的特征。若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型，例如，假若对应于视频会议的数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为视频会议的数据流类型；假若对应于桌面云的数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为桌面云的数据流类型。

举例来说，假若置信度权重向量 (w1，w2) 为 (0.4，0.6)，即第一置信度权重 (也可以认为是行为识别模型的权重) 为 0.6，第二置信度权重 (也可以认为是内容识别模型的权重) 为 0.4、数据流类型的第一预设阈值01等于 0.5。在第二设备进行数据流类型识别之初，由于内容识别模型未得到充分训练，因此行为识别模型能够对输入的数据流进行数据流类型的识别，而内容识别模型无法对输入的流数据进行数据流类型的识别，因此起初内容识别模型

Exhibit 1005
Netskope v. Fortinet IPR2026-00515
Page 00475

别的对应于任何数据流类型的置信度均为0。

假若存在两条桌面云的数据流，如图4所示，横轴代表数据流序号，纵轴代表报文长度，报文长度大于0为上行报文，报文长度小于0为下行报文；数据流a和数据流b均为桌面云的数据流类型，其中，数据流b有上行的报文，相对来说更能代表桌面云场景的特性，因此认为数据流b的报文行为是典型的行为。数据流a在很长一段时间内没有上行报文，无法明显表明其为桌面云场景，因此认为数据流a的报文行为是不典型的行为；行为识别模型通常是通过对典型行为的数据流进行训练得到，因此行为识别模型能够识别出数据流b的数据流类型，但无法识别出数据流a的数据流类型。数据流a和数据流b的特征信息如下。

数据流a的协议类型是TCP，目的IP地址是10.129.74.5，目的端口号是8443。

数据流b的协议类型是TCP，目的IP地址是10.129.56.39，目的端口号是443。

那么针对数据流a，内容识别模型识别是桌面云的数据流类型、语音会议的数据流类型、视频会议的数据流类型的第二置信度均是0。行为识别模型基于报文特征识别是桌面云的数据流类型的第一置信度是0.5、识别是语音会议的数据流类型的第一置信度均是0、识别是视频会议的数据流类型的第一置信度是0。因此，对应于这三种数据流类型的综合置信度如下。

桌面云：0*0.4+0.5*0.6=0.3，小于01，因此当前数据流不是桌面云的数据流类型。

语音会议：0*0.4+0*0.6=0，小于01，因此当前数据流不是语音会议的数据流类型。

视频会议：0*0.4+0*0.6=0，小于01，因此当前数据流不是视频会议的数据流类型。

那么针对数据流b，内容识别模型识别是桌面云的数据流类型、语音会议的数据流类型、视频会议的数据流类型的第二置信度均是0。行为识别模型基于报文特征识别是桌面云的数据流类型的第一置信度是0.9、识别是语音会议的数据流类型的第一置信度均是0、识别是视频会议的数据流类型的第一置信度是0。因此，对应于这三种数据流类型的综合置信度如下。

桌面云：0*0.4+0.9*0.6=0.54，大于01，因此当前数据流是桌面云的数据流类型。

语音会议：0*0.4+0*0.6=0，小于01，因此当前数据流不是语音会议的数据流类型。

视频会议：0*0.4+0*0.6=0，小于01，因此当前数据流不是视频会议的数据流类型。

本申请实施例中，该内容识别模型还可以进行更新，下面提供两种不同的更新方案。

方案一，若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值且小于第二预设阈值02，则第二设备向第一设备发送所述当前数据流的特征信息和所述第一数据流类型的信息，所述第二预设阈值大于所述第一预设阈值。例如，针对数据流a来说，对应于桌面云的数据流类型的综合置信度0.3不在区间（01，02）内，因此无需向第一设备发送所述当前数据流的特征信息和桌面云的数据流类型的信息；再如，针对数据流b来说，对应于桌面云的数据流类型的综合置信度0.54不在区间（01，02）内，因此需要向第一设备发送所述当前数据流的特征信息（如目的IP地址10.129.56.39、目的端口号443、协议类TCP型）和桌面云的数据流类型的信息（如名称、标识等信息）。

相应的，第一设备接收第二设备发送的所述当前数据流的特征信息和所述第一数据流类型的信息，即该第一设备上就多了一条数据流记录，如图5，多了一条针对数据流b的记录。然后，该第一设备根据当前数据流的特征信息和所述第一数据流类型的信息得到的第一信息，例如，对计算置信度有影响的一些参数。可选的，第一信息属于模型文件，因此可以以模型文件的方式传输，常见开源keras库的ai模型文件是h5文件/json文件；开源skleam库的ai模型文件是pkl/m文件。这些文件都是二进制的，用于保存模型的结构、模型的权重。

接着，第二设备接收第一设备发送的第一信息，然后根据所述第一信息更新所述内容识别模型，以得到新的内容识别模型。由于数据流b的记录中，目的IP地址为10.129.56.39、

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00476

目的端口号为443、协议类型为TCP共同对应的是桌面云的数据流类型，因此，更新后的内容识别模型再对输入的数据流b进行估计时，估计出的对应于桌面云的数据流类型的第二置信度为1。可选的，由于数据流a的特征信息与数据流b的特征信息有相似之处，例如，目的IP地址在同一网段，端口号相似，协议类型相似，因此，更新后的内容识别模型再对输入的数据流a进行估计时，估计的结果会更接近对数据流b的估计结果，例如，估计出的对应于桌面云的数据流类型的第二置信度可能为0.6。

后续网络中再出现数据流a和数据流b需要估计数据流类型时，执行流程如下。

置信度权重向量 (w1，w2)、第一预设阈值01、第二预设阈值02依旧不变。

那么针对数据流a，内容识别模型识别是桌面云的数据流类型的第二置信度为0.6，识别是语音会议的数据流类型、视频会议的数据流类型的第二置信度均是0。行为识别模型基于报文特征识别是桌面云的数据流类型的第一置信度是0.5、识别是语音会议的数据流类型的第一置信度均是0、识别是视频会议的数据流类型的第一置信度是0。因此，对应于这三种数据流类型的综合置信度如下。

桌面云：0.6*0.4+0.5*0.6=0.54 ，大于01，因此当前数据流是桌面云的数据流类型。

语音会议：0*0.4+0*0.6=0 ，小于01，因此当前数据流不是语音会议的数据流类型。

视频会议：0*0.4+0*0.6=0 ，小于01，因此当前数据流不是视频会议的数据流类型。

由于识别是桌面云的数据流类型的第二置信度0.54在区间 (01，02) 内，因此需要向第一设备发送所述当前数据流的特征信息和桌面云的数据流类型的信息，以用于后续对内容识别模型进行更新 (更新原理前面已有介绍，此处不再赘述)。

那么针对数据流b，内容识别模型识别是桌面云的数据流类型的数据流类型的第二置信度是1、识别是语音会议的数据流类型、视频会议的数据流类型的第二置信度均是0。行为识别模型基于报文特征识别是桌面云的数据流类型的第一置信度是0.9、识别是语音会议的数据流类型的第一置信度均是0、识别是视频会议的数据流类型的第一置信度是0。因此，对应于这三种数据流类型的综合置信度如下。

桌面云：1*0.4+0.9*0.6=0.94 ，大于01，因此当前数据流是桌面云的数据流类型。

语音会议：0*0.4+0*0.6=0 ，小于01，因此当前数据流不是语音会议的数据流类型。

视频会议：0*0.4+0*0.6=0 ，小于01，因此当前数据流不是视频会议的数据流类型。

由于识别是桌面云的数据流类型的第二置信度0.94不在区间 (01，02) 内，因此不需要向第一设备发送所述当前数据流的特征信息和桌面云的数据流类型的信息。

在网络中存在云化的弹性部署场景，例如，目的IP地址为10.129.56.39，目的端口号为443，协议类型为TCP的云资源从为桌面云提供服务变成为视频会议提供服务。针对这种场景，上述第一设备根据当前数据流的特征信息和所述第一数据流类型的信息得到的第一信息，可以包括：若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同，则将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；然后对包括所述第二记录的多条记录进行训练，以得到第一信息。

举例来说，数据流c的协议类型是TCP，目的IP地址是10.129.56.39，目的端口号是443。

那么针对数据流c，内容识别模型识别是桌面云的数据流类型的数据流类型的第二置信度是1、识别是语音会议的数据流类型、视频会议的数据流类型的第二置信度均是0。行为识别模型基于报文特征识别是桌面云的数据流类型的第一置信度是0、识别是语音会议的数据流类型的第一置信度是0、识别是视频会议的数据流类型的第一置信度是0.9。因此，对应于

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00477

这三种数据流类型的综合置信度如下。

桌面云：1*0.4+0*0.6=0.4 ，小于01，因此当前数据流不是桌面云的数据流类型。

语音会议：0*0.4+0*0.6=0 ，小于01，因此当前数据流不是语音会议的数据流类型。

视频会议：0*0.4+0.9*0.6=0.54 ，大于01，因此当前数据流是视频会议的数据流类型。

由于识别是视频会议的数据流类型的第二置信度0.54在区间（01，02）内，因此需要向第一设备发送所述当前数据流的特征信息和桌面云的数据流类型的信息。相应的，第一设备接收第二设备发送的所述当前数据流的特征信息和所述第一数据流类型的信息，即该第一设备上就有了一条数据流记录，如图6，多了一条针对数据流c的记录。

对比图6和图5就可以发现，针对数据流c的记录与针对数据流b的记录相比，数据流类型（即应用类别）不同但特征信息（如目的IP地址、目的端口号和协议类型）相同，因此修改已有的记录，使得协议类型TCP、目的IP地址10.129.56.39、目的端口号443共同对应的数据流类型为视频会议的数据流类型，而不再是桌面云对应的数据流类型，修改前的记录即为第一记录，修改后的记录即为第二记录。

可以理解，将第一记录更新为第二记录之后，基于第二记录训练得到的第一信息来更新内容识别模型后，上述数据流c再次输入到更新后的内容识别模型时，识别其为桌面云的数据流类型的第二置信度为0，识别其为视频会议的数据流类型的第二置信度为1。

方案二，若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，第二设备无需将当前数据流的特征信息和所述第一数据流类型的信息发送给第一设备，则而是由自己根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，所述第二预设阈值大于所述第一预设阈值。可选的，若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同，则第二设备将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。为了便于理解，具体原理可以参照上述方案一，由该第二设备替代上述方案一种第一设备执行的操作即可。

步骤S304：第二设备向运维支持系统OSS发送所述当前数据流的数据流类型的信息。

具体地，第二设备可以在每次确定出当前数据流的数据流类型时，向OSS发送当前数据流的数据流类型的信息，例如，第一次生成数据流a的数据流类型时，向OSS发送数据流a的数据流类型的信息，第一次生成数据流b的数据流类型时，向OSS发送数据流b的数据流类型的信息，第二次生成数据流a的数据流类型时，向OSS发送数据流a的数据流类型的信息，第二次生成数据流b的数据流类型时，向OSS发送数据流b的数据流类型的信息；生成数据流c的数据流类型时，向OSS发送数据流c的数据流类型的信息。可以理解的是，假若第二设备为该OSS，则不需要执行上述步骤S304。

步骤S305：OSS根据当前数据流的数据流类型的信息生成针对当前数据流的流量控制策略。例如，如果当前数据流的数据流类型信息表明该当前数据流为视频桌面云的数据流类型，或者视频会议的数据流类型，则将当前数据流定义为高优先级的QoS。

步骤S306：OSS向转发设备或者终端发送所述流量控制策略。

具体地，该转发设备103可以为路由器或者交换机等设备，该终端104为输出上述当前数据流的设备。假若转发设备或者终端根据流量控制策略获知当前数据流属于高优先级的QoS，则在发现有多种数据流待发送时，优先发送被配置为高优先级的当前数据流。

在图3所描述的方法中，涉及行为识别模型和内容识别模型，行为识别模型为多个或

Netskope Ex1005
Netskope v. Fortinet IPR2026-00515
Page 00478

据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

上述详细阐述了本发明实施例的方法，下面提供了本发明实施例的装置。

请参见图7，图7是本发明实施例提供的一种数据流类型识别设备70的结构示意图，该设备70可以为图3所示方法实施例中的第二设备，该设备70可以包括第一识别单元701、第二识别单元702和确定单元703，其中，各个单元的详细描述如下。

第一识别单元701，用于根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度，其中，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项；

第二识别单元702，用于根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度，其中，所述特征信息包括目的地址和协议类型，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据所述行为识别模型得到的；

确定单元703，用于根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

上述设备中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

在一种可能的实现方式中，所述确定单元703，用于根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，具体为：

用于根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

在上述设备中，预先根据内容识别模型和行为识别模型各自对最终识别结果的影响程度，

Netskope Exhibit 1005
Netskope v. Fortinet IPR2026-00515
Page 00479

分别为两种模型配置置信度的权重 (为行为识别模型配置的权重为第一置信度的权重，为内容识别模型配置的权重为第二置信度的权重),因此基于这两个置信度的权重计算出的综合置信度更能体现当前数据流的实际类型。另外，引入第一预设阈值来衡量对应的数据流类型是否可取，能够提高确定数据流类型的效率和准确性。

在又一种可能的实现方式中，所述设备 70 还包括：

第一发送单元，用于在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值的情况下，向其他设备 (如图 3 所示方法实施例中的第一设备) 发送所述当前数据流的特征信息和所述第一数据流类型的信息，所述第二预设阈值大于所述第一预设阈值；

接收单元，用于接收所述其他设备 (如图 3 所示方法实施例中的第一设备) 发送的第一信息，所述第一信息是所述其他设备 (如图 3 所示方法实施例中的第一设备) 根据所述当前数据流的特征信息和所述第一数据流类型的标识信息得到的；

更新单元，用于根据所述第一信息更新所述内容识别模型，以得到新的内容识别模型。

在又一种可能的实现方式中，该设备 70 还包括：

更新单元，用于在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值的情况下，根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，所述第二预设阈值大于所述第一预设阈值。

上述方法中，本申请发明人利用当前数据流的数据流类型的认定结果对内容识别模型进行校正。具体的，引入了第二预设阈值，在对应于所述第一数据流类型的所述综合置信度小于第二预设阈值时，将当前数据流的相关信息发送给设备进行训练，以用于获得新的内容识别模型，以使得下一次的认定结果更加准确。

在又一种可能的实现方式中，所述根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，具体为：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同，则将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

也即是说，如果在已经识别过的记录中，存在历史数据流的特征信息与当前数据流的特征信息相同但数据流类型不同的情况，则将记录中的数据流类型更新为当前数据流的第一数据流类型，这主要是为了适应云资源的弹性部署，例如，同样的云资源在前一段时间用于视频会议，在下一段时间用于桌面云；采用上述方法能够在下一段时间内及时将数据流类型更新为桌面云，从而在弹性部署云资源的情况下仍然可以准确识别出当前数据流的数据流类型。

在又一种可能的实现方式中，所述设备 70 还包括：

第二发送单元，用于在所述确定单元根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型之后，向运维支持系统 OSS 发送所述当前数据流的数据流类型的信息，所述当前数据流的数据流类型的所述信息用于所述 OSS 生成针对所述当前数据流的流量控制策略。

也即是说，在确定出当前数据流的数据流类型之后，将当前数据流类型的相关信息通知给 OSS 系统，这样 OSS 系统就可以基于当前数据流的数据流类型来生成针对所述当前数据流的流量控制策略，例如当前数据流的第一数据流类型为视频会议的视频流时，将其对应的流量控制策略定义为优先传输的策略，即当有多个数据流待传输时，优先传输该当前数据流。

在又一种可能的实现方式中，所述报文长度包括报文中以太帧长度、IP 长度、传输层协议

Netskope Ex1005
Netskope v. Fortinet IPR2026-00515
Page 00480

长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议 TCP 和/或用户数据报协议 UDP。

需要说明的是，各个单元的实现还可以对应参照图 3 所示的方法实施例的相应描述。

请参见图 8，图 8 是本发明实施例提供的一种数据流类型识别设备 80 的结构示意图，该设备 80 可以为图 3 所示方法实施例中的第一设备，该设备 80 可以包括接收单元 801、生成单元 802 和发送单元 803，其中，各个单元的详细描述如下。

接收单元 801，用于接收其他设备（如图 3 所示方法实施例中的第二设备）发送的当前数据流的特征信息和数据流类型的信息；

生成单元 802，用于根据所述当前数据流的特征信息和数据流类型的信息生成第一信息；

发送单元 803，用于向所述其他设备（如图 3 所示方法实施例中的第二设备）发送所述第一信息，以用于更新内容识别模型，所述内容识别模型用于得到对应于至少一个数据流类型的至少一个第二置信度，其中，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据行为识别模型得到的，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项。

上述设备 80 中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

需要说明的是，各个单元的实现还可以对应参照图 3 所示的方法实施例的相应描述。

请参见图 9，图 9 是本发明实施例提供的一种设备 90，该设备 90 可以为图 3 所示方法实施例中的第二设备，该设备 90 包括处理器 901、存储器 902 和收发器 903，所述处理器 901、存储器 902 和收发器 903 通过总线相互连接。

存储器 902 包括但不限于是随机存储记忆体（random access memory，RAM）、只读存储器（read-only memory，ROM）、可擦除可编程只读存储器（erasable programmable read only memory，EPROM）、或便携式只读存储器（compact disc read-only memory，CD-ROM），该存储器 902 用于相关计算机程序及数据。收发器 903 用于接收和发送数据。

处理器 901 可以是一个或多个中央处理器（central processing unit，CPU），在处理器 901 是一个 CPU 的情况下，该 CPU 可以是单核 CPU，也可以是多核 CPU。

处理器 901 读取所述存储器 902 中存储的计算机程序代码，用于执行以下操作：

根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度，其中，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项；

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00481

根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度，其中，所述特征信息包括目的地址和协议类型，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据所述行为识别模型得到的；

根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

上述操作中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

在一种可能的实现方式中，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，具体为：

根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

在上述操作中，预先根据内容识别模型和行为识别模型各自对最终识别结果的影响程度，分别为两种模型配置置信度的权重（为行为识别模型配置的权重为第一置信度的权重，为内容识别模型配置的权重为第二置信度的权重），因此基于这两个置信度的权重计算出的综合置信度更能体现当前数据流的实际类型。另外，引入第一预设阈值来衡量对应的数据流类型是否可取，能够提高确定数据流类型的效率和准确性。

在一种可能的实现方式中，所述处理器还用于：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则通过所述收发器向其他设备（如图3所示方法实施例中的第一设备）发送所述当前数据流的特征信息和所述第一数据流类型的信息，所述第二预设阈值大于所述第一预设阈值；

通过所述收发器接收所述其他设备（如图3所示方法实施例中的第一设备）发送的第一信息，所述第一信息是所述其他设备（如图3所示方法实施例中的第一设备）根据所述当前数据流的特征信息和所述第一数据流类型的标识信息得到的；

根据所述第一信息更新所述内容识别模型，以得到新的内容识别模型。

在又一种可能的实现方式中，所述处理器还用于：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，所述第二预设阈值大于所述第一预设阈值。

上述方法中，本申请发明人利用当前数据流的数据流类型的认定结果对内容识别模型进行校正。具体的，引入了第二预设阈值，在对应于所述第一数据流类型的所述综合置信度小

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00482

于第二预设阈值时，将当前数据流的相关信息发送给设备进行训练，以用于获得新的内容识别模型，以使得下一次的认定结果更加准确。

在又一种可能的实现方式中，所述根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，具体为：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同，则将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

也即是说，如果在已经识别过的记录中，存在历史数据流的特征信息与当前数据流的特征信息相同但数据流类型不同的情况，则将记录中的数据流类型更新为当前数据流的第一数据流类型，这主要是为了适应云资源的弹性部署，例如，同样的云资源在前一段时间用于视频会议，在下一段时间用于桌面云；采用上述方法能够在下一段时间内及时将数据流类型更新为桌面云，从而在弹性部署云资源的情况下仍然可以准确识别出当前数据流的数据流类型。

在又一种可能的实现方式中，所述处理器还用于，在所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型之后，通过所述收发器向运维支持系统 OSS 发送所述当前数据流的数据流类型的信息，所述当前数据流的数据流类型的所述信息用于所述 OSS 生成针对所述当前数据流的流量控制策略。

也即是说，在确定出当前数据流的数据流类型之后，将当前数据流类型的相关信息通知给 OSS 系统，这样 OSS 系统就可以基于当前数据流的数据流类型来生成针对所述当前数据流的流量控制策略，例如当前数据流的第一数据流类型为视频会议的视频流时，将其对应的流量控制策略定义为优先传输的策略，即当有多个数据流待传输时，优先传输该当前数据流。

在又一种可能的实现方式中，所述报文长度包括报文中以太帧长度、IP 长度、传输协议长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议 TCP 和/或用户数据报协议 UDP。

需要说明的是，各个操作的实现还可以对应参照图 3 所示的方法实施例的相应描述。

请参见图 10，图 10 是本发明实施例提供的一种设备 100，该设备 100 可以为图 3 所示方法实施例中的第一设备，该设备 100 包括处理器 1001、存储器 1002 和收发器 1003，所述处理器 1001、存储器 1002 和收发器 1003 通过总线相互连接。

存储器 1002 包括但不限于是随机存储记忆体（random access memory，RAM）、只读存储器（read-only memory，ROM）、可擦除可编程只读存储器（erasable programmable read only memory，EPROM）、或便携式只读存储器（compact disc read-only memory，CD-ROM），该存储器 1002 用于相关计算机程序及数据。收发器 1003 用于接收和发送数据。

处理器 1001 可以是一个或多个中央处理器（central processing unit，CPU），在处理器 1001 是一个 CPU 的情况下，该 CPU 可以是单核 CPU，也可以是多核 CPU。

处理器 1001 读取所述存储器 1002 中存储的计算机程序代码，用于执行以下操作：

通过所述收发器接收其他设备（如图 3 所示方法实施例中的第二设备）发送的当前数据流的特征信息和数据流类型的信息；

根据所述当前数据流的特征信息和数据流类型的信息生成第一信息；

通过所述收发器向所述其他设备（如图 3 所示方法实施例中的第二设备）发送所述第一信息，以用于更新内容识别模型，所述内容识别模型用于得到对应于至少一个数据流类型的

Exhibit 1005
Netskope v. Fortinet IPR2026-00515
Page 00483

至少一个第二置信度，其中，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据行为识别模型得到的，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项。

上述操作中，涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

需要说明的是，各个操作的实现还可以对应参照图3所示的方法实施例的相应描述。

需说明的是，以上描述的任意装置实施例都仅仅是示意性的，其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。另外，本发明提供的网络设备或主机的实施例附图中，模块之间的连接关系表示它们之间具有通信连接，具体可以实现为一条或多条通信总线或信号线。本领域普通技术人员在不付出创造性劳动的情况下，即可以理解并实施。

本发明实施例还提供一种芯片系统，所述芯片系统包括至少一个处理器，存储器和接口电路，所述存储器、所述收发器和所述至少一个处理器通过线路互联，所述至少一个存储器中存储有计算机程序；所述计算机程序被所述处理器执行时，实现图3所示的方法流程。

本发明实施例还提供一种计算机可读存储介质，所述计算机可读存储介质中存储有计算机程序，当其在处理器上运行时，实现图3所示的方法流程。

本发明实施例还提供一种计算机程序产品，当所述计算机程序产品在处理器上运行时，实现图3所示的方法流程得以实现。

综上所述，本申请涉及行为识别模型和内容识别模型，行为识别模型为由多个数据流样本的报文特征和数据流类型预先训练得到，而内容识别模型是基于数据流的特征信息和由行为模型识别出的数据流类型训练得到，因此内容识别模型就是一个在线学习模型；行为识别模型能够识别出一些基本（或说典型）数据流的数据流类型，同样类型的数据流的某些特征（例如报文长度、报文传输速度等）可能在后续传输过程中发生变化，因此结合内容识别模型的在线学习特性从其他方面（如目的地址、协议类型等）对该数据流的类型进行识别，能够对冲行为识别模型在数据流传输过程中对数据流的数据流类型的识别误差，因此本申请采用行为识别模型和内容识别模型相结合的识别方式，提高了数据流的数据流类型的识别准确度，也提升了识别的泛化性，能够适用于应用云化部署、应用传输加密、私有应用等多种场景。

本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，该流程可以由计算机程序来计算机程序相关的硬件完成，该计算机程序可存储于计算机可读取存储介质中，该计算机程序在执行时，可包括如上述各方法实施例的流程。而前述的存储介质包括

Netskope Exh. 1005
Netskope v. Fortinet IPR2026-00515
Page 00484

或随机存储记忆体 RAM、磁碟或者光盘等各种可存储计算机程序代码的介质。

本发明实施例中提到的第一设备、第一置信度、第一数据流类型、第一预设阈值第一信息以及第一记录中的“第一”只是用来做名字标识，并不代表顺序上的第一。该规则同样适用于“第二”、“第三”和“第四”等。然而，本发明实施例中提到的第一个标识中的“第一个”代表顺序上的第一。该规则同样适用于“第 N 个”。

以上所述的具体实施方式，对本发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施方式而已，并不用于限定本发明的保护范围，凡在本发明的技术方案的基础之上，所做的任何修改、替换、改进等，均应包括在本发明的保护范围之内。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00485

权利要求书

1、一种数据流类型识别方法，其特征在于，包括：

根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度，其中，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项；

根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度，其中，所述特征信息包括目的地址和协议类型，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据所述行为识别模型得到的；

根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

2、根据权利要求1所述的方法，其特征在于，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，包括：

根据对应于第一数据流类型的所述第一置信度和对应于所述第一数据流类型的所述第二置信度计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

3、根据权利要求1所述的方法，其特征在于，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，包括：

根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

4、根据权利要求2或3所述的方法，其特征在于，所述方法还包括：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则向设备发送所述当前数据流的特征信息和所述第一数据流类型的信息，所述第二预设阈值大于所述第一预设阈值；

接收所述设备发送的第一信息，所述第一信息是所述设备根据所述当前数据流的特征信息和所述第一数据流类型的标识信息得到的；

根据所述第一信息更新所述内容识别模型，以得到新的内容识别模型。

5、根据权利要求2或3所述的方法，其特征在于，还包括：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，所述第二预设阈值大于所述第一预设阈值。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00486

6、根据权利要求5所述的方法，其特征在于，所述根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，包括：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述第一数据流类型不同，则将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

7、根据权利要求1-6任一项所述的方法，其特征在于，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型之后，还包括：

向运维支持系统OSS发送所述当前数据流的数据流类型的信息，所述当前数据流的数据流类型的信息用于所述OSS生成针对所述当前数据流的流量控制策略。

8、根据权利要求1-7任一项所述的方法，其特征在于，所述报文长度包括报文中以太帧长度、IP长度、传输协议长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议TCP和/或用户数据报协议UDP。

9、一种数据流类型识别方法，其特征在于，包括：

接收设备发送的当前数据流的特征信息和数据流类型的信息；

根据所述当前数据流的特征信息和数据流类型的信息生成第一信息；

向所述设备发送所述第一信息，以用于更新内容识别模型，所述内容识别模型用于得到对应于至少一个数据流类型的至少一个第二置信度，其中，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据行为识别模型得到的，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项。

10、根据权利要求9所述的方法，其特征在于，所述根据所述当前数据流的特征信息和数据流类型的信息生成第一信息，包括：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述当前数据流的数据流类型不同，则将所述第一记录的数据流类型更新为所述当前数据流的数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

11、一种数据流类型识别设备，其特征在于，所述设备是第一设备，所述第一设备包括存储器和处理器，其中，所述存储器用于存储计算机程序，所述处理器调用所述计算机程序，用于执行如下操作：

根据当前数据流的报文特征和行为识别模型得到所述当前数据流的对应于至少一个数据流类型的至少一个第一置信度，其中，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型；所述报文特征包括报文长度、报文传输速度、报文间隔时间和

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00487

文方向中的一项或者多项；

根据所述当前数据流的特征信息和内容识别模型得到所述当前数据流的对应于所述至少一个数据流类型的至少一个第二置信度，其中，所述特征信息包括目的地址和协议类型，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据所述行为识别模型得到的；

根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型。

12、根据权利要求11所述的设备，其特征在于，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，具体为：

根据对应于第一数据流类型的所述第一置信度和对应于所述第一数据流类型的所述第二置信度计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

13、根据权利要求11所述的设备，其特征在于，所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型，具体为：

根据对应于第一数据流类型的所述第一置信度、所述第一置信度的权重值、对应于所述第一数据流类型的所述第二置信度和所述第二置信度的权重值计算对应于所述第一数据流类型的综合置信度，所述第一数据流类型为所述至少一个数据流类型中的任意一个；

若对应于所述第一数据流类型的所述综合置信度大于第一预设阈值，则确定所述当前数据流的数据流类型为所述第一数据流类型。

14、根据权利要求12或13所述的设备，其特征在于，所述第一设备还包括收发器，所述处理器还用于：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则通过所述收发器向第二设备发送所述当前数据流的特征信息和所述第一数据流类型的信息，所述第二预设阈值大于所述第一预设阈值；

通过所述收发器接收所述第二设备发送的第一信息，所述第一信息是所述第二设备根据所述当前数据流的特征信息和所述第一数据流类型的标识信息得到的；

根据所述第一信息更新所述内容识别模型，以得到新的内容识别模型。

15、根据权利要求12或13所述的设备，其特征在于，所述处理器还用于：

若对应于所述第一数据流类型的所述综合置信度小于第二预设阈值，则根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，所述第二预设阈值大于所述第一预设阈值。

16、根据权利要求15所述的设备，其特征在于，所述根据所述当前数据流的特征信息和所述第一数据流类型信息更新所述内容识别模型，以得到新的内容识别模型，具体为：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00488

的数据流类型与所述第一数据流类型不同，则将所述第一记录的数据流类型更新为所述第一数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

17、根据权利要求 11-16 任一项所述的设备，其特征在于，所述第一设备还包括收发器，所述处理器还用于，在所述根据所述至少一个第一置信度和所述至少一个第二置信度确定所述当前数据流的数据流类型之后，通过所述收发器向运维支持系统 OSS 发送所述当前数据流的数据流类型的信息，所述当前数据流的数据流类型的所述信息用于所述 OSS 生成针对所述当前数据流的流量控制策略。

18、根据权利要求 11-17 任一项所述的设备，其特征在于，所述报文长度包括报文中以太帧长度、IP 长度、传输协议长度和报头长度中的一项或者多项，所述传输协议包括传输控制协议 TCP 和/或用户数据报协议 UDP。

19、一种数据流类型识别设备，所述设备是第一设备，其特征在于，所述第一设备包括存储器、处理器和收发器，其中，所述存储器用于存储计算机程序，所述处理器调用所述计算机程序，用于执行如下操作：

通过所述收发器接收第二设备发送的当前数据流的特征信息和数据流类型的信息；

根据所述当前数据流的特征信息和数据流类型的信息生成第一信息；

通过所述收发器向所述第二设备发送所述第一信息，以用于更新内容识别模型，所述内容识别模型用于得到对应于至少一个数据流类型的至少一个第二置信度，其中，所述内容识别模型为根据一条或多条历史数据流的特征信息和数据流类型得到的模型，所述历史数据流的数据流类型是根据行为识别模型得到的，所述行为识别模型为根据多个数据流样本的报文特征和数据流类型得到的模型，所述报文特征包括报文长度、报文传输速度、报文间隔时间和报文方向中的一项或者多项。

20、根据权利要求 19 所述的设备，其特征在于，所述根据所述当前数据流的特征信息和数据流类型的信息生成第一信息，具体为：

若多条记录中的第一记录的特征信息与所述当前数据流的特征信息相同但所述第一记录的数据流类型与所述当前数据流的数据流类型不同，则将所述第一记录的数据流类型更新为所述当前数据流的数据流类型，以获得第二记录；所述多条记录中的每一条记录包括特征信息和数据流类型；

对包括所述第二记录的多条记录进行训练，以得到新的内容识别模型。

21、一种数据流类型识别设备，其特征在于，包括用于执行权利要求 1-10 任一项所述的方法的单元。

22、一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中存储有计算机程序，当其在处理器上运行时，实现权利要求 1-10 任一所述的方法。

23、一种计算机程序产品，其特征在于，所述计算机程序产品存储在存储器上，当所述计算机程序产品在处理器上运行时，实现权利要求 1-10 任一项所述的方法。

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00489

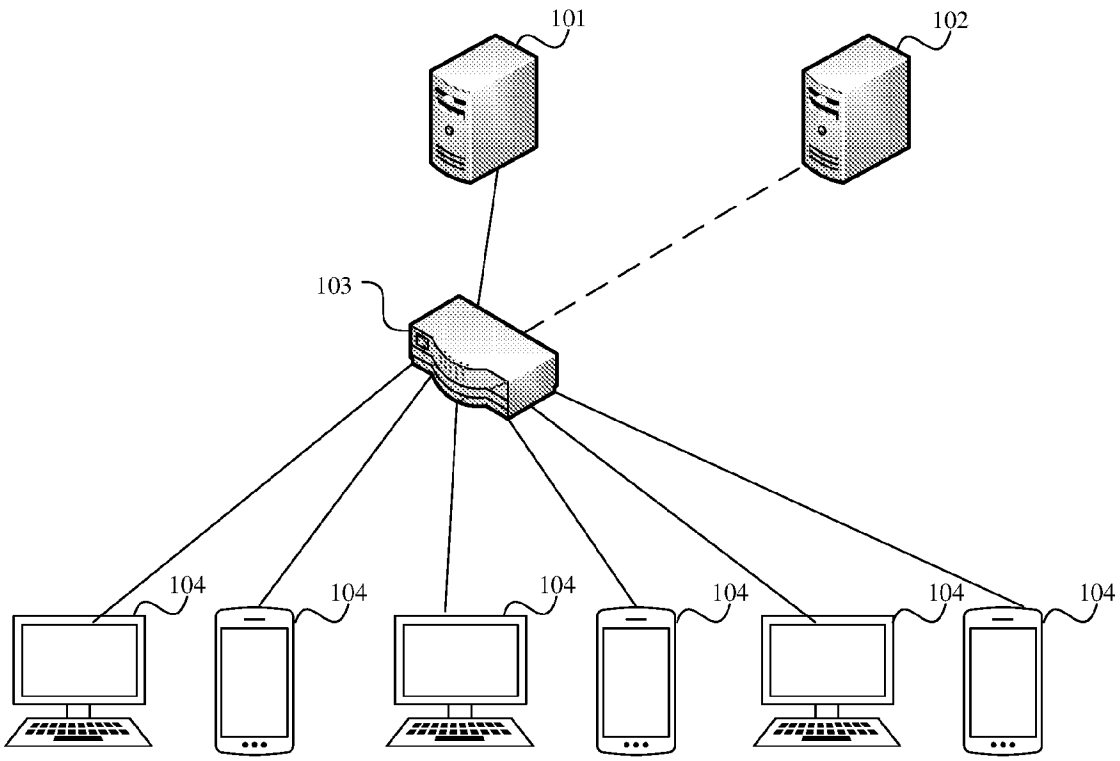


图 1

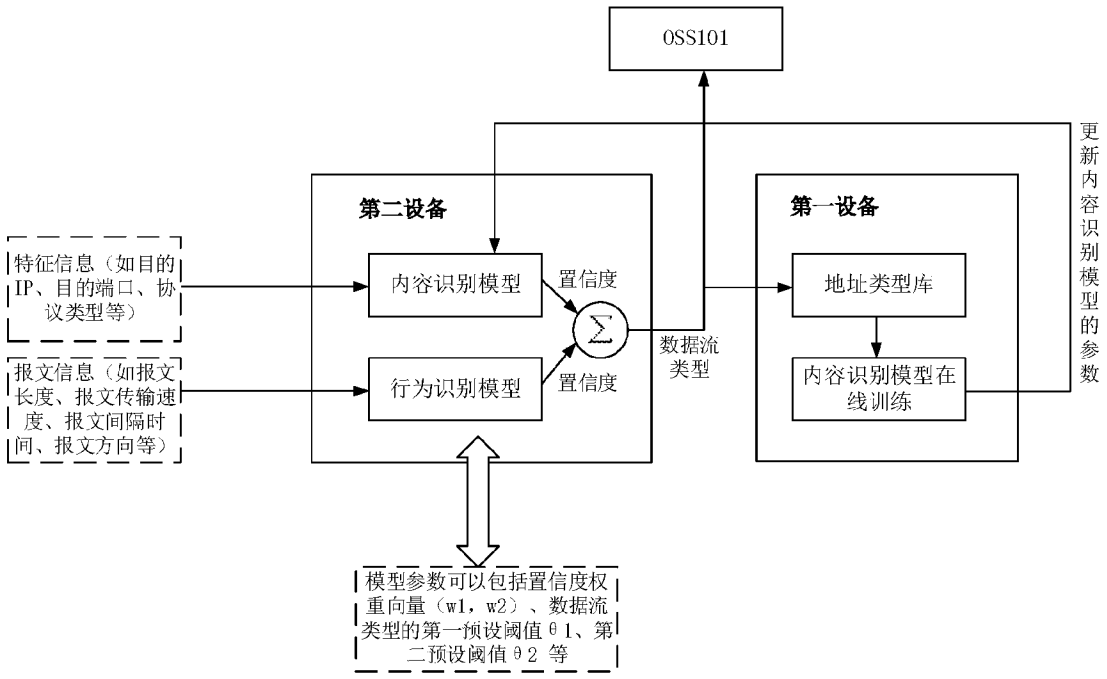


图 2A

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00490

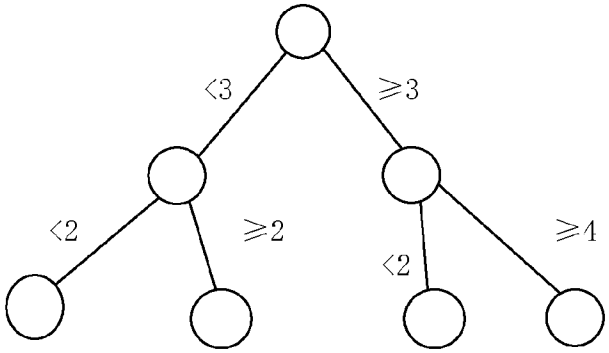


图 2B

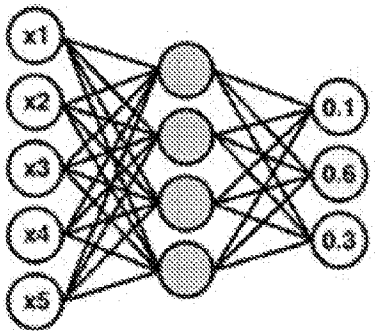


图 2C

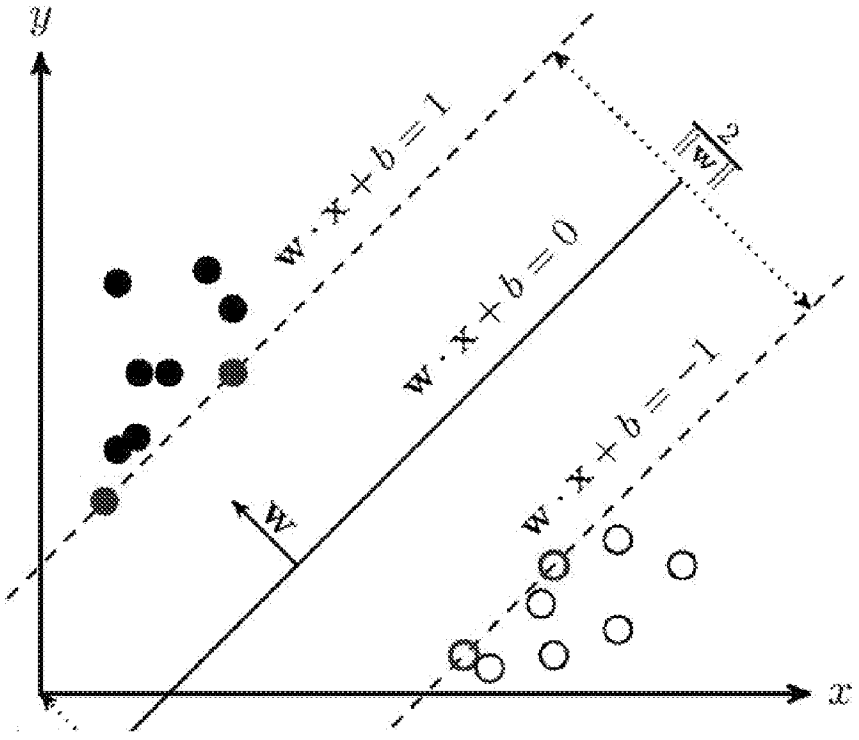


图 2D

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00491

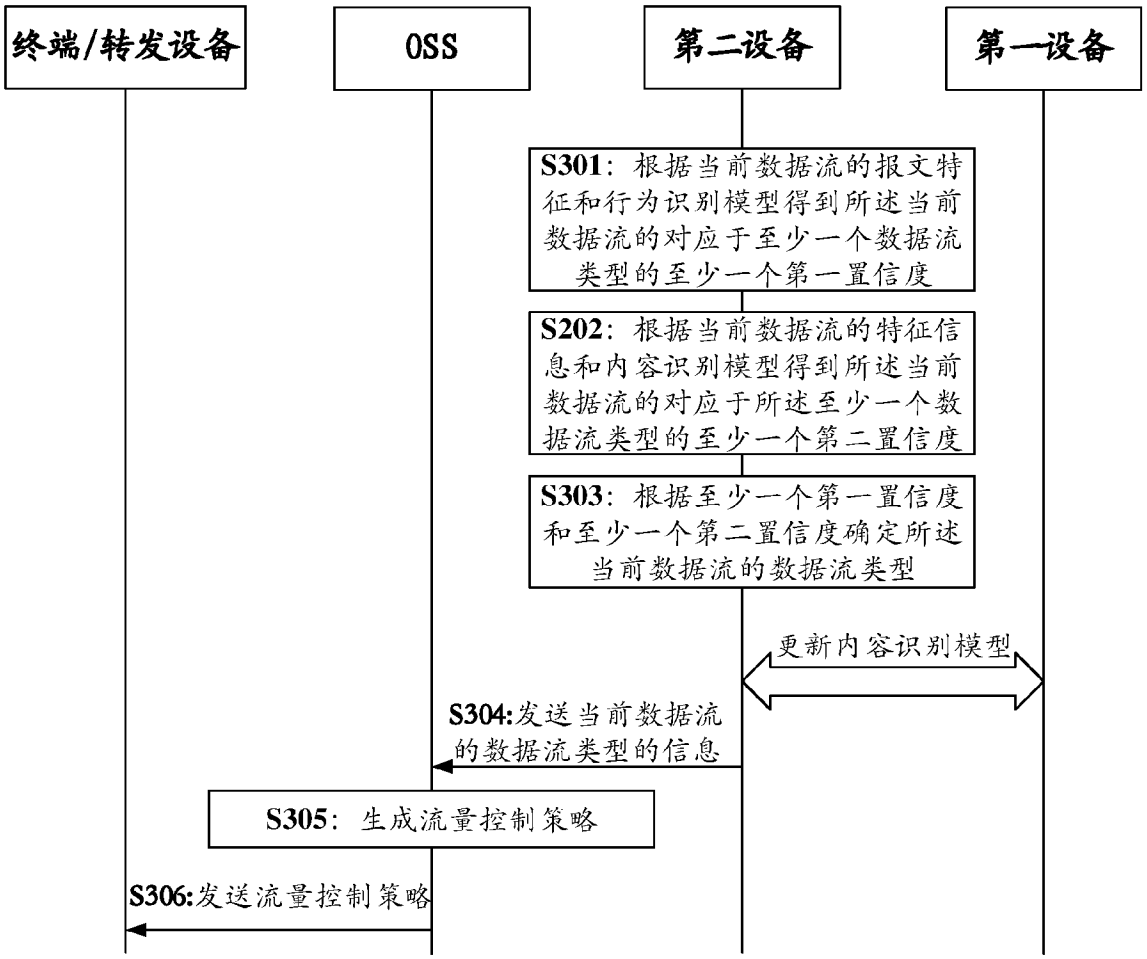


图 3

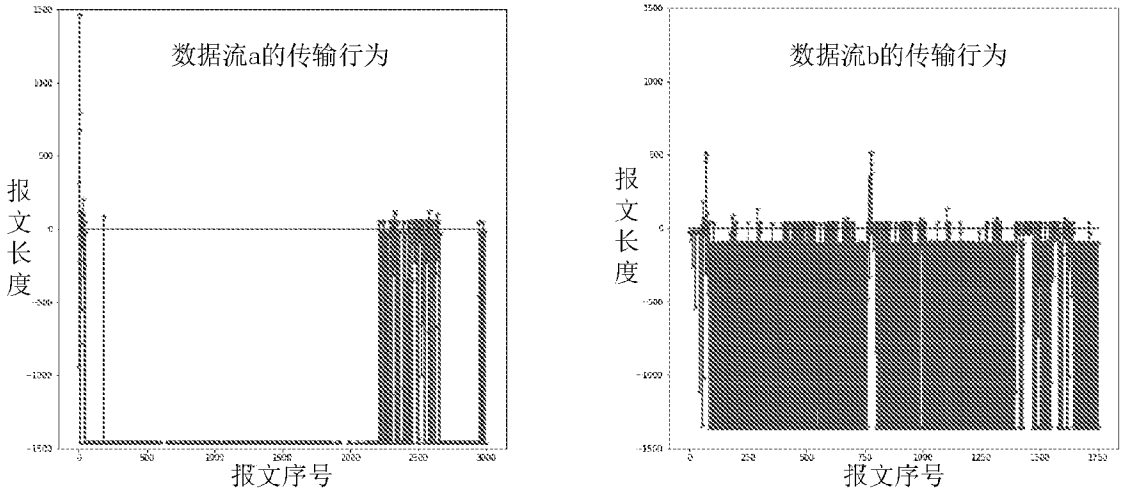


图 4

IP地址	端口	协议	数据流类型
10.129.56.39	443	TCP	桌面云

图 5

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00492

IP地址	端口	协议	数据流类型
10.129.56.39	443	TCP	视频会议

图 6

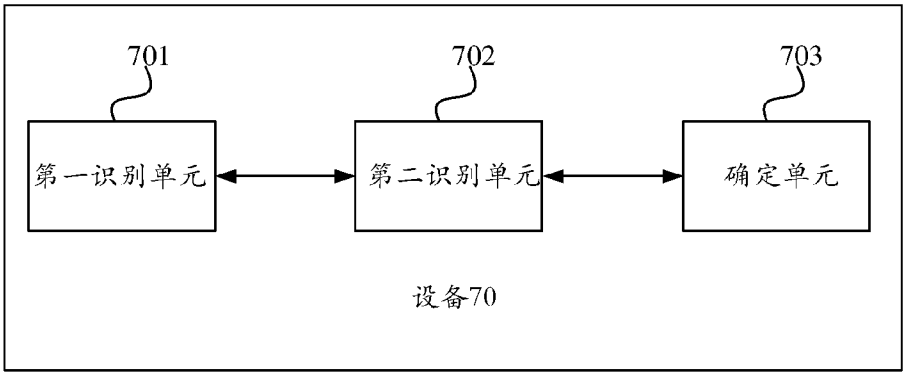


图 7

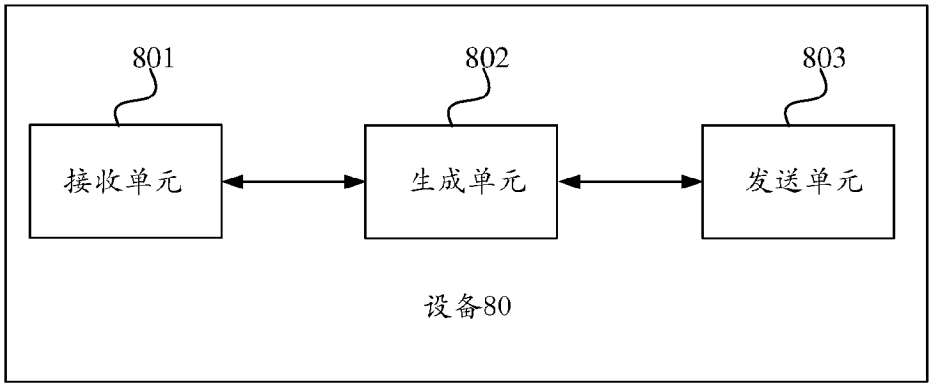


图 8

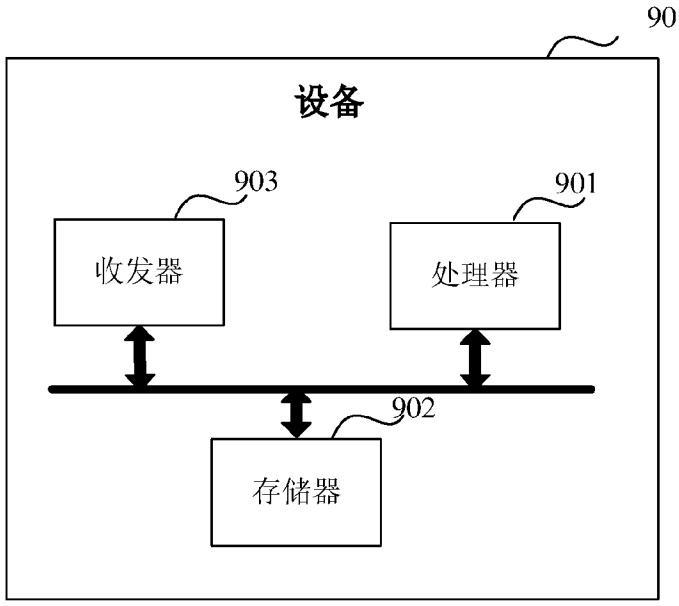


图 9

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00493

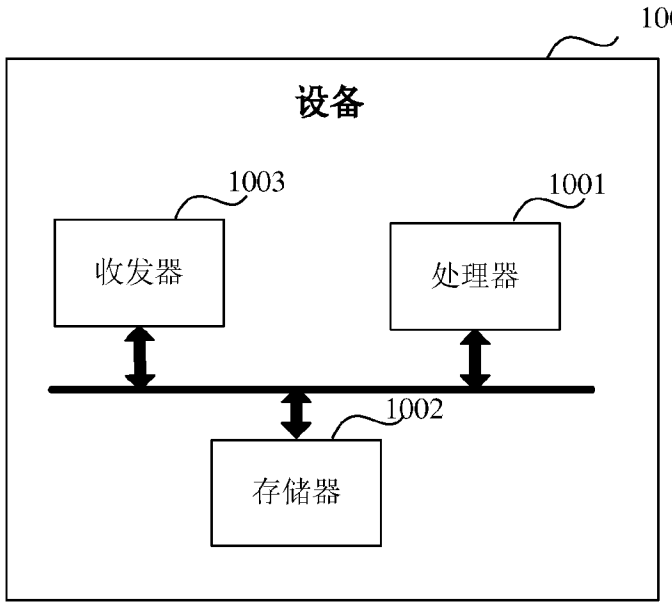


图 10

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00494

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/115693

A. CLASSIFICATION OF SUBJECT MATTER

H04L 12/851(2013.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC: 网络,数据,流,报文,类型,分类,识别,置信度,相似度,概率,特征,行为,内容,模型,历史 network, data, flow, message, type, classification, identify, confidence, probability, feature, behavior, content, model, history

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 107360032 A (CHINA SOUTHERN POWER GRID CO., LTD.) 17 November 2017 (2017-11-17) description, paragraphs 0066-0127	9-10, 19-23
Y	CN 108667747 A (SHENZHEN INSTITUTE OF INFORMATION TECHNOLOGY) 16 October 2018 (2018-10-16) description, paragraphs 0051-0103	9-10,19-23
A	CN 107360032 A (CHINA SOUTHERN POWER GRID CO., LTD.) 17 November 2017 (2017-11-17) description, paragraphs 0066-0127	1-8, 11-18
A	CN 108900432 A (SUN YAT-SEN UNIVERSITY) 27 November 2018 (2018-11-27) entire document	1-23
A	CN 110048962 A (GUANGDONG UNIVERSITY OF TECHNOLOGY) 23 July 2019 (2019-07-23) entire document	1-23
A	WO 2011114060 A2 (THALES) 22 September 2011 (2011-09-22) entire document	1-23

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
04 December 2020	**16 December 2020**
Name and mailing address of the ISA/CN **China National Intellectual Property Administration (ISA/CN)** **No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing 100088** **China**	Authorized officer
Facsimile No. **(86-10)62019451**	Telephone No.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00495

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	周丁丁 等 (ZHOU, Dingding et al.). "基于贝叶斯更新机制的流量分类算法 (Traffic Classification Based on Bayesian Updating Method)" *系统仿真学报 (Journal of System Simulation)*, Vol. 25, No. 11, 30 November 2013 (2013-11-30), entire document	1-23

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00496

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/115693

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	107360032	A	17 November 2017	None			
CN	108667747	A	16 October 2018	None			
CN	108900432	A	27 November 2018	None			
CN	110048962	A	23 July 2019	None			
WO	2011114060	A2	22 September 2011	US	2013117205	A1	09 May 2013
				SG	184120	A1	30 October 2012
				ES	2546129	T3	18 September 2015
				EP	2548337	A2	23 January 2013
				FR	2957738	B1	20 April 2012
				PT	2548337	E	16 October 2015
				MY	157412	A	15 June 2016
				FR	2957738	A1	23 September 2011

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00497

国际检索报告

国际申请号
PCT/CN2020/115693

A. 主题的分类

H04L 12/851(2013.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNPAT,CNKI,WPI,EPODOC:网络，数据，流，报文，类型，分类，识别，置信度，相似度，概率，特征，行为，内容，模型，历史，network, data, flow, message, type, classification, identify, confidence, probability, feature, behavior, content, model, history

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
Y	CN 107360032 A (中国南方电网有限责任公司) 2017年 11月 17日 (2017-11-17) 说明书第0066-0127段	9-10,19-23
Y	CN 108667747 A (深圳信息职业技术学院) 2018年 10月 16日 (2018-10-16) 说明书第0051-0103段	9-10,19-23
A	CN 107360032 A (中国南方电网有限责任公司) 2017年 11月 17日 (2017-11-17) 说明书第0066-0127段	1-8,11-18
A	CN 108900432 A (中山大学) 2018年 11月 27日 (2018-11-27) 全文	1-23
A	CN 110048962 A (广东工业大学) 2019年 7月 23日 (2019-07-23) 全文	1-23
A	WO 2011114060 A2 (THALES) 2011年 9月 22日 (2011-09-22) 全文	1 23
A	周丁丁等. "基于贝叶斯更新机制的流量分类算法" 系统仿真学报, 第25卷, 第11期, 2013年 11月 30日 (2013-11-30), 全文	1-23

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型:

"A" 认为不特别相关的表示了现有技术一般状态的文件

"E" 在国际申请日的当天或之后公布的在先申请或专利

"L" 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

"O" 涉及口头公开、使用、展览或其他方式公开的文件

"P" 公布日先于国际申请日但迟于所要求的优先权日的文件

"T" 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

"X" 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

"Y" 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

"&" 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2020年 12月 4日	2020年 12月 16日
ISA/CN的名称和邮寄地址 中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 靳晶 电话号码 86-(10)-53961790

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00498

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2020/115693

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	107360032	A	2017年 11月 17日	无			
CN	108667747	A	2018年 10月 16日	无			
CN	108900432	A	2018年 11月 27日	无			
CN	110048962	A	2019年 7月 23日	无			
WO	2011114060	A2	2011年 9月 22日	US	2013117205	A1	2013年 5月 9日
				SG	184120	A1	2012年 10月 30日
				ES	2546129	T3	2015年 9月 18日
				EP	2548337	A2	2013年 1月 23日
				FR	2957738	B1	2012年 4月 20日
				PT	2548337	E	2015年 10月 16日
				MY	157412	A	2016年 6月 15日
				FR	2957738	A1	2011年 9月 23日

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00499

Global Dossier

Office

US

Type

Application

17671199

0 0

Global Dossier

Global Dossier Initiative

The Global Dossier Initiative is a set of business services being developed by the IP5 Offices (http://www.fiveipoffices.org/index.html) (USPTO, EPO, JPO, KIPO, and SIPO) aimed at modernizing the global patent system and delivering benefits to all stakeholders through a single portal/user interface. Global Dossier will provide a single, secure point of access for the management of dossier and examination information, enabling and encouraging streamlining of office procedures among different IP Offices. This will lead to improved efficiency and predictability of global patent family prosecution with increased cost savings provided to patent applicants.

For more information, on the please visit the Global Dossier Initiative (http://www.uspto.gov/patents-getting-started/international-protection/global-dossier-initiative) page.

Global Dossier Public Access

The first service being provided by the Global Dossier Initiative is a secure, online access to the file histories of related applications from participating IP Offices, which currently include the IP5 Offices.

By using this service, users can see the patent family for a specific application, containing all related applications filed at participating IP Offices, along with the dossier, classification, and citation data for these applications. This service also provides Office Action Indicators to help users identify applications that contain office actions, a Collections View for saving documents and applications for review later on in the session, and the ability to download the documents in an application.

Citation List BETA

As part of the Global Dossier Initiative, the USPTO is pleased to announce the beta release of the Citation List. The Citation List is part of the USPTO's ongoing efforts to provide a more comprehensive listing of relevant citations that are available in related applications that share a common priority claim.

During the beta, the USPTO hopes to receive feedback from users on future enhancements and functionality to be included in subsequent releases of the Citation List. Users may submit their feedback to GlobalDossier@uspto.gov (mailto:GlobalDossier@uspto.gov?subject=Global%20Dossier%20Citation%20List%20Feedback) or through the Ideascale website (https://uspto-globaldossier.ideascale.com/a/index).

View: Citation List (/citations.html#)

(http://www.uspto.gov/)

BROWSE BY TOPIC

ABOUT THIS SITE

USPTO BACKGROUND

FEDERAL GOVERNMENT

cloud deploy* learn* similar* | Everything | ADVANCED SEA

0 selected PAGE 1 1-10 of 365 Results

1 ARTICLE

An evaluation of multi-probe locality sensitive hashing for computing similarities over web-scale query logs

Cormode, Graham ; Dasgupta, Anirban ; Goyal, Amit ; Lee, Chi Hoon; Wang, Yeng-Tseng

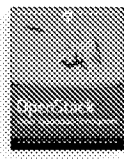
PloS one, 2018, Vol.13 (1), p.e0191175-e0191175

Many modern applications of AI such as web search, mobile browsing, image processing, and natural language processing rely on finding similar items from a large database of complex objects...

PEER REVIEWED OPEN ACCESS

Available Online

2 BOOK



Openstack cloud application development

Adkins, Scott, author.; Adkins, Scott, author.

2016

Available Online

3 ARTICLE

Development and clinical deployment of a smartphone-based visual field deep learning system for glaucoma detection

Li, Fei ; Song, Diping ; Chen, Han ; Xiong, Jian ; Li, Xingyi ; Zhong, Hua ; Tang, Guangxian ; Fan, Sujie ; Lam, Dennis S C ; Pan, Weihua ; Zheng, Yajuan ; Li, Ying ; Qu, Guoxiang ; He, Junjun ; Wang, Zhe ; Jin, Ling ; Zhou, Rouxi ; Song, Yunhe ; Sun, Yi ; Cheng, Weijing ; Yang, Chunman ; Fan, Yazhi ; Li, Yingjie ; Zhang, Hengli ; Yuan, Ye ; Xu, Yang ; Xiong, Yunfan ; Jin, Lingfei ; Lv, Aiguo ; Niu, Lingzhi ; Liu, Yuhong ;...

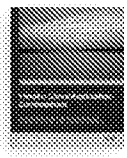
NPJ digital medicine, 2020, Vol.3 (1), p.123-123

...). Herein, we develop and evaluate the performance of 'iGlaucoma', a smartphone application-based deep learning system (DLS...

PEER REVIEWED OPEN ACCESS

Available Online

4 BOOK



Heroku cloud application development : a comprehensive guide to help you build, deploy, and troubleshoot cloud applications seamlessly using Heroku

Hanjura, Anubhav, author.; Shiriskar, Sagar, cover designer.

2014

Available Online

5 BOOK



Green Mobile Cloud Computing.

De, Debashis.; Mukherjee, Anwesha.; Buyya, Rajkumar.

2022

Available Online

6



BOOK

Docker in the Cloud

Goasguen, Sébastien, author.

2016

Available Online

7



VIDEO

Deploying Elastic Cloud Compute (EC2) Instances

Suehring, Steve, author.

2017

Available Online

8



ARTICLE

Semantic tracking and recommendation using fourfold similarity measure from large scale data using hadoop distributed framework in cloud

R, Priyadarshini ; Tamilselvan, Latha ; N, Rajendran

International journal of intelligent unmanned systems, 2019, Vol.7 (4), p.189-208

Purpose The purpose of this paper is to propose a fourfold semantic similarity that results in more accuracy compared to the existing literature...

PEER REVIEWED

Available Online

9



ARTICLE

RAFALE: Rethinking the provisioning of virtuAl network services using a Fast and scAlable machine LEarning approach

Suwi, Hanan ; Lahlou, Laaziz ; Kara, Nadjia ; Edstrom, Claes

The Journal of supercomputing, 2022, Vol.78 (14), p.15786-15819

.... To reap the benefits of this new technology, there is a need for novel mechanisms that help cloud and network/service providers deploy the increasingly complex virtual network services seamlessly...

PEER REVIEWED

Available Online

10



ARTICLE

A Graph Representation Learning Algorithm for Low-Order Proximity Feature Extraction to Enhance Unsupervised IDS Preprocessing

Hao, Yiran ; Sheng, Yiqiang ; Wang, Jinlin

Applied sciences, 2019, Vol.9 (20), p.4473

.... According to the homophily hypothesis, the local proximity structure in the similarity relational graph has similar embedding after preprocessing...

PEER REVIEWED OPEN ACCESS

Available Online

1 - 50

1. Cloud platform big data system based on open source technology

The invention discloses a kind of cloud platform big data system based on open source technology, its basic engineering platform mainly includes calculating, stores, the network equipment, and basic engineering platform is the hardware supported system of cloud platform; By the extension to...

FOSHAN ZUWO TECH CO LTD

CN107959728A | China Applications | 2018-04-24 | 4

2. Machine-learning model to predict probability of success of operator in paas cloud environment

Systems and methods are provided that integrate a machine-learning model, and more specifically, utilizing a platform as a service (PaaS) cloud to predict probability of success for an operator in an environment. An embodiment comprises a system having: a processor that executes computer executable...

INTL BUSINESS MACHINES CORP [+1]

WO2022105607A1 | WIPO Applications | 2022-05-27 | 4

3. **Machine-learning model to predict probability of success of an operator in a paas cloud enviornment**

Systems and methods are provided that integrate a machine-learning model, and more specifically, utilizing a platform as a service (PaaS) cloud to predict probability of success for an operator in an environment. An embodiment comprises a system having: a processor that executes computer executable...

INTL BUSINESS MACHINES CORP

US20220156631 | US Applications | 2022-05-19 | 4

4. **Acceleration element for cloud deployment**

High-level properties are extracted from a stream requesting virtual machine deployment to a cloud computing environment. A variety of prediction models are generated for a variety of different types of virtual machines. A regression-correlation map is applied to appropriate ones of the prediction...

INTL BUSINESS MACHINES CORP

DE112012004998T5 | Germany Applications | 2014-08-21 | 4

5. **Cognitive process enactment**

One embodiment provides for continuously adaptive business process management definition and execution including generating a continuously adaptive business process model and execution environment. New goals are discovered. Entity information is extracted from input documents. A model knowledge...

INTL BUSINESS MACHINES CORP

US10936998 | US Patents | 2021-03-02 | 4

6. **Learning from similar cloud deployments**

Result 1

Relevance

Publication Date

2011

First Cur Assignees by Relevanc

FOSHAN ZUWO TECH C...	(1)
INTL BUSINESS MACHIN...	(645)
LACEWORK INC	(6)
G SHIVNANI	(2)
ORACLE INTL CORP	(38)
HEWLETT PACKARD ENT...	(7)
BEIJING BAIDU NETCOM ...	(1)
MCAFEE LLC	(3)

Charts based on top 1500 results

More V

Search Results

cloud deploy* learn* similar*

Advanced Search

545,258 *Results for:* ***All: cloud deploy* learn* similar****

Edit Search | Save Search | RSS

Searched The ACM Full-Text Collection (686,483 records)
Expand your search to The ACM Guide to Computing Literature (3,449,908 records)

Feedback

RESULTS VIDEOS PERIODICALS SOFTWARE PEOPLE

Showing 1 - 20 of 545,258 Results

FILTERS | ***Sort by:*** *Relevance*

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

October 2022

PSDF: Privacy-aware IoV Service Deployment with Federated Learning in Cloud-Edge Computing

Xiaolong Xu, Wentao Liu, Yulan Zhang, Xuyun Zhang, Wanchun Dou, + 2

ACM Transactions on Intelligent Systems and Technology (TIST), Volume 13, Issue 5 • October 2022, Article No.: 70, pp 1–22 • https://doi.org/10.1145/3501810

Through the collaboration of cloud and edge, cloud-edge computing allows the edge that approximates end-users undertakes those non-computationally intensive service processing of the cloud, reducing the communication overhead and satisfying the low ...

Highlights

1 270

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00507

October 2022

Towards Edge-Cloud Collaborative Machine Learning: A Quality-aware Task Partition Framework

Zimu Zheng, Yunzhe Li, Han Song, Lanjun Wang, Fei Xia

CIKM '22: Proceedings of the 31st ACM International Conference on Information & Knowledge Management • October 2022, pp 3705–3714 • https://doi.org/10.1145/3511808.3557080

Edge-cloud collaborative tasks with real-world services emerge in recent years and attract worldwide attention. Unfortunately, state-of-the-art edge-cloud collaborative machine-learning services are still not that reliable due to the data heterogeneity ...

Highlights

0 105

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

OPEN ACCESS

February 2022

Deployment Archetypes for Cloud Applications

Anna Berenberg, Brad Calder

ACM Computing Surveys (CSUR), Volume 55, Issue 3 • April 2023, Article No.: 61, pp 1–48 • https://doi.org/10.1145/3498336

This is a survey article that explores six Cloud-based deployment archetypes for Cloud applications and the tradeoffs between them to achieve high availability, low end-user latency, and acceptable costs. These are (1) Zonal, (2) Regional, (3) Multi-...

Highlights

0 11,445

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00509

October 2018

Hybrid Cloud: An Inter Cloud Communication Mechanism

M. S. Raghunandan

CCIOT '18: Proceedings of the 2018 International Conference on Cloud Computing and Internet of Things • October 2018, pp 62–66 • https://doi.org/10.1145/3291064.3291071

Cloud computing has been in existence since the early 2000s and the growth in this domain of Computer Science has been by leaps and bounds. Cloud computing service providers offer 3 different services – Software, Platform and Infrastructure. Hybrid ...

Highlights

1 260

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

December 2019

The Journey of Cloud Computing with Open Source

Feilong Wang

UCC '19 Companion: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion • December 2019, pp 167 • https://doi.org/10.1145/3368235.3369378

Cloud computing is changing from a buzz word to a common technology nowadays. Though there is clarification for cloud computing for three types/layers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), ...

Highlights

1 398

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Upcoming Conferences

FPGA '23

February 12 - 14, 2023

Monterey Marriott, Monterey, CA, USA

FPGA '23 Website

HotMobile '23

February 22 - 23, 2023

Hyatt Regency Newport Beach, Newport Beach, CA, USA

HotMobile '23 Website

CG

Mo

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

December 2016

Performance-efficient service deployment and scheduling methods for composite cloud services

Kuo-Chan Huang, Yu-Chun Lu, Meng-Han Tsai, Ying-Ihih Wu, Hsi-Ya Chang

UCC '16: Proceedings of the 9th International Conference on Utility and Cloud Computing • December 2016, pp 240–244 • https://doi.org/10.1145/2996890.3007866

Cloud computing and services have greatly changed the way how people develop and use software, and also raised a lot of new research issues. In this paper, we investigate two such important issues, service deployment and service request scheduling, for ...

Highlights

8 86

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

PUBLIC ACCESS

December 2017

HPC Meets Cloud: Building Efficient Clouds for HPC, Big Data, and Deep Learning Middleware and Applications

Dhabaleswar K. Panda, Xiaoyi Lu

UCC '17: Proceedings of the10th International Conference on Utility and Cloud Computing • December 2017, pp 189–190 • https://doi.org/10.1145/3147213.3149455

Significant growth has been witnessed during the last few years in HPC clusters with multi-/many-core processors, accelerators, and high-performance interconnects (such as InfiniBand, Omni-Path, iWARP, and RoCE). To alleviate the cost burden, sharing ...

Highlights

2 374

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00514

May 2017

A Comparison of Reinforcement Learning Techniques for Fuzzy Cloud Auto-Scaling

Hamid Arabnejad, Claus Pahl, Pooyan Jamshidi, Giovani Estrada

CCGrid '17: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing • May 2017, pp 64–73 • https://doi.org/10.1109/CCGRID.2017.15

A goal of cloud service management is to design self-adaptable auto-scaler to react to workload fluctuations and changing the resources assigned. The key problem is how and when to add/remove resources in order to meet agreed service-level agreements. ...

Highlights

9 265

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

March 2017

Risk tolerance and trust issues in cloud-based E-learning

Siti Salmah Md Kassim, Mazleena Salleh, Anazida Zainal, Ab. Razak Che Husin

ICC '17: Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing • March 2017, Article No.: 73, pp 1–11 • https://doi.org/10.1145/3018896.3018973

Moving in tandem with internet services nowadays, education institutions should leverage cloud computing which is considered as ready-made platform. One of the key performance index (KPI) for educational institutions progress in teaching and learning ...

Highlights

2 152

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00516

May 2017

Optimized Cloud Deployment of Multi-tenant Software Considering Data Protection Concerns

Zoltán Ádám Mann, Andreas Metzger

CCGrid '17: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing • May 2017, pp 609–618 • https://doi.org/10.1109/CCGRID.2017.10

Concerns about protecting personal data and intellectual property are major obstacles to the adoption of cloud services. To ensure that a cloud tenant's data cannot be accessed by malicious code from another tenant, critical software components of ...

Highlights

1 88

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00517

May 2011

edubase Cloud: an open-source cloud platform for cloud engineers

Nobukazu Yoshioka, Shigetoshi Yokoyama, Yoshionori Tanabe, Shinichi Honiden

SECLOUD '11: Proceedings of the 2nd International Workshop on Software Engineering for Cloud Computing • May 2011, pp 73 • https://doi.org/10.1145/1985500.1985517

Education for cloud engineers is crucial in terms of innovation in the development of cloud technologies. We propose a new cloud platform based on open-source software that uses multi-clouds for the education.

A Highlights

5 389

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

December 2014

Cloud MF: Applying MDE to Tame the Complexity of Managing Multi-cloud Applications

Nicolas Ferry, Hui Song, Alessandro Rossini, Franck Chauvel, Arnor Solberg

UCC '14: Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing • December 2014, pp 269–277 • https://doi.org/10.1109/UCC.2014.36

The market of cloud computing encompasses an ever-growing number of cloud providers offering a multitude of infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS) solutions. The heterogeneity of these solutions hinders the proper ...

Highlights

12 116

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

August 2011

The architecture design of micro-learning platform based on cloud computing

Xiang Li, Jing Liu, Jun Han, Qian Zhang

ICCC '11: Proceedings of the 2011 International Conference on Innovative Computing and Cloud Computing • August 2011, pp 80–83 • https://doi.org/10.1145/2071639.2071659

Cloud computing is the synthesis of a variety of technology and innovation of business model. It is the revolution of the entire information industry operation mode, which have a profound impact on socio-economic structure of the future even the forms ...

Highlights

5 408

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00520

May 2016

Machine learning approach for cloud NoSQL databases performance modeling

Victor A. E. Farias, Flávio R. C. Sousa, José G. R. Maia, João P. P. Gomes, + 1

CCGRID '16: Proceedings of the 16th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing • May 2016, pp 617–620 • https://doi.org/10.1109/CCGrid.2016.83

Cloud computing is a successful, emerging paradigm that supports on-demand services with pay-as-you-go model. With the exponential growth of data, NoSQL databases have been used to manage data in the cloud. In these newly emerging settings, mechanisms ...

Highlights

1 49

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

January 2017

Cloud SLA relationships in multi-cloud environment: models and practices

Seokho Son, Hyun-Hwa Choi, Byeong Thaek Oh, Sun Wook Kim, + 1

ICCMS '17: Proceedings of the 8th International Conference on Computer Modeling and Simulation • January 2017, pp 1–6 • https://doi.org/10.1145/3036331.3050422

In the past few years, cloud computing has been realized and has achieved advancement. Sincemany cloud systems and providers have been deployed in this world, various models and platforms have been developed to support multiple cloud (i.e., multi-cloud) ...

Highlights

4 277

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00522

October 2011

To cloud or not to cloud?: musings on costs and viability

Yao Chen, Radu Sion

SOCC '11: Proceedings of the 2nd ACM Symposium on Cloud Computing • October 2011, Article No.: 29, pp 1–7 • https://doi.org/10.1145/2038916.2038945

In this paper we aim to understand the types of applications for which cloud computing is economically tenable, i.e., for which the cost savings associated with cloud placement outweigh any associated deployment costs.
We discover two scenarios. (i) In ...

Highlights

40 1,638

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

March 2017

Cloud security certifications: a comparison to improve cloud service provider security

Carlo Di Giulio, Read Sprabery, Charles Kamhoua, Kevin Kwiat, + 2

ICC '17: Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing • March 2017, Article No.: 120, pp 1–12 • https://doi.org/10.1145/3018896.3025169

The great diffusion of cloud computing applications and services in the last years has brought new threats to security of information. [1] IT Certification and authorization mechanisms try to provide assurance against those threats by leveraging high ...

Highlights

4 531

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00524

May 2015

Cloud-based machine learning tools for enhanced big data applications

Alfredo Cuzzocrea, Enzo Mumolo, Pietro Corona

CCGRID '15: Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing • May 2015, pp 908–914 • https://doi.org/10.1109/CCGrid.2015.170

We propose *Cloud-based machine learning tools for enhanced Big Data applications*, where the main idea is that of predicting the "*next*" *workload* occurring against the target Cloud infrastructure via an innovative *ensemble-based approach* that combine the ...

Highlights

0 34

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

October 2018

Energy-aware and Machine Learning-based Resource Provisioning of In-Memory Analytics on Cloud

Hosein Mohammadi Makrani, Hossein Sayadi, Devang Motwani, Han Wang, + 2

SoCC '18: Proceedings of the ACM Symposium on Cloud Computing • October 2018, pp 517 • https://doi.org/10.1145/3267809.3275459

In this work, we propose a proactive online resource provisioning methodology that addresses the challenge of resource provisioning for IMC workloads in heterogeneous cloud platforms consist of diverse types of servers. As cloud platforms provide a wide ...

Highlights

11 259

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00526

December 2016

Predicting cloud resource utilization

Michael Borkowski, Stefan Schulte, Christoph Hochreiner

UCC '16: Proceedings of the 9th International Conference on Utility and Cloud Computing • December 2016, pp 37–42 • https://doi.org/10.1145/2996890.2996907

A major challenge in Cloud computing is resource provisioning for computational tasks. Not surprisingly, previous work has established a number of solutions to provide Cloud resources in an efficient manner. However, in order to realize a holistic ...

Highlights

30 340

Categories

Journals
Magazines
Books
Proceedings

About

About ACM Digital Library
ACM Digital Library Board
Subscription Information
Author Guidelines

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Join ACM
Join SIGs
Subscribe to Publications
Institutions and Libraries

Contact
Facebook
Twitter
LinkedIn
Feedback
Bug Report

The ACM Digital Library is published by the Association for Computing Machinery. Copyright © 2023 ACM, Inc.

Terms of Usage | Privacy Policy | Code of Ethics

ACM DIGITAL LIBRARY
Association for Computing Machinery

About Cookies On This Site

We use cookies to ensure that we give you the best experience on our website.

Learn more

Got it!

Patents

cloud deploy* learn* similar*

cloud deploy* learn* similar*;

About 135,828 results

Download · Side-by-si

Sort by · Relevance · Group by · None · Deduplicate by · Family · Results / page · 10

Learning from similar cloud deployments



US · US20220247760A1 · Ofer Eilingsson · Lacework, Inc.
Priority 2017-11-27 · Filed 2022-02-14 · Published 2022-08-04
Learning from **similar cloud deployments**, including: identifying, for at least a portion of a first **cloud deployment**, one or more additional **cloud deployments** to utilize for cross-customer **learning**; receiving information describing configurations associated with the additional **cloud deployments**;

Detecting anomalies in a computer network based on usage similarity scores



US · US10911470B2 · Sudhakar Muddu · Splunk Inc.
Priority 2015-08-31 · Filed 2019-09-24 · Granted 2021-02-02 · Published 2021-02-02
claim 1 , wherein detecting the anomaly comprises: identifying a first group of the plurality of network devices with which the particular user interacts and that have usage **similarity** scores that satisfy a closeness criterion; identifying a second group of the plurality of network devices with ...

Methods and systems for the industrial internet of things



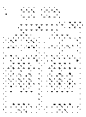
US · US20190174207A1 · Charles Howard Cella · StrongForce IoT Portfolio 2016, LLC
Priority 2016-05-09 · Filed 2018-11-09 · Published 2019-06-06
wherein the network control circuit bridges another **similarly** configured system from a first network to a second network by utilizing the plurality of network communication interfaces. 20 . The system of claim 19 , wherein the other **similarly** configured system has one or more operational ...

Method for data collection and frequency analysis with self-organization ...



US · US10983507B2 · Charles Howard Cella · Strong Force Iot Portfolio 2016, Llc
Priority 2016-05-09 · Filed 2020-02-27 · Granted 2021-04-20 · Published 2021-04-20
2150 . A data collection module 2160 can connect to (or be resident on) the machine 2020 . In one example, the data collection module 2160 can be located and accessible through a **cloud** network facility 2170 , can collect the waveform data 2010 from the machine 2020 , and deliver the waveform data 2010

Executing A Machine Learning Model In An Artificial Intelligence Infrastructure



US · US20220091823A1 · Brian Gold · Pure Storage, Inc.
Priority 2017-10-19 · Filed 2021-11-30 · Published 2022-03-24
What is claimed is: 1 . A method comprising: receiving, by a graphical processing unit ('GPU') server, a dataset transformed by a system that is external to the GPU server; and executing, by the GPU server, one or more machine **learning** algorithms using the transformed dataset as input. 2 . The ...

Machine Learning Models In An Artificial Intelligence Infrastructure



WO EP US CN · US20220358443A1 · Brian Gold · Pure Storage, Inc.
Priority 2017-10-19 · Filed 2022-04-26 · Published 2022-08-11
identifying differences between a machine **learning** model and a machine **learning** model previously executed by the artificial intelligence infrastructure; and storing, within the one or more storage systems, only a portion of the machine **learning** model that differs from the machine **learning** models ...

Method and system for intelligent cloud planning and decommissioning



EP US CN JP AU CA · US10268508B2 · Pramod Bijani · Accenture Global Solutions Limited
Priority 2015-07-17 · Filed 2016-07-17 · Granted 2019-04-09 · Published 2019-04-09
generating a recommendation report to recommend that the target application be **deployed** to a **cloud** computing system according to the **cloud deployment** information associated with the at least one application of the plurality of applications, wherein the recommendation includes a schematic ...

Systems and methods for data collection and signal conditioning in an ...



US · US11112785B2 · Charles Howard Cella · Strong Force Iot Portfolio 2016, Llc
Priority 2016-05-09 · Filed 2018-12-12 · Granted 2021-09-07 · Published 2021-09-07
23. The data collection and processing system of claim 17 , further comprising: a ~~cloud~~-based policy automation engine comprising at least one of: a sensor selection of the variable groups of sensor inputs; a sensor **deployment** of variable groups of sensor inputs; a sensor fusion of the variable ...

Hierarchical split of application between cloud and edge



WO EP US ES • EP3622397B1 • Samuel John George • Microsoft Technology Licensing, LLC
Priority 2017-05-09 • Filed 2018-04-11 • Granted 2022-11-02 • Published 2022-11-02
The apparatus of claim 2, the actions further including automatically **deploying** the further module to edge devices lacking the determined further module that become part of the determined position in at least one hierarchical category of the at least one hierarchical category of the plurality of ...

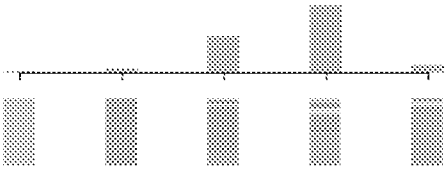
Intelligent backup and recovery of **cloud** computing environment



US • US11080252B2 • Jagaran Das • Accenture Global Solutions Limited
Priority 2018-07-02 • Filed 2018-08-31 • Granted 2021-08-31 • Published 2021-08-31
attaching the set of static backup rules to computing entities of the actively **deployed** target **cloud** computing environment; receiving a plurality of operating status of the actively **deployed** target **cloud** computing environment using a plurality of monitoring agents distributed within the target ...

Next › » About 135,828 results

Top 1000 results by filing date



Relative count of top 5 values

Assignees	Inventors	CPCs
Pure Storage, Inc.		18
International Business Machines Corporation		16
Oracle International Corporation		3
Intel Corporation		2
Cisco Technology, Inc.		2

Expand

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00530

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17671199
Filing Date	2022-02-14
First Named Inventor	ÚLFAR ERLINGSSON
Art Unit	OPAP
Examiner Name	DOCKET CENTRAL
Attorney Docket Number	L0041B1CIP2

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	10033611	B1	2018-07-24	Linkous et al.	
	2	10115111	B2	2018-10-30	Miltonberger	
	3	10127273	B2	2018-11-13	Dickey	
	4	10142357	B1	2018-11-27	Tamersoy et al.	
	5	10205736	B2	2019-02-12	Rieke et al.	
	6	10339309	B1	2019-07-02	Kling et al.	
	7	10389742	B2	2019-08-20	Reddy et al.	
	8	10581891	B1	2020-03-03	Kapoor et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	9	10594718	B1	2020-03-17	Deaguero et al.	
	10	10708082	B1	2020-07-07	Bakiaraj et al.	
	11	10735329	B2	2020-08-04	WANG et al.	
	12	11153339	B1	2021-10-19	Kapoor et al.	
	13	5584024	A	1996-12-10	Shwartz	
	14	5806062	A	1998-09-08	Chen et al.	
	15	8103906	B1	2012-01-24	Alibakhsh et al.	
	16	8122122	B1	2012-02-21	Clingenpeel et al.	
	17	8443442	B2	2013-05-14	Wang et al.	
	18	8671453	B2	2014-03-11	Underwood et al.	
	19	8862524	B2	2014-10-14	Zheng et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	20	9021583	B2	2015-04-28	Wittenstein et al.	
	21	9037273	B2	2015-05-19	Mikkelsen	
	22	9332020	B2	2016-05-03	Thomas et al.	
	23	9515999	B2	2016-12-06	Ylonen	
	24	9516053	B1	2016-12-06	Muddu et al.	
	25	9558265	B1	2017-01-31	Tacchi et al.	
	26	9596253	B2	2017-03-14	Chauhan et al.	
	27	9654503	B1	2017-05-16	Kowalyshyn	
	28	9734040	B2	2017-08-15	Gounares	
	29	9749339	B2	2017-08-29	Kadambe et al.	
	30	9853968	B2	2017-12-26	Shen et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	31	9953014	B1	2018-04-24	Reshadi et al.	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20020184225	A1	2002-12-05	Ghukasyan	
	2	20030037136	A1	2003-02-20	Labovitz et al.	
	3	20030179227	A1	2003-09-25	Ahmad et al.	
	4	20040015470	A1	2004-01-22	Smith et al.	
	5	20050102284	A1	2005-05-12	Srinivasan et al.	
	6	20050188222	A1	2005-08-25	Motsinger et al.	
	7	20060085437	A1	2006-04-20	Brodhun et al.	
	8	20060259470	A1	2006-11-16	Chandrasekharan et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	9	20060288415	A1	2006-12-21	Wong	
	10	20070050497	A1	2007-03-01	Haley et al.	
	11	20070118909	A1	2007-05-24	Hertzog et al.	
	12	20080148180	A1	2008-06-19	Liu et al.	
	13	20080151893	A1	2008-06-26	Nordmark et al.	
	14	20090019160	A1	2009-01-15	Schuler	
	15	20090228474	A1	2009-09-10	Chiu et al.	
	16	20090271504	A1	2009-10-29	Ginter et al.	
	17	20100094767	A1	2010-04-15	Miltonberger	
	18	20100114931	A1	2010-05-06	Xie et al.	
	19	20100172261	A1	2010-07-08	Shinbo et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	20	20100274785	A1	2010-10-28	Procopiuc et al.	
	21	20110055138	A1	2011-03-03	Khanduja et al.	
	22	20120005243	A1	2012-01-05	Merwe et al.	
	23	20120317151	A1	2012-12-13	Ruf et al.	
	24	20120323956	A1	2012-12-20	Dumitru et al.	
	25	20130205357	A1	2013-08-08	Bahnck et al.	
	26	20130304915	A1	2013-11-14	KAWAI	
	27	20140115001	A1	2014-04-24	Arroyo et al.	
	28	20140115011	A1	2014-04-24	Buerner et al.	
	29	20140208191	A1	2014-07-24	Zaric et al.	
	30	20140245443	A1	2014-08-28	Chakraborty	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	31	20140280068	A1	2014-09-18	DHOOPAR et al.	
	32	20140359558	A1	2014-12-04	Chamberlain	
	33	20150135312	A1	2015-05-14	Wada et al.	
	34	20150188751	A1	2015-07-02	Vasseur et al.	
	35	20150341379	A1	2015-11-26	Lefebvre et al.	
	36	20150356144	A1	2015-12-10	Chawla et al.	
	37	20160078365	A1	2016-03-17	Baumard	
	38	20160080404	A1	2016-03-17	KOHOUT et al.	
	39	20160110434	A1	2016-04-21	Kakaraddi et al.	
	40	20160149937	A1	2016-05-26	KATMOR et al.	
	41	20160205125	A1	2016-07-14	Kim et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	42	20160218911	A1	2016-07-28	WESSELS et al.	
	43	20160359592	A1	2016-12-08	Kulshreshtha et al.	
	44	20160359872	A1	2016-12-08	Yadav et al.	
	45	20170118240	A1	2017-04-27	Reddy et al.	
	46	20170134240	A1	2017-05-11	Hévizi et al.	
	47	20170147646	A1	2017-05-25	LEE et al.	
	48	20170163666	A1	2017-06-08	Venkatramani et al.	
	49	20170251013	A1	2017-08-31	Kirti et al.	
	50	20170272344	A1	2017-09-21	Tang et al.	
	51	20170279827	A1	2017-09-28	Savalle et al.	
	52	20170288974	A1	2017-10-05	Yoshihira et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	53	20170359361	A1	2017-12-14	Modani et al.	
	54	20170366492	A1	2017-12-21	Ho et al.	
	55	20180020015	A1	2018-01-18	Munro et al.	
	56	20180034840	A1	2018-02-01	Marquardt et al.	
	57	20180063178	A1	2018-03-01	Jadhav et al.	
	58	20180103052	A1	2018-04-12	Choudhury et al.	
	59	20180115578	A1	2018-04-26	Subbarayan et al.	
	60	20180137858	A1	2018-05-17	Saxena et al.	
	61	20180173789	A1	2018-06-21	Llagostera et al.	
	62	20180174062	A1	2018-06-21	Simo et al.	
	63	20180181750	A1	2018-06-28	Lamothe-Brassard	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	64	20180248901	A1	2018-08-30	Rieke	
	65	20180268078	A1	2018-09-20	Gianetto et al.	
	66	20180288063	A1	2018-10-04	Koottayi et al.	
	67	20180329958	A1	2018-11-15	Choudhury et al.	
	68	20180336353	A1	2018-11-22	Manadhata et al.	
	69	20180357422	A1	2018-12-13	Telang et al.	
	70	20180367548	A1	2018-12-20	Stokes III et al.	
	71	20190028327	A1	2019-01-24	SILVA et al.	
	72	20190109870	A1	2019-04-11	BEDHAPUDI et al.	
	73	20190132224	A1	2019-05-02	VERMA et al.	
	74	20190222597	A1	2019-07-18	Crabtree et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	75	20190259033	A1	2019-08-22	Reddy et al.	
	76	20190339965	A1	2019-11-07	Garvey et al.	
	77	20190349305	A1	2019-11-14	WANG et al.	
	78	20190378050	A1	2019-12-12	Edkin et al.	
	79	20200076685	A1	2020-03-05	Vaidya et al.	
	80	20200192690	A1	2020-06-18	Gupta et al.	
	81	20200252376	A1	2020-08-06	Feng et al.	
	82	20200272740	A1	2020-08-27	Obee et al.	
	83	20200278892	A1	2020-09-03	Nainar et al.	
	84	20200280592	A1	2020-09-03	Ithal et al.	
	85	20200311644	A1	2020-10-01	WILLARD III et al.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	86	20200403860	A1	2020-12-24	Lewis et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	AI-YASEEN, et al., "Real-Time Intrusion Detection System Using Multi-Agent System", IAENG International Journal of Computer Science, volume 43, no. 1, February 2016, pp. 80-90, International Association of Engineers (IAENG), Hong Kong.	
	2	AKOGLU, et al., "Graph-based Anomaly Detection and Description: A Survey", April 28, 2014.	
	3	AMMAR, et al., "Query Optimization Techniques in Graph Databases", International Journal of Database Management Systems (IIDMS), Vol. 8, No. 4, August 2016, pages 1-14 (Year: 2016).	
	4	BALASUBRAMANIYAN, et al., "An Architecture For Intrusion Detection Using Autonomous Agents", In Proceedings 14th Annual Computer Security Applications Conference (Cat. No. 98EX217), 19 pages, June 1998, IEEE, DOI: 10.1109/CSAC.1998.738563.	
	5	BEUTEL, et al., "User Behavior Modeling with Large-Scale Graph Analysis, Computer Science Department, Carnegie Mellon University, May 2016.	

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	6	BUGIEL, et al., "Towards Taming Privilege-Escalation Attacks on Android", In NOSS (Vol. 17, p. 19), 2012, February.
	7	CHANG, et al., "Reality Bites - Progressive Querying and Result Visualization in Logical and VR Spaces", Proceedings of 1994 IEEE Symposium on Visual Languages, pp. 100-109, October 1994, IEEE, DOI: 10.1109NL.1994.363635,
	8	CHESSON, et al., "Communication And Control In A Cluster Network", ACM '74: Proceedings of the 1974 annual ACM conference - Volume 2, January 1974, pp 509-514, http://doi.org/10.1145/1408839 (Year 1974).
	9	CROSBIE et al., "Defending a Computer System using Autonomous Agents", docs.lib.purdue.edu (online), March 1995, 11 pages.
	10	HAUTAMAKI, et al., "Outlier Detection Using k-Nearest Neighbour Graph", Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), Vol. 3, August 2004, IEEE, DOI: 10.1109/ICPR.2004.1334558.
	11	HOOPER, et al., "Medusa: a simple tool for interaction graph analysis", Bioinformatics, Vol. 21 no. 24, September 2005, pages 4432-4433, Oxford University Press (online), URL: https://academic.oup.com/bioinformatics/article/21/24/4432/179694.
	12	KOUTRA, et al., "Exploring and Making Sense of Large Graphs", Computer Science Department, Carnegie Mellon University, August 2015.
	13	LEOPOLD, et al., "A Visual Query System for the Specification and Scientific Analysis off Continual Queries", Proceedings, IEEE Symposia on Human-Centric Computing Languages and Environments (Cat. No.01TH8587), September 2001, pp. 203-211, IEEE, doi: 10.1109/HCC.2001.995260.
	14	LIAO, et al., "Visualizing Graph Dynamics And Similarity For Enterprise Network Security And Management", VizSec '10: Proceedings of the Seventh International Symposium on Visualization for Cyber Security, September 2010, pp. 34-45, URL: https://doi.org/10.1145/1850795.1850799.
	15	MATEESCU, et al., "Join-Graph Propagation Algorithms", Journal of Artificial Intelligence Research, vol. 37, March 2010, pages 279-328, AI Access Foundation, Inc. (online), URL: https://doi.org/10.1613/jair.2842.
	16	MORIANO, et al., "Insider Threat Event Detection in User-System Interactions", MIST '17: Proceedings of the 2017 International Workshop on Managing Insider Security Threats, October 2017, pp 1-12, ACM Digital Library (online), URL: https://doi.Jrg/10.1145/3139923.3139928.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

	17	RANSHOUS, et al., "Anomaly detection in dynamic networks: a survey", WIREs Computational Statistics, volume 7, May/June 2015, pp. 223-247, Wiley Periodicals, Inc, United States.
	18	TAMASSIA, et al., "Graph Drawing for Security Visualization", In: Graph Drawing (GD 2008), Lecture Notes in Computer Science, vol 5417, Springer, Berlin, Heidelberg (online), URL: https://doi.org/10.1007/978-3-642-00219-9_2.
	19	VAAS, et al., "Detecting disguised processes using Application-Behavior Profiling", In 2017 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-6, June 2017, IEEE, DOI: 10.1109/THS.2017.7943508.
	20	YU, et al., "Recommending Join Queries Based on Path Frequency", 2015 12th Web Information System and Application Conference (WISA), September 2015, pp. 21-26, IEEE, DOI: 10.1109/WISA.2015.52.

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	17671199
	Filing Date	2022-02-14
	First Named Inventor	ÚLFAR ERLINGSSON
	Art Unit	OPAP
	Examiner Name	DOCKET CENTRAL
	Attorney Docket Number	L0041B1CIP2

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Edward Lenart/	Date (YYYY-MM-DD)	2022-10-03
Name/Print	Edward Lenart	Registration Number	63665

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/671,199	**10/03/2022 04:51:48 PM ET**	**L0041B1CIP2**

Title of Invention

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1070	FILED BY	Kate Hill
PATENT CENTER #	61028616	FILING DATE	02/14/2022
CUSTOMER #	182115	FIRST NAMED INVENTOR	ÚLFAR ERLINGSSON
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Edward Lenart

Documents

TOTAL DOCUMENTS: 21

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
2022-10-03_L0041B1CIP2_IDS.pdf	16	Information Disclosure Statement (IDS) Form (SB08)	1257 KB
NPL_Al-Yaseen_Real-timeIntrusion.pdf	11	Non Patent Literature	449 KB
NPL_Akoglu_GraphBasedAnomalyDetection.pdf	68	Non Patent Literature	806 KB
NPL_Ammar_QueryOptimizationTechniques.pdf	14	Non Patent Literature	311 KB
NPL_Balasubramaniyan_AnArchitectureForIntrusion.pdf	19	Non Patent Literature	224 KB

NPL_Beutel_UserBehaviorModeling.pdf	275	Non Patent Literature	5352 KB
NPL_Bugiel_TowardsTamingPrivilege-Escalation.pdf	18	Non Patent Literature	5950 KB
NPL_Chang_RealityBitesProgressiveQuerying.pdf	10	Non Patent Literature	742 KB
NPL_Chesson_CommunicationAndControl.pdf	6	Non Patent Literature	702 KB
NPL_Crosbie_DefendingAComputerSystem.pdf	11	Non Patent Literature	289 KB
NPL_Hautamaki_OutlierDetectionUsingK-Nearest.pdf	4	Non Patent Literature	338 KB
NPL_Hooper_Medusa-ASimpleTool.pdf	2	Non Patent Literature	214 KB
NPL_Koutra_ExploringAndMakingSense.pdf	232	Non Patent Literature	3825 KB
NPL_Leopold_AVisualQuerySystem.pdf	9	Non Patent Literature	396 KB
NPL_Laio_VisualizingGraphDynamics.pdf	12	Non Patent Literature	4704 KB
NPL_Mateescu_Join-GraphPropogation.pdf	50	Non Patent Literature	701 KB
NPL_Moriano_InsiderThreatEvent.pdf	12	Non Patent Literature	2137 KB
NPL_Ranshous_AnomalyDetectionInDynamicNetworks.pdf	25	Non Patent Literature	493 KB
NPL_Tamassia_GraphDrawin	12	Non Patent Literature	2074 KB

gForSecurity.pdf			
NPL_Vaas_DetectingDisguisedProcesses.pdf	6	Non Patent Literature	1083 KB
NPL_Yu_RecommendingJoinQueries.pdf	6	Non Patent Literature	611 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
2022-10-03_L0041B1CIP2_IDS.pdf	9E84FE13A6348506F833877586619E3417AB35F7D60CD7AC03C4F5829EF8ACC722AE4DC3222003B56243E53E059F59338796FB899E61E142C81B9D9112C100D9
NPL_Al-Yaseen_Real-timeIntrusion.pdf	410D4DC41C77675E56DA21A96B493CF00C08E22AAE966408AE09E7FFFF61E430D32F54CEE2A0ABB6D9190159EFD3B8DF0BC6AD0A7E2F05333B07625264CA9B41
NPL_Akoglu_GraphBasedAnomalyDetection.pdf	B29A5D3C6B5D2B61818CB3610CDFEA65FE981A62778C3C145BA31E27B69F0EDE7FA6CDDFFF259BF50B3655BFF1BD581567422EB6CB933ACBBE7BE1AAF7FD8838
NPL_Ammar_QueryOptimizationTechniques.pdf	5450BE9D00EE222F51311A0E76BDDA29E49D0D42E050CBAE2AF6F481B5E00200026AC4F585E30704A41547DE02453168F935F48B22D259DA9E0E4311B4869C20
NPL_Balasubramaniyan_AnArchitectureForIntrusion.pdf	89D89D9869DCBBE55A4153A48E1BD4CED5FBF83E3D42AD8A7E6B48D2EF1F5988072BCDF32F79A066A708CE2E85D2EC0FF4036FD1EDF166D17C684E4027B7A93F
NPL_Beutel_UserBehaviorModeling.pdf	F5DF9EE6942BFC7F40B394F4A813718E7C3FC191A3195EC077E7F91310D11ACB4DAB3D7D7BF9764D765C3331DF1669FA9FD2BD8F39B43A1ED3C0D1A00D3C2561
NPL_Bugiel_TowardsTamingPrivilege-Escalation.pdf	AB4A380C4B69D6746D2E55C35C8472398C45D815B9B14BA12ED45665EE24F28D6766733FFB6F9D371D9511FC8EB36DCC85

	6F64F0C069B4818320B119E2488BB7
NPL_Chang_RealityBitesProgressiveQuerying.pdf	36FFFB3A06A3DD8F0977DF78B963E835CDBBFAA3F4C431BF41C6061D0A88FBC603019431A55DFF73C2E886FF223F487F0CFCBE6CFDF1E6AE13F907E045869F15
NPL_Chesson_CommunicationAndControl.pdf	72FF001DF810C923A0A811B06C6EB03B8DCB29812B2E5FC76F8AC800444F1E1AED684B9055EDCDFD41263868D809822463E745D5D187C4B9A49DB3380167958C
NPL_Crosbie_DefendingAComputerSystem.pdf	D65F6310889E53834E679AF8C8296FDC35FDEF9839D7158ABE5F2A6BD9498927D6B700DE6CEA8D6C6B4677FD19DA3D5AAC1E8B976A4BD92A3E4BDDC62348BE7B
NPL_Hautamaki_OutlierDetectionUsingK-Nearest.pdf	8037EC36114C79375D3E058FD9739C2F3E8D02D00E4FA1A5C8902CC7711AFAE3651FE91633DB6A833E448F38163E626348404CADE416B2CE6362339C0CC6A057
NPL_Hooper_Medusa-ASimpleTool.pdf	BADCC82C81EB5A717282FE221B2DA79DDE5F844E1D77947C1B10A0172D6E1100A6FB314AF8BA7FA22B3A5A7BA555666C5C5E5399AE00399E9940DC3DE54E12A0
NPL_Koutra_ExploringAndMakingSense.pdf	234A544FE73F84AE279FC4F0975B640DD129EBAA9AA8A2C6DDCA07BD85B273404CD7CB34F0F0C5C39400ABC08DE842231FCD82267321DE58B9615011EF49C8C7
NPL_Leopold_AVisualQuerySystem.pdf	2DBBCB42678A322C54D58BBD9BF7D8B950B37B081928F4644496E5B65635F403C142238EE62544792958 2DA9A964465FC4CE86113C586524BB9D20F7B25B73ED
NPL_Laio_VisualizingGraphDynamics.pdf	655425F1461CA49285FC1541D4BC3C8213580A7412C032D76A7957A4F11F82BB8964C43BA5E0584A919AF7D9691F4CEAE02E75B735FB14DE75B16DE0B520832A
NPL_Mateescu_Join-GraphPropogation.pdf	240C1B56027F98F776F7616A57CA1C63F6B2B6FA985197300E22D2DFA92FB757D28DC00DA5E4EBFE2EA583872A966C02D95F06D45F08CC280D635635D5697732
NPL_Moriano_InsiderThreatEvent.pdf	883CDB3BE49DD3012BEB1F829023F06BB59D15BECEF2AB53AF27993E86FCA7363D8BF0600D1FFA9461393D0E3F69C7BCB

	A05F04BCC224A4DF995F22DB209A447
NPL_Ranshous_AnomalyDetectionInDynamicNetworks.pdf	25E465CAC9E23F8E0F5C46C006FC398BBE2B21BE567F31704F18E9A79ADCCD956A6199D0C6D90B791E03075E18E95D90453E8B8FFFC84E6D495B0C48B3F692AB
NPL_Tamassia_GraphDrawingForSecurity.pdf	50D27B55A1265D5471EA36A1A05254AB81F273FC80FDC0A0512A0958AB42CCDBF51B98BDE00A5938251DE58AAA00774A58494D97CFDE610EE76842E137049327
NPL_Vaas_DetectingDisguisedProcesses.pdf	A4CC045FC2C3A718E6D815CCEBC2F20E3B11D69453239F74EE37BC3FC19DF125DB1BB1431229BDAD797EC3B012247A6AE36D19EEE44FACA85710AA90F1F80C07
NPL_Yu_RecommendingJoinQueries.pdf	453387A31544F555B48A673B5B4BF42BB7A8247AC21511F5140B57B14F4D4E6F957DF84297CDFAA72436CC2B907AFC933F42BBBE4116C0858F35B25FE61D14C5

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
17/671,199	02/14/2022	ULFAR ERLINGSSON	L0041B1CIP2

CONFIRMATION NO. 1070

182115
Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

PUBLICATION NOTICE

OC000000135622222

Title:LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Publication No.US-2022-0247769-A1
Publication Date:08/04/2022

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Public Records Division. The Public Records Division can be reached by telephone at (571) 272-3150 or (800) 972-6382, by facsimile at (571) 273-3250, by mail addressed to the United States Patent and Trademark Office, Public Records Division, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently https://portal.uspto.gov/pair/PublicPair. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00552

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 17/671,199

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	320
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	700
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	800
TOTAL CLAIMS (37 CFR 1.16(i))	20 minus 20 =	*			OR	x 100 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*				x 480 =	0.00
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						420
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	2240

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)		RATE($)	ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
17/671,199	02/14/2022		2240	L0041B1CIP2	20	3

CONFIRMATION NO. 1070

UPDATED FILING RECEIPT

182115
Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

OC000000133398119

Date Mailed: 04/26/2022

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

ÚLFAR ERLINGSSON, PALO ALTO, CA;
YIJOU CHEN, CUPERTINO, CA;

Applicant(s)

LACEWORK, INC., San Jose, CA;

Power of Attorney: The patent practitioners associated with Customer Number 182115

Domestic Priority data as claimed by applicant

This application is a CIP of 17/504,311 10/18/2021
which is a CON of 16/665,961 10/28/2019 PAT 11153339
which is a CON of 16/134,794 09/18/2018 PAT 10581891
which claims benefit of 62/590,986 11/27/2017
and claims benefit of 62/650,971 03/30/2018
This application 17/671,199
claims benefit of 63/243,013 09/10/2021

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00554

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 02/18/2022

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/671,199**

Projected Publication Date: 08/04/2022

Non-Publication Request: No

Early Publication Request: No

Title

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Preliminary Class

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00555

countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00556

technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00557

Replacement
Sheet - Fig. 2G

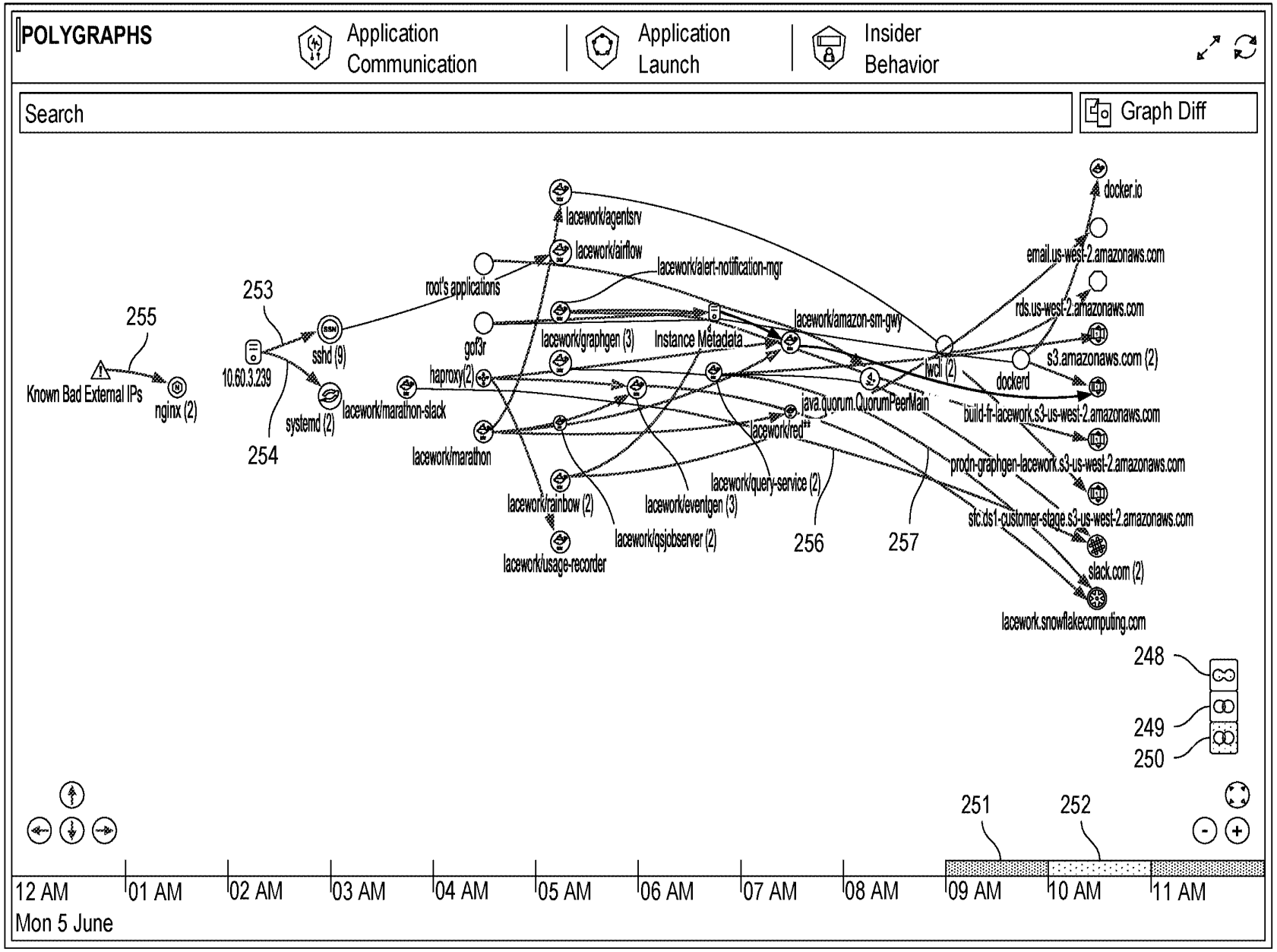


Fig. 2G

Replacement Sheet Markup - Fig. 7

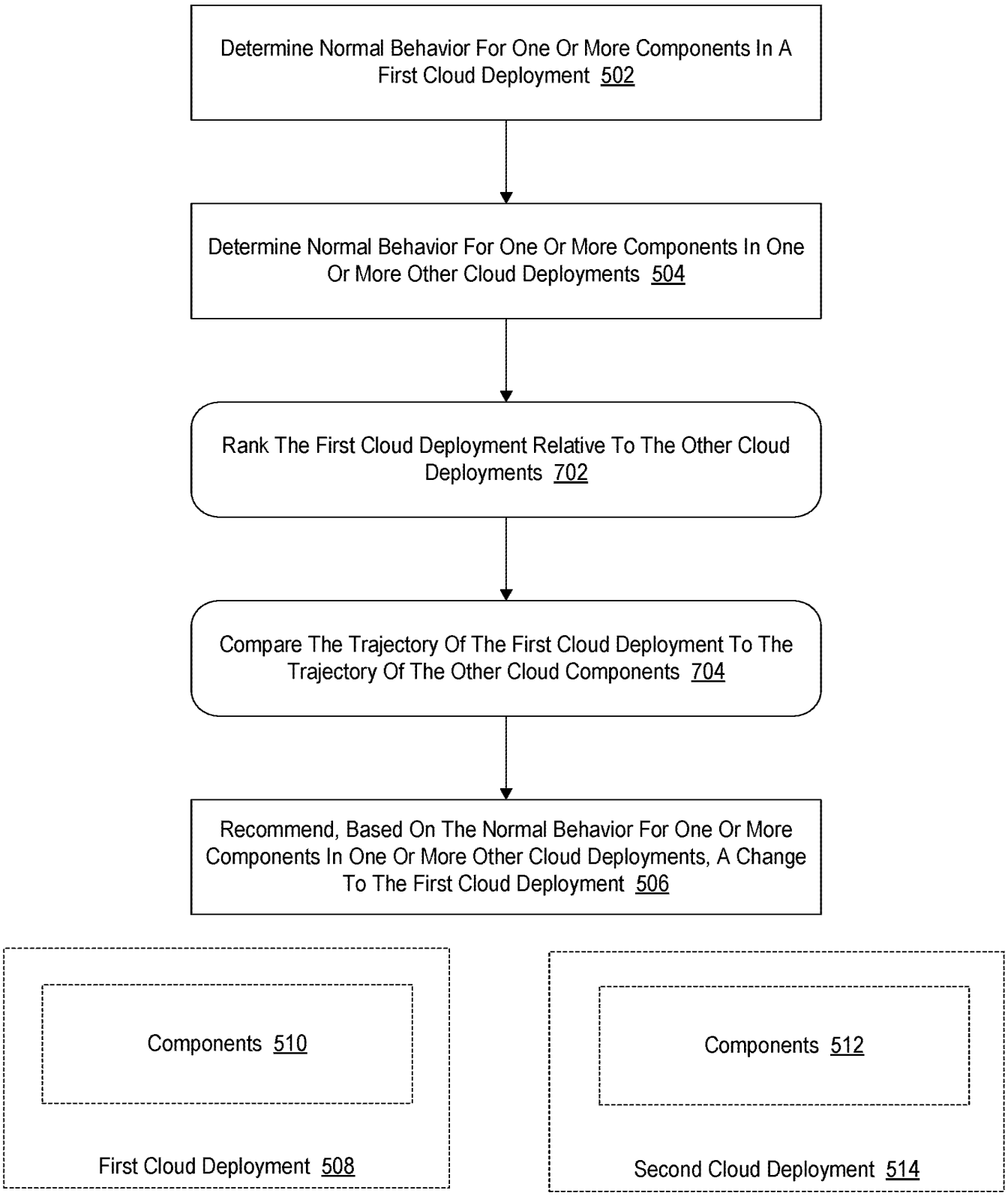


Fig. [[5]]7

Replacement Sheet Clean - Fig. 7

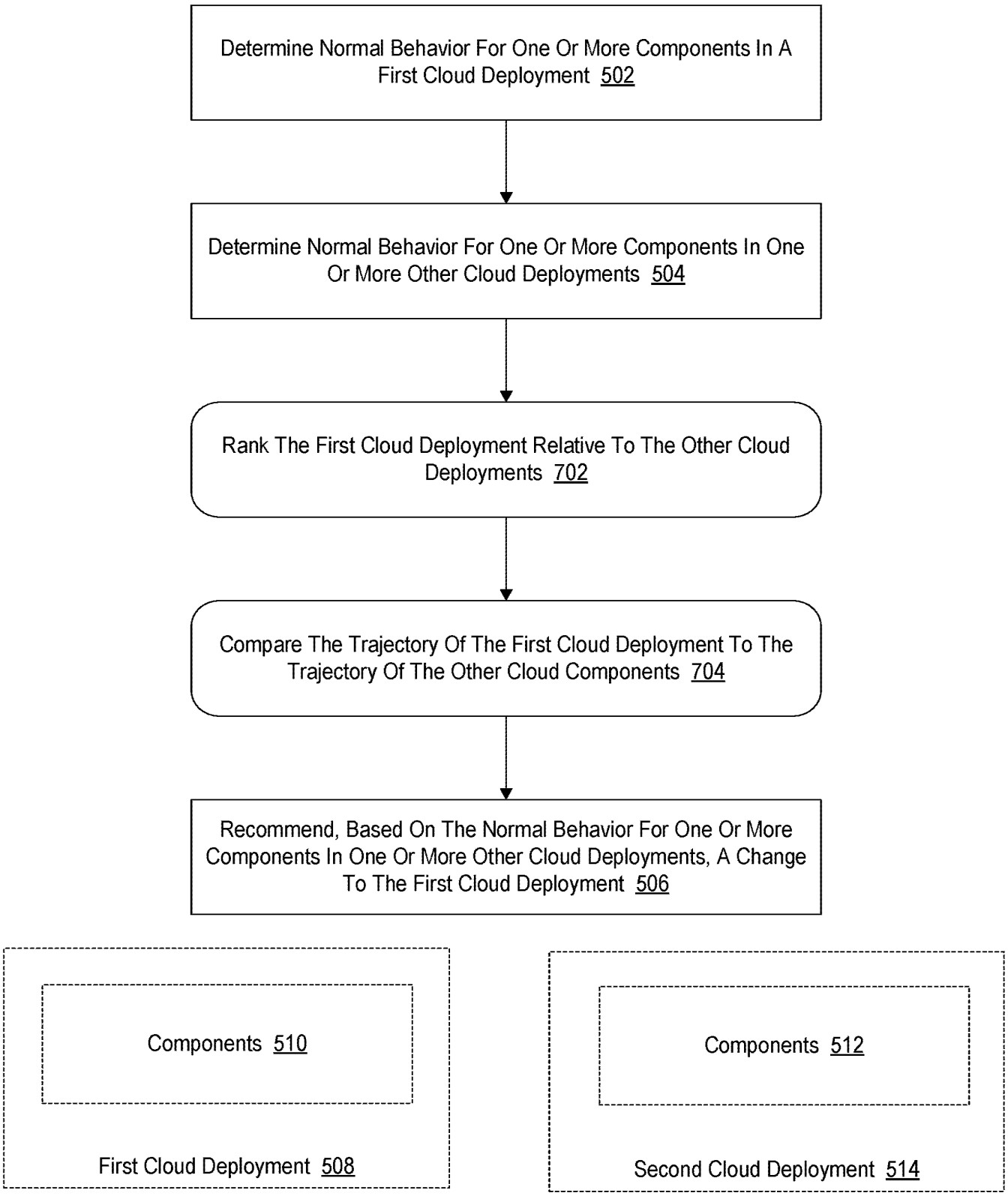


Fig. 7

Electronic Acknowledgement Receipt

EFS ID:	45536386
Application Number:	17671199
International Application Number:	
Confirmation Number:	1070
Title of Invention:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS
First Named Inventor/Applicant Name:	ÚLFAR ERLINGSSON
Customer Number:	182115
Filer:	Edward Joseph Lenart/Kate Hill
Filer Authorized By:	Edward Joseph Lenart
Attorney Docket Number:	L0041B1CIP2
Receipt Date:	22-APR-2022
Filing Date:	14-FEB-2022
Time Stamp:	10:42:03
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Applicant Response to Pre-Exam Formalities Notice	2022-04-22_L0041B1CIP2_ResponsetoCorrectedAppPapers.pdf	731250 5c8fa098afd897a561337e87757b3115909e17b8	no	5

Warnings:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00561

Information:					
2	Drawings-only black and white line drawings	2022-04-22_L0041B1CIP2_ReplacementDrawings.pdf	226475 b3f2166e06d75c8b61856f9da25e0b4b6ed15984	no	3
Warnings:					
Information:					
			Total Files Size (in bytes):	957725	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:	ÚLFAR ERLINGSSON
Serial No.:	17/671,199
Atty Docket No.:	L0041B1CIP2
Group Art Unit:	UNASSIGNED
Filed:	02/14/2022
Customer No.:	182115
Confirmation No.:	1070
Examiner:	UNASSIGNED
Title:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Mail Stop: Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

RESPONSE TO NOTICE TO FILE CORRECTED APPLICATION PAPERS

This is a Response to the Notice to File Corrected Application Papers dated February 22, 2022 (hereafter, "the Notice"), a copy of which is attached. Applicants present the following amendments and remarks.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00563

AMENDMENT

Please amend the above-identified application as follows:

Amendments to the Drawings:

Please replace Figure 2G and Figure 7 with the attached replacement sheets.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00564

REMARKS

The Notice indicates that Figure 2G cannot be reproduced due to blurriness. A replacement Figure 2G is included to correct these informalities. As no text has been added or removed (items have only been moved around), a drawing with markups is not attached.

The Notice also indicates that Figure 7 is not numbered numerically. A replacement Figure 7 is included to correct these informalities. A marked up version of the drawing is also included to identify the changes that were made to the original drawing.

The Commissioner is hereby authorized to charge or credit Deposit Account No. 50-3082 for any fees required or overpaid.

Respectfully submitted,

Date: April 22, 2022

By: /Edward J. Lenart/

Edward J. Lenart, Reg. No. 63,665
Kennedy Lenart Spraggins LLP
Tel. (512) 472-9881
Fax (512) 472-9887
ATTORNEY FOR APPLICANT

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00565

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	LACEP073

CONFIRMATION NO. 1070

182115
Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

FORMALITIES LETTER

OC00000131957340

Date Mailed: 02/22/2022

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given **TWO MONTHS** from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

The required item(s) identified below must be timely submitted to avoid abandonment.

- Replacement drawings in compliance with 37 CFR 1.84 and 37 CFR 1.121(d) are required. The drawings submitted are not acceptable because:
  - More than one figure is present and one or more figure appears to be unnumbered or misnumbered in comparison to the brief description of the figures. See Figure(s) 7. The figures must be labeled "Fig." with a consecutive Arabic numeral (1, 2, etc.) or an Arabic numeral and capital letter in the English alphabet (A, B, etc.) and correspond with a brief description having the same figure number. See 37 CFR 1.84(u)(1).
    (1) To correct the figure numbering, replacement figures must be submitted.
    (2) To add a brief description of an unnumbered figure, or correct the reference in the specification to the figures, a proper substitute specification in compliance with 37 CFR 1.121(b)(3) and 1.125, providing a brief description of each drawing figure presented, added or amended to correspond to the corrected numbering of the figures, must be submitted. See also 37 CFR 1.77(b)(9).
  - The drawings submitted to the Office are not electronically reproducible because portions of figures 2G are missing and/or blurry.

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

COPY

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice". https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

COPY

/bzewdie/

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00567

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
17/671,199	02/14/2022		2240	LACEP073	20	3

CONFIRMATION NO. 1070

CORRECTED FILING RECEIPT

182115
Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

OC000000132396755

Date Mailed: 03/22/2022

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

ÚLFAR ERLINGSSON, PALO ALTO, CA;
YIJOU CHEN, CUPERTINO, CA;

Applicant(s)

LACEWORK, INC., San Jose, CA;

Power of Attorney: The patent practitioners associated with Customer Number 182115

Domestic Priority data as claimed by applicant

This application is a CIP of 17/504,311 10/18/2021
which is a CON of 16/665,961 10/28/2019 PAT 11153339
which is a CON of 16/134,794 09/18/2018 PAT 10581891
which claims benefit of 62/590,986 11/27/2017
and claims benefit of 62/650,971 03/30/2018
This application 17/671,199
claims benefit of 63/243,013 09/10/2021

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00568

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 02/18/2022

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/671,199**

Projected Publication Date: To Be Determined - pending completion of Corrected Papers

Non-Publication Request: No

Early Publication Request: No

Title

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Preliminary Class

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00569

countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00570

technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00571

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor: Úlfar Erlingsson	§	Group Art Unit: UNKNOWN
	§	Examiner: UNKNOWN
Serial No.: 17/671,199	§	Atty Docket No.: LACEP073
Filed: 02/14/2022	§	Customer No.: 182115
Title: Learning From Similar Cloud Deployments	§	Confirmation No.: 1070

Mail Stop: Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR CORRECTION OF FILING RECEIPT DATED FEBRUARY 22, 2022

This is a Request for correction of the Filing Receipt dated February 22, 2022, for the above-referenced application. The Domestic Benefit/National Stage portion of the original Application Data Sheet (hereafter, "ADS"), filed with the application February 14, 2022, correctly included a priority claim to U.S. Patent Application No. 17/504,311 (hereafter, "the '311 application"). However, the '311 application claims the benefit of the following applications, which were inadvertently excluded from the original ADS:

Which is a CON of 16/665,961 filed 10/28/2019 (U.S. Patent No. 11,153,339)
Which is a CON of 16/134,794 filed 09/18/2018 (U.S. Patent No. 10,581,891)
Which claims benefit of U.S. Provisional Application No. 62/590,986 filed 11/27/2017
And claims benefit of U.S. Provisional Application No. 62/650,971 filed 03/30/2018

The subject application also claims benefit of U.S. Provisional Application No. 62/243,013 filed 09/10/2021, as presented in the originally filed ADS, and should remain as part of the priority claim.

Applicants file along with this request a supplemental ADS showing markup, which corrects the original ADS's Domestic Benefit/National Stage section, and respectfully request a corrected Filing Receipt.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00572

The Commissioner is hereby authorized to charge or credit Deposit Account No. 50-3082 for any fees required or overpaid.

Date: March 18, 2022

By: /Edward J. Lenart/

Edward J. Lenart, Reg. No. 63,665
Kennedy Lenart Spraggins LLP
Telephone: (512) 472-9881
Facsimile: (512) 472-9887
ATTORNEY FOR APPLICANTS

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00573

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Corrected Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	ÚLFAR		ERLINGSSON	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	PALO ALTO	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	6201 AMERICA CENTER DRIVE		
Address 2	SUITE 200		
City	SAN JOSE	State/Province	CA
Postal Code	95002	Country i	US

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	YIJOU		CHEN	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	CUPERTINO	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	6201 AMERICA CENTER DRIVE		
Address 2	SUITE 200		
City	SAN JOSE	State/Province	CA
Postal Code	95002	Country i	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Corrected **Application Data Sheet 37 CFR 1.76**	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

☐ **An Address is being provided for the correspondence Information of this application.**			
Customer Number	182115		
Email Address		Add Email	Remove Email

Application Information:

Title of the Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS		
Attorney Docket Number	LACEP073	**Small Entity Status Claimed**	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	49	**Suggested Figure for Publication (if any)**	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	182115		

Corrected **Application Data Sheet 37 CFR 1.76**	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation in part of	17504311	2021-10-18

Prior Application Status	Patented		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17504311	Continuation of	16665961	2019-10-28	11153339	2021-10-19

Prior Application Status	Patented		Remove		
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16665961	Continuation of	16134794	2018-09-18	10581891	2020-03-03

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16134794	Claims benefit of provisional	62590986	2017-11-27

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16134794	Claims benefit of provisional	62650971	2018-03-30

Prior Application Status			Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Claims benefit of provisional	63243013	2021-09-10

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button.

Foreign Priority Information:

Corrected **Application Data Sheet 37 CFR 1.76**	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Corrected Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Corrected Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

◉ Assignee	○ Legal Representative under 35 U.S.C. 117	○ Joint Inventor
○ Person to whom the inventor is obligated to assign.	○ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:	

If the Applicant is an Organization check here. ☒

Organization Name	LACEWORK, INC.

Mailing Address Information For Applicant:

Address 1	6201 AMERICA CENTER DRIVE		
Address 2	SUITE 200		
City	SAN JOSE	**State/Province**	CA
Country	US	Postal Code	95002
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button.

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Corrected **Application Data Sheet 37 CFR 1.76**	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.

Signature:

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	~~/Edward J. Lenart/~~ /Edward J. Lenart/			Date (YYYY-MM-DD)	2022-03-18 ~~2022-02-14~~
First Name	Edward J.	Last Name	Lenart	Registration Number	63665

Additional Signature may be generated within this form by selecting the Add button.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Corrected **Application Data Sheet 37 CFR 1.76**	Attorney Docket Number	LACEP073
	Application Number	17/671,199
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Acknowledgement Receipt

EFS ID:	45263013
Application Number:	17671199
International Application Number:	
Confirmation Number:	1070
Title of Invention:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS
First Named Inventor/Applicant Name:	ÚLFAR ERLINGSSON
Customer Number:	182115
Filer:	Edward Joseph Lenart/Ashley Jenkins
Filer Authorized By:	Edward Joseph Lenart
Attorney Docket Number:	LACEP073
Receipt Date:	18-MAR-2022
Filing Date:	14-FEB-2022
Time Stamp:	14:09:29
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Corrected Filing Receipt	2022-03-18_L0041B1CIP2_RequestforCorrectedFR.pdf	77481 47cea62f28278901addc117acc25fcaacd0e54ea	no	2

Warnings:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515

Information:					
2	Application Data Sheet	2022-03-18_L0041B1CIP2_CorrectedADS.pdf	929328 1438d3fac35f8bd721e762fd2a12fff9a665764d	no	9
Warnings:					
Information:					
This is not an USPTO supplied ADS fillable form					
			Total Files Size (in bytes):	1006809	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 17/671,199

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	320
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	700
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	800
TOTAL CLAIMS (37 CFR 1.16(i))	20 minus 20 =	*			OR	x 100 =	0.00
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	*				x 480 =	0.00
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						420
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0.00
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	2240

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)		RATE($)	ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
17/671,199	02/14/2022	ÚLFAR ERLINGSSON	LACEP073

CONFIRMATION NO. 1070

182115
Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

FORMALITIES LETTER

OC00000131957340

Date Mailed: 02/22/2022

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given **TWO MONTHS** from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

The required item(s) identified below must be timely submitted to avoid abandonment:

- Replacement drawings in compliance with 37 CFR 1.84 and 37 CFR 1.121(d) are required. The drawings submitted are not acceptable because:
  - More than one figure is present and one or more figure appears to be unnumbered or misnumbered in comparison to the brief description of the figures. See Figure(s) 7. The figures must be labeled "Fig." with a consecutive Arabic numeral (1, 2, etc.) or an Arabic numeral and capital letter in the English alphabet (A, B, etc.) and correspond with a brief description having the same figure number. See 37 CFR 1.84(u)(1).
    (1) To correct the figure numbering, replacement figures must be submitted.
    (2) To add a brief description of an unnumbered figure, or correct the reference in the specification to the figures, a proper substitute specification in compliance with 37 CFR 1.121(b)(3) and 1.125, providing a brief description of each drawing figure presented, added or amended to correspond to the corrected numbering of the figures, must be submitted. See also 37 CFR 1.77(b)(9).
  - The drawings submitted to the Office are not electronically reproducible because portions of figures 2G are missing and/or blurry.

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00586

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice". https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/bzewdie/

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00587

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
17/671,199	02/14/2022		2240	LACEP073	20	3

CONFIRMATION NO. 1070

FILING RECEIPT

182115
Lacework, Inc.
c/o Kennedy Lenart Spraggins LLP
797 Sam Bass Road #2559
ROUND ROCK, TX 78681

OC000000131957339

Date Mailed: 02/22/2022

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

ÚLFAR ERLINGSSON, PALO ALTO, CA;
YIJOU CHEN, CUPERTINO, CA;

Applicant(s)

LACEWORK, INC., San Jose, CA;

Power of Attorney: The patent practitioners associated with Customer Number 182115

Domestic Priority data as claimed by applicant

This application is a CIP of 17/504,311 10/18/2021
and claims benefit of 63/243,013 09/10/2021

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00588

If Required, Foreign Filing License Granted: 02/18/2022

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 17/671,199**

Projected Publication Date: To Be Determined - pending completion of Corrected Papers

Non-Publication Request: No

Early Publication Request: No
Title

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

Preliminary Class

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00589

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00590

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 03/31/2021. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	UNKNOWN
Filing Date	HEREWITH
First Named Inventor	ÚLFAR ERLINGSSON
Title	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS
Art Unit	UNKNOWN
Examiner Name	UNKNOWN
Attorney Docket Number	LACEP073

SIGNATURE of Applicant or Patent Practitioner

Signature	/Edward J. Lenart/	Date (Optional)	2022-02-14
Name	Edward J. Lenart	Registration Number	63665
Title (if Applicant is a juristic entity)	Patent Practitioner for Applicant		
Applicant Name (if Applicant is a juristic entity)	Lacework, Inc.		

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Doc Code: PA..

Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 03/31/2021. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 182115

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Lacework, Inc.

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	/Michael Moore/	Date (Optional)	
Name	Michael Moore		
Title	Vice President, Privacy and Intellectual Property		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ Total of forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

LEARNING FROM SIMILAR CLOUD DEPLOYMENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This is a continuation in-part application for patent entitled to a filing date and claiming the benefit of earlier-filed U.S. Patent Application No. 17/504,311, filed October 18, 2021, herein incorporated by reference in its entirety, and claims priority from U.S. Provisional Patent Application No. 63/243,013, filed September 10, 2021.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The accompanying drawings illustrate various embodiments and are a part of the specification. The illustrated embodiments are merely examples and do not limit the scope of the disclosure. Throughout the drawings, identical or similar reference numbers designate identical or similar elements.

[0003] Fig. 1A shows an illustrative configuration in which a data platform is configured to perform various operations with respect to a cloud environment that includes a plurality of compute assets.

[0004] Fig. 1B shows an illustrative implementation of the configuration of Fig. 1A.

[0005] Fig. 1C illustrates an example computing device.

[0006] Fig. 1D illustrates an example of an environment in which activities that occur within datacenters are modeled.

[0007] Fig. 2A illustrates an example of a process, used by an agent, to collect and report information about a client.

[0008] Fig. 2B illustrates a 5-tuple of data collected by an agent, physically and logically.

[0009] Fig. 2C illustrates a portion of a polygraph.

[0010] Fig. 2D illustrates a portion of a polygraph.

[0011] Fig. 2E illustrates an example of a communication polygraph.

[0012] Fig. 2F illustrates an example of a polygraph.

[0013] Fig. 2G illustrates an example of a polygraph as rendered in an interface.

[0014] Fig. 2H illustrates an example of a portion of a polygraph as rendered in an interface.

[0015] Fig. 2I illustrates an example of a portion of a polygraph as rendered in an interface.

[0016] Fig. 2J illustrates an example of a portion of a polygraph as rendered in an interface.

[0017] Fig. 2K illustrates an example of a portion of a polygraph as rendered in an interface.

[0018] Fig. 2L illustrates an example of an insider behavior graph as rendered in an interface.

[0019] Fig. 2M illustrates an example of a privilege change graph as rendered in an interface.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00594

[0020] Fig. 2N illustrates an example of a user login graph as rendered in an interface.

[0021] Fig. 2O illustrates an example of a machine server graph as rendered in an interface.

[0022] Fig. 3A illustrates an example of a process for detecting anomalies in a network environment.

[0023] Fig. 3B depicts a set of example processes communicating with other processes.

[0024] Fig. 3C depicts a set of example processes communicating with other processes.

[0025] Fig. 3D depicts a set of example processes communicating with other processes.

[0026] Fig. 3E depicts two pairs of clusters.

[0027] Fig. 3F is a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection.

[0028] Fig. 3G is an alternate representation of actions occurring in Fig. 3F.

[0029] Fig. 3H illustrates an example of a process for performing extended user tracking.

[0030] Fig. 3I is a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection.

[0031] Fig. 3J illustrates an example of a process for performing extended user tracking.

[0032] Fig. 3K illustrates example records.

[0033] Fig. 3L illustrates example output from performing an ssh connection match.

[0034] Fig. 3M illustrates example records.

[0035] Fig. 3N illustrates example records.

[0036] Fig. 3O illustrates example records.

[0037] Fig. 3P illustrates example records.

[0038] Fig. 3Q illustrates an adjacency relationship between two login sessions.

[0039] Fig. 3R illustrates example records.

[0040] Fig. 3S illustrates an example of a process for detecting anomalies.

[0041] Fig. 4A illustrates a representation of an embodiment of an insider behavior graph.

[0042] Fig. 4B illustrates an embodiment of a portion of an insider behavior graph.

[0043] Fig. 4C illustrates an embodiment of a portion of an insider behavior graph.

[0044] Fig. 4D illustrates an embodiment of a portion of an insider behavior graph.

[0045] Fig. 4E illustrates a representation of an embodiment of a user login graph.

[0046] Fig. 4F illustrates an example of a privilege change graph.

[0047] Fig. 4G illustrates an example of a privilege change graph.

[0048] Fig. 4H illustrates an example of a user interacting with a portion of an interface.

[0049] Fig. 4I illustrates an example of a dossier for an event.

[0050] Fig. 4J illustrates an example of a dossier for a domain.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00595

[0051] Fig. 4K depicts an example of an Entity Join graph by FilterKey and FilterKey Group (implicit join).

[0052] Fig. 4L illustrates an example process for dynamically generating and executing a query.

[0053] Fig. 5 sets forth a flowchart illustrating an example method of improving developer efficiency and application quality in accordance with some embodiments.

[0054] Fig. 6 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0055] Fig. 7 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0056] Fig. 8 sets forth a flowchart illustrating an additional example method of improving developer efficiency and application quality in accordance with some embodiments.

[0057] Fig. 9 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0058] Fig. 10 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

[0059] Fig. 11 sets forth a flowchart illustrating an additional example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure.

DETAILED DESCRIPTION

[0060] Various illustrative embodiments are described herein with reference to the accompanying drawings. It will, however, be evident that various modifications and changes may be made thereto, and additional embodiments may be implemented, without departing from the scope of the invention as set forth in the claims. For example, certain features of one embodiment described herein may be combined with or substituted for features of another embodiment described herein. The description and drawings are accordingly to be regarded in an illustrative rather than a restrictive sense.

[0061] Fig. 1A shows an illustrative configuration 10 in which a data platform 12 is configured to perform various operations with respect to a cloud environment 14 that includes a plurality of compute assets 16-1 through 16-N (collectively "compute assets 16). For example, data platform 12 may include data ingestion resources 18 configured to ingest data from cloud environment 14 into data platform 12, data processing resources 20 configured to perform data processing operations with respect to the data, user interface resources 22 configured to provide one or more external users and/or compute resources (e.g., computing device 24) with access to an output of data processing resources 20. Each of these resources are described in detail herein.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00596

[0062] Cloud environment 14 may include any suitable network-based computing environment as may serve a particular application. For example, cloud environment 14 may be implemented by one or more compute resources provided and/or otherwise managed by one or more cloud service providers, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), Microsoft Azure, and/or any other cloud service provider configured to provide public and/or private access to network-based compute resources.

[0063] Compute assets 16 may include, but are not limited to, containers (e.g., container images, deployed and executing container instances, etc.), virtual machines, workloads, applications, processes, physical machines, compute nodes, clusters of compute nodes, software runtime environments (e.g., container runtime environments), and/or any other virtual and/or physical compute resource that may reside in and/or be executed by one or more computer resources in cloud environment 14. In some examples, one or more compute assets 16 may reside in one or more datacenters.

[0064] A compute asset 16 may be associated with (e.g., owned, deployed, or managed by) a particular entity, such as a customer or client of cloud environment 14 and/or data platform 12. Accordingly, for purposes of the discussion herein, cloud environment 14 may be used by one or more entities.

[0065] Data platform 12 may be configured to perform one or more data security monitoring and/or remediation services, compliance monitoring services, anomaly detection services, DevOps services, compute asset management services, and/or any other type of data analytics service as may serve a particular implementation. Data platform 12 may be managed or otherwise associated with any suitable data platform provider, such as a provider of any of the data analytics services described herein. The various resources included in data platform 12 may reside in the cloud and/or be located on-premises and be implemented by any suitable combination of physical and/or virtual compute resources, such as one or more computing devices, microservices, applications, etc.

[0066] Data ingestion resources 18 may be configured to ingest data from cloud environment 14 into data platform 12. This may be performed in various ways, some of which are described in detail herein. For example, as illustrated by arrow 26, data ingestion resources 18 may be configured to receive the data from one or more agents deployed within cloud environment 14, utilize an event streaming platform (e.g., Kafka) to obtain the data, and/or pull data (e.g., configuration data) from cloud environment 14. In some examples, data ingestion resources 18 may obtain the data using one or more agentless configurations.

[0067] The data ingested by data ingestion resources 18 from cloud environment 14 may include any type of data as may serve a particular implementation. For example, the data may include

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00597

data representative of configuration information associated with compute assets 16, information about one or more processes running on compute assets 16, network activity information, information about events (creation events, modification events, communication events, user-initiated events, etc.) that occur with respect to compute assets 16, etc. In some examples, the data may or may not include actual customer data processed or otherwise generated by compute assets 16.

[0068] As illustrated by arrow 28, data ingestion resources 18 may be configured to load the data ingested from cloud environment 14 into a data store 30. Data store 30 is illustrated in Fig. 1A as being separate from and communicatively coupled to data platform 12. However, in some alternative embodiments, data store 30 is included within data platform 12.

[0069] Data store 30 may be implemented by any suitable data warehouse, data lake, data mart, and/or other type of database structure as may serve a particular implementation. Such data stores may be proprietary or may be embodied as vendor provided products or services such as, for example, Snowflake, Google BigQuery, Druid, Amazon Redshift, IBM Db2, Dremio, Databricks Lakehouse Platform, Cloudera, Azure Synapse Analytics, and others.

[0070] Although the examples described herein largely relate to embodiments where data is collected from agents and ultimately stored in a data store such as those provided by Snowflake, in other embodiments data that is collected from agents and other sources may be stored in different ways. For example, data that is collected from agents and other sources may be stored in a data warehouse, data lake, data mart, and/or any other data store.

[0071] A data warehouse may be embodied as an analytic database (e.g., a relational database) that is created from two or more data sources. Such a data warehouse may be leveraged to store historical data, often on the scale of petabytes. Data warehouses may have compute and memory resources for running complicated queries and generating reports. Data warehouses may be the data sources for business intelligence ('BI') systems, machine learning applications, and/or other applications. By leveraging a data warehouse, data that has been copied into the data warehouse may be indexed for good analytic query performance, without affecting the write performance of a database (e.g., an Online Transaction Processing ('OLTP') database). Data warehouses also enable the joining data from multiple sources for analysis. For example, a sales OLTP application probably has no need to know about the weather at various sales locations, but sales predictions could take advantage of that data. By adding historical weather data to a data warehouse, it would be possible to factor it into models of historical sales data.

[0072] Data lakes, which store files of data in their native format, may be considered as "schema on read" resources. As such, any application that reads data from the lake may impose its own types and relationships on the data. Data warehouses, on the other hand, are "schema on write,"

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00598

meaning that data types, indexes, and relationships are imposed on the data as it is stored in an enterprise data warehouse (EDW). "Schema on read" resources may be beneficial for data that may be used in several contexts and poses little risk of losing data. "Schema on write" resources may be beneficial for data that has a specific purpose, and good for data that must relate properly to data from other sources. Such data stores may include data that is encrypted using homomorphic encryption, data encrypted using privacy-preserving encryption, smart contracts, non-fungible tokens, decentralized finance, and other techniques.

[0073] Data marts may contain data oriented towards a specific business line whereas data warehouses contain enterprise-wide data. Data marts may be dependent on a data warehouse, independent of the data warehouse (e.g., drawn from an operational database or external source), or a hybrid of the two. In embodiments described herein, different types of data stores (including combinations thereof) may be leveraged.

[0074] Data processing resources 20 may be configured to perform various data processing operations with respect to data ingested by data ingestion resources 18, including data ingested and stored in data store 30. For example, data processing resources 20 may be configured to perform one or more data security monitoring and/or remediation operations, compliance monitoring operations, anomaly detection operations, DevOps operations, compute asset management operations, and/or any other type of data analytics operation as may serve a particular implementation. Various examples of operations performed by data processing resources 20 are described herein.

[0075] As illustrated by arrow 32, data processing resources 20 may be configured to access data in data store 30 to perform the various operations described herein. In some examples, this may include performing one or more queries with respect to the data stored in data store 30. Such queries may be generated using any suitable query language.

[0076] In some examples, the queries provided by data processing resources 20 may be configured to direct data store 30 to perform one or more data analytics operations with respect to the data stored within data store 30. These data analytics operations may be with respect to data specific to a particular entity (e.g., data residing in one or more silos within data store 30 that are associated with a particular customer) and/or data associated with multiple entities. For example, data processing resources 20 may be configured to analyze data associated with a first entity and use the results of the analysis to perform one or more operations with respect to a second entity.

[0077] One or more operations performed by data processing resources 20 may be performed periodically according to a predetermined schedule. For example, one or more operations may be performed by processing resources 20 every hour or any other suitable time interval. Additionally or alternatively, one or more operations performed by data processing resources 20 may be

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00599

performed in substantially real-time (or near real-time) as data is ingested into data platform 12. In this manner, the results of such operations (e.g., one or more detected anomalies in the data) may be provided to one or more external entities (e.g., computing device 24 and/or one or more users) in substantially real-time and/or in near real-time.

[0078] User interface resources 22 may be configured to perform one or more user interface operations, examples of which are described herein. For example, user interface resources 22 may be configured to present one or more results of the data processing performed by data processing resources 20 to one or more external entities (e.g., computing device 24 and/or one or more users), as illustrated by arrow 34. As illustrated by arrow 36, user interface resources 22 may access data in data store 30 to perform the one or more user interface operations

[0079] Fig. 1B illustrates an implementation of configuration 10 in which an agent 38 (e.g., agent 38-1 through agent 38-N) is installed on each of compute assets 16. As used herein, an agent may include a self-contained binary and/or other type of code or application that can be run on any appropriate platforms, including within containers and/or other virtual compute assets. Agents 38 may monitor the nodes on which they execute for a variety of different activities, including but not limited to, connection, process, user, machine, and file activities. In some examples, agents 38 can be executed in user space, and can use a variety of kernel modules (e.g., auditd, iptables, netfilter, pcap, etc.) to collect data. Agents can be implemented in any appropriate programming language, such as C or Golang, using applicable kernel APIs.

[0080] Agents 38 may be deployed in any suitable manner. For example, an agent 38 may be deployed as a containerized application or as part of a containerized application. As described herein, agents 38 may selectively report information to data platform 12 in varying amounts of detail and/or with variable frequency.

[0081] Also shown in Fig. 1B is a load balancer 40 configured to perform one or more load balancing operations with respect to data ingestion operations performed by data ingestion resources 18 and/or user interface operations performed by user interface resources 22. Load balancer 40 is shown to be included in data platform 12. However, load balancer 40 may alternatively be located external to data platform 12. Load balancer 40 may be implemented by any suitable microservice, application, and/or other computing resources. In some alternative examples, data platform 12 may not utilize a load balancer such as load balancer 40.

[0082] Also shown in Fig. 1B is long term storage 42 with which data ingestion resources may interface, as illustrated by arrow 44. Long term storage 42 may be implemented by any suitable type of storage resources, such as cloud-based storage (e.g., AWS S3, etc.) and/or on-premises storage and may be used by data ingestion resources 18 as part of the data ingestion process.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00600

Examples of this are described herein. In some examples, data platform 12 may not utilize long term storage 42.

[0083] The embodiments described herein can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the principles described herein. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0084] In some examples, a non-transitory computer-readable medium storing computer-readable instructions may be provided in accordance with the principles described herein. The instructions, when executed by a processor of a computing device, may direct the processor and/or computing device to perform one or more operations, including one or more of the operations described herein. Such instructions may be stored and/or transmitted using any of a variety of known computer-readable media.

[0085] A non-transitory computer-readable medium as referred to herein may include any non-transitory storage medium that participates in providing data (e.g., instructions) that may be read and/or executed by a computing device (e.g., by a processor of a computing device). For example, a non-transitory computer-readable medium may include, but is not limited to, any combination of non-volatile storage media and/or volatile storage media. Exemplary non-volatile storage media include, but are not limited to, read-only memory, flash memory, a solid-state drive, a magnetic storage device (e.g. a hard disk, a floppy disk, magnetic tape, etc.), ferroelectric random-access memory ("RAM"), and an optical disc (e.g., a compact disc, a digital video disc, a Blu-ray disc, etc.). Exemplary volatile storage media include, but are not limited to, RAM (e.g., dynamic RAM).

[0086] Fig. 1C illustrates an example computing device 50 that may be specifically configured to perform one or more of the processes described herein. Any of the systems, microservices, computing devices, and/or other components described herein may be implemented by computing device 50.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00601

[0087] As shown in Fig. 1C, computing device 50 may include a communication interface 52, a processor 54, a storage device 56, and an input/output ("I/O") module 58 communicatively connected one to another via a communication infrastructure 60. While an exemplary computing device 50 is shown in Fig. 1C, the components illustrated in Fig. 1C are not intended to be limiting. Additional or alternative components may be used in other embodiments. Components of computing device 50 shown in Fig. 1C will now be described in additional detail.

[0088] Communication interface 52 may be configured to communicate with one or more computing devices. Examples of communication interface 52 include, without limitation, a wired network interface (such as a network interface card), a wireless network interface (such as a wireless network interface card), a modem, an audio/video connection, and any other suitable interface.

[0089] Processor 54 generally represents any type or form of processing unit capable of processing data and/or interpreting, executing, and/or directing execution of one or more of the instructions, processes, and/or operations described herein. Processor 54 may perform operations by executing computer-executable instructions 62 (e.g., an application, software, code, and/or other executable data instance) stored in storage device 56.

[0090] Storage device 56 may include one or more data storage media, devices, or configurations and may employ any type, form, and combination of data storage media and/or device. For example, storage device 56 may include, but is not limited to, any combination of the non-volatile media and/or volatile media described herein. Electronic data, including data described herein, may be temporarily and/or permanently stored in storage device 56. For example, data representative of computer-executable instructions 62 configured to direct processor 54 to perform any of the operations described herein may be stored within storage device 56. In some examples, data may be arranged in one or more databases residing within storage device 56.

[0091] I/O module 58 may include one or more I/O modules configured to receive user input and provide user output. I/O module 58 may include any hardware, firmware, software, or combination thereof supportive of input and output capabilities. For example, I/O module 58 may include hardware and/or software for capturing user input, including, but not limited to, a keyboard or keypad, a touchscreen component (e.g., touchscreen display), a receiver (e.g., an RF or infrared receiver), motion sensors, and/or one or more input buttons.

[0092] I/O module 58 may include one or more devices for presenting output to a user, including, but not limited to, a graphics engine, a display (e.g., a display screen), one or more output drivers (e.g., display drivers), one or more audio speakers, and one or more audio drivers. In certain embodiments, I/O module 58 is configured to provide graphical data to a display for presentation

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00602

to a user. The graphical data may be representative of one or more graphical user interfaces and/or any other graphical content as may serve a particular implementation.

[0093] Fig. 1D illustrates an example implementation 100 of configuration 10. As such, one more components shown in Fig. 1D may implement one or more components shown in Fig. 1A and/or Fig. 1B. In particular, implementation 100 illustrates an environment in which activities that occur within datacenters are modeled using data platform 12. Using techniques described herein, a baseline of datacenter activity can be modeled, and deviations from that baseline can be identified as anomalous. Anomaly detection can be beneficial in a security context, a compliance context, an asset management context, a DevOps context, and/or any other data analytics context as may serve a particular implementation.

[0094] Two example datacenters (104 and 106) are shown in Fig. 1D, and are associated with (e.g., belong to) entities named entity A and entity B, respectively. A datacenter may include dedicated equipment (e.g., owned and operated by entity A, or owned/leased by entity A and operated exclusively on entity A's behalf by a third party). A datacenter can also include cloud-based resources, such as infrastructure as a service (IaaS), platform as a service (PaaS), and/or software as a service (SaaS) elements. The techniques described herein can be used in conjunction with multiple types of datacenters, including ones wholly using dedicated equipment, ones that are entirely cloud-based, and ones that use a mixture of both dedicated equipment and cloud-based resources.

[0095] Both datacenter 104 and datacenter 106 include a plurality of nodes, depicted collectively as set of nodes 108 and set of nodes 110, respectively, in Fig. 1D. These nodes may implement compute assets 16. Installed on each of the nodes are in-server / in-virtual machine (VM) / embedded in IoT device agents (e.g., agent 112), which are configured to collect data and report it to data platform 12 for analysis. As described herein, agents may be small, self-contained binaries that can be run on any appropriate platforms, including virtualized ones (and, as applicable, within containers). Agents may monitor the nodes on which they execute for a variety of different activities, including: connection, process, user, machine, and file activities. Agents can be executed in user space, and can use a variety of kernel modules (e.g., auditd, iptables, netfilter, pcap, etc.) to collect data. Agents can be implemented in any appropriate programming language, such as C or Golang, using applicable kernel APIs.

[0096] As described herein, agents can selectively report information to data platform 12 in varying amounts of detail and/or with variable frequency. As is also described herein, the data collected by agents may be used by data platform 12 to create polygraphs, which are graphs of logical entities, connected by behaviors. In some embodiments, agents report information directly to data platform 12. In other embodiments, at least some agents provide information to a data

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00603

aggregator, such as data aggregator 114, which in turn provides information to data platform 12. The functionality of a data aggregator can be implemented as a separate binary or other application (distinct from an agent binary), and can also be implemented by having an agent execute in an "aggregator mode" in which the designated aggregator node acts as a Layer 7 proxy for other agents that do not have access to data platform 12. Further, a chain of multiple aggregators can be used, if applicable (e.g., with agent 112 providing data to data aggregator 114, which in turn provides data to another aggregator (not pictured) which provides data to data platform 12). An example way to implement an aggregator is through a program written in an appropriate language, such as C or Golang.

[0097] Use of an aggregator can be beneficial in sensitive environments (e.g., involving financial or medical transactions) where various nodes are subject to regulatory or other architectural requirements (e.g., prohibiting a given node from communicating with systems outside of datacenter 104). Use of an aggregator can also help to minimize security exposure more generally. As one example, by limiting communications with data platform 12 to data aggregator 114, individual nodes in nodes 108 need not make external network connections (e.g., via Internet 124), which can potentially expose them to compromise (e.g., by other external devices, such as device 118, operated by a criminal). Similarly, data platform 12 can provide updates, configuration information, etc., to data aggregator 114 (which in turn distributes them to nodes 108), rather than requiring nodes 108 to allow incoming connections from data platform 12 directly.

[0098] Another benefit of an aggregator model is that network congestion can be reduced (e.g., with a single connection being made at any given time between data aggregator 114 and data platform 12, rather than potentially many different connections being open between various of nodes 108 and data platform 12). Similarly, network consumption can also be reduced (e.g., with the aggregator applying compression techniques/bundling data received from multiple agents).

[0099] One example way that an agent (e.g., agent 112, installed on node 116) can provide information to data aggregator 114 is via a REST API, formatted using data serialization protocols such as Apache Avro. One example type of information sent by agent 112 to data aggregator 114 is status information. Status information may be sent by an agent periodically (e.g., once an hour or once any other predetermined amount of time). Alternatively, status information may be sent continuously or in response to occurrence of one or more events. The status information may include, but is not limited to, a. an amount of event backlog (in bytes) that has not yet been transmitted, b. configuration information, c. any data loss period for which data was dropped, d. a cumulative count of errors encountered since the agent started, e. version

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00604

information for the agent binary, and/or f. cumulative statistics on data collection (e.g., number of network packets processed, new processes seen, etc.).

[00100] A second example type of information that may be sent by agent 112 to data aggregator 114 is event data (described in more detail herein), which may include a UTC timestamp for each event. As applicable, the agent can control the amount of data that it sends to the data aggregator in each call (e.g., a maximum of 10MB) by adjusting the amount of data sent to manage the conflicting goals of transmitting data as soon as possible, and maximizing throughput. Data can also be compressed or uncompressed by the agent (as applicable) prior to sending the data.

[00101] Each data aggregator may run within a particular customer environment. A data aggregator (e.g., data aggregator 114) may facilitate data routing from many different agents (e.g., agents executing on nodes 108) to data platform 12. In various embodiments, data aggregator 114 may implement a SOCKS 5 caching proxy through which agents can connect to data platform 12. As applicable, data aggregator 114 can encrypt (or otherwise obfuscate) sensitive information prior to transmitting it to data platform 12, and can also distribute key material to agents which can encrypt the information (as applicable). Data aggregator 114 may include a local storage, to which agents can upload data (e.g., pcap packets). The storage may have a key-value interface. The local storage can also be omitted, and agents configured to upload data to a cloud storage or other storage area, as applicable. Data aggregator 114 can, in some embodiments, also cache locally and distribute software upgrades, patches, or configuration information (e.g., as received from data platform 12).

[00102] Various examples associated with agent data collection and reporting will now be described.

[00103] In the following example, suppose that a user (e.g., a network administrator) at entity A (hereinafter "user A") has decided to begin using the services of data platform 12. In some embodiments, user A may access a web frontend (e.g., web app 120) using a computer 126 and enrolls (on behalf of entity A) an account with data platform 12. After enrollment is complete, user A may be presented with a set of installers, pre-built and customized for the environment of entity A, that user A can download from data platform 12 and deploy on nodes 108. Examples of such installers include, but are not limited to, a Windows executable file, an iOS app, a Linux package (e.g., .deb or .rpm), a binary, or a container (e.g., a Docker container). When a user (e.g., a network administrator) at entity B (hereinafter "user B") also signs up for the services of data platform 12, user B may be similarly presented with a set of installers that are pre-built and customized for the environment of entity B.

[00104] User A deploys an appropriate installer on each of nodes 108 (e.g., with a Windows executable file deployed on a Windows-based platform or a Linux package deployed on a Linux

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00605

platform, as applicable). As applicable, the agent can be deployed in a container. Agent deployment can also be performed using one or more appropriate automation tools, such as Chef, Puppet, Salt, and Ansible. Deployment can also be performed using managed/hosted container management/orchestration frameworks such as Kubernetes, Mesos, and/or Docker Swarm.

[00105] In various embodiments, the agent may be installed in the user space (i.e., is not a kernel module), and the same binary is executed on each node of the same type (e.g., all Windows-based platforms have the same Windows-based binary installed on them). An illustrative function of an agent, such as agent 112, is to collect data (e.g., associated with node 116) and report it (e.g., to data aggregator 114). Other tasks that can be performed by agents include data configuration and upgrading.

[00106] One approach to collecting data as described herein is to collect virtually all information available about a node (and, e.g., the processes running on it). Alternatively, the agent may monitor for network connections, and then begin collecting information about processes associated with the network connections, using the presence of a network packet associated with a process as a trigger for collecting additional information about the process. As an example, if a user of node 116 executes an application, such as a calculator application, which does not typically interact with the network, no information about use of that application may be collected by agent 112 and/or sent to data aggregator 114. If, however, the user of node 116 executes an ssh command (e.g., to ssh from node 116 to node 122), agent 112 may collect information about the process and provide associated information to data aggregator 114. In various embodiments, the agent may always collect/report information about certain events, such as privilege escalation, irrespective of whether the event is associated with network activity.

[00107] An approach to collecting information (e.g., by an agent) is as follows, and described in conjunction with process 200 depicted in Fig. 2A. An agent (e.g., agent 112) monitors its node (e.g., node 116) for network activity. One example way that agent 112 can monitor node 116 for network activity is by using a network packet capture tool (e.g., listening using libpcap). As packets are received (201), the agent obtains and maintains (e.g., in an in-memory cache) connection information associated with the network activity (202). Examples of such information include DNS query/response, TCP, UDP, and IP information.

[00108] The agent may also determine a process associated with the network connection (203). One example approach is for the agent to use a kernel network diagnostic API (e.g., netlink_diag) to obtain inode/process information from the kernel. Another example approach is for the agent to scan using netstat (e.g., on /proc/net/tcp, /proc/net/tcp6, /proc/net/udp, and /proc/net/udp6) to obtain sockets and relate them to processes. Information such as socket state (e.g., whether a socket is connected, listening, etc.) can also be collected by the agent.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00606

[00109] One way an agent can obtain a mapping between a given inode and a process identifier is to scan within the /proc/pid directory. For each of the processes currently running, the agent examines each of their file descriptors. If a file descriptor is a match for the inode, the agent can determine that the process associated with the file descriptor owns the inode. Once a mapping is determined between an inode and a process identifier, the mapping is cached. As additional packets are received for the connection, the cached process information is used (rather than a new search being performed).

[00110] In some cases, exhaustively scanning for an inode match across every file descriptor may not be feasible (e.g., due to CPU limitations). In various embodiments, searching through file descriptors is accordingly optimized. User filtering is one example of such an optimization. A given socket is owned by a user. Any processes associated with the socket will be owned by the same user as the socket. When matching an inode (identified as relating to a given socket) against processes, the agent can filter through the processes and only examine the file descriptors of processes sharing the same user owner as the socket. In various embodiments, processes owned by root are always searched against (e.g., even when user filtering is employed).

[00111] Another example of an optimization is to prioritize searching the file descriptors of certain processes over others. One such prioritization is to search through the subdirectories of /proc/ starting with the youngest process. One approximation of such a sort order is to search through /proc/ in reverse order (e.g., examining highest numbered processes first). Higher numbered processes are more likely to be newer (i.e., not long-standing processes), and thus more likely to be associated with new connections (i.e., ones for which inode-process mappings are not already cached). In some cases, the most recently created process may not have the highest process identifier (e.g., due to the kernel wrapping through process identifiers).

[00112] Another example prioritization is to query the kernel for an identification of the most recently created process and to search in a backward order through the directories in /proc/ (e.g., starting at the most recently created process and working backwards, then wrapping to the highest value (e.g., 32768) and continuing to work backward from there). An alternate approach is for the agent to keep track of the newest process that it has reported information on (e.g., to data aggregator 114), and begin its search of /proc/ in a forward order starting from the PID of that process.

[00113] Another example prioritization is to maintain, for each user actively using node 116, a list of the five (or any other number) most recently active processes. Those processes are more likely than other processes (less active, or passive) on node 116 to be involved with new connections, and can thus be searched first. For many processes, lower valued file descriptors tend to correspond to non-sockets (e.g., stdin, stdout, stderr). Yet another optimization is to

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00607

preferentially search higher valued file descriptors (e.g., across processes) over lower valued file descriptors (that are less likely to yield matches).

[00114] In some cases, while attempting to locate a process identifier for a given inode, an agent may encounter a socket that does not correspond to the inode being matched against and is not already cached. The identity of that socket (and its corresponding inode) can be cached, once discovered, thus removing a future need to search for that pair.

[00115] In some cases, a connection may terminate before the agent is able to determine its associated process (e.g., due to a very short-lived connection, due to a backlog in agent processing, etc.). One approach to addressing such a situation is to asynchronously collect information about the connection using the audit kernel API, which streams information to user space. The information collected from the audit API (which can include PID/inode information) can be matched by the agent against pcap/inode information. In some embodiments, the audit API is always used, for all connections. However, due to CPU utilization considerations, use of the audit API can also be reserved for short/otherwise problematic connections (and/or omitted, as applicable).

[00116] Once the agent has determined which process is associated with the network connection (203), the agent can then collect additional information associated with the process (204). As will be described in more detail below, some of the collected information may include attributes of the process (e.g., a process parent hierarchy, and an identification of a binary associated with the process). As will also be described in more detail below, other of the collected information is derived (e.g., session summarization data and hash values).

[00117] The collected information is then transmitted (205), e.g., by an agent (e.g., agent 112) to a data aggregator (e.g., data aggregator 114), which in turn provides the information to data platform 12. In some embodiments, all information collected by an agent may be transmitted (e.g., to a data aggregator and/or to data platform 12). In other embodiments, the amount of data transmitted may be minimized (e.g., for efficiency reasons), using various techniques.

[00118] One approach to minimizing the amount of data flowing from agents (such as agents installed on nodes 108) to data platform 12 is to use a technique of implicit references with unique keys. The keys can be explicitly used by data platform 12 to extract/derive relationships, as necessary, in a data set at a later time, without impacting performance.

[00119] As previously mentioned, some data collected about a process is constant and does not change over the lifetime of the process (e.g., attributes), and some data changes (e.g., statistical information and other variable information). Constant data can be transmitted (210) once, when the agent first becomes aware of the process. And, if any changes to the constant data are

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00608

detected (e.g., a process changes its parent), a refreshed version of the data can be transmitted (210) as applicable.

[00120] In some examples, an agent may collect variable data (e.g., data that may change over the lifetime of the process). In some examples, variable data can be transmitted (210) at periodic (or other) intervals. Alternatively, variable data may be transmitted in substantially real time as it is collected. In some examples, the variable data may indicate a thread count for a process, a total virtual memory used by the process, the total resident memory used by the process, the total time spent by the process executing in user space, and/or the total time spent by the process executing in kernel space. In some examples, the data may include a hash that may be used within data platform 12 to join process creation time attributes with runtime attributes to construct a full dataset.

[00121] Below are additional examples of data that an agent, such as agent 112, can collect and provide to data platform 12.

[00122] 1. User Data

[00123] Core User Data: user name, UID (user ID), primary group, other groups, home directory.

[00124] Failed Login Data: IP address, hostname, username, count.

[00125] User Login Data: user name, hostname, IP address, start time, TTY (terminal), UID (user ID), GID (group ID), process, end time.

[00126] 2. Machine Data

[00127] Dropped Packet Data: source IP address, destination IP address, destination port, protocol, count.

[00128] Machine Data: hostname, domain name, architecture, kernel, kernel release, kernel version, OS, OS version, OS description, CPU, memory, model number, number of cores, last boot time, last boot reason, tags (e.g., Cloud provider tags such as AWS, GCP, or Azure tags), default router, interface name, interface hardware address, interface IP address and mask, promiscuous mode.

[00129] 3. Network Data

[00130] Network Connection Data: source IP address, destination IP address, source port, destination port, protocol, start time, end time, incoming and outgoing bytes, source process, destination process, direction of connection, histograms of packet length, inter packet delay, session lengths, etc.

[00131] Listening Ports in Server: source IP address, port number, protocol, process.

[00132] Dropped Packet Data: source IP address, destination IP address, destination port, protocol, count.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00609

[00133] Arp Data: source hardware address, source IP address, destination hardware address, destination IP address.

[00134] DNS Data: source IP address, response code, response string, question (request), packet length, final answer (response).

[00135] 4. Application Data

[00136] Package Data: exe path, package name, architecture, version, package path, checksums (MD5, SHA-1, SHA-256), size, owner, owner ID.

[00137] Application Data: command line, PID (process ID), start time, UID (user ID), EUID (effective UID), PPID (parent process ID), PGID (process group ID), SID (session ID), exe path, username, container ID.

[00138] 5. Container Data

[00139] Container Image Data: image creation time, parent ID, author, container type, repo, (AWS) tags, size, virtual size, image version.

[00140] Container Data: container start time, container type, container name, container ID, network mode, privileged, PID mode, IP addresses, listening ports, volume map, process ID.

[00141] 6. File Data

[00142] File path, file data hash, symbolic links, file creation data, file change data, file metadata, file mode.

[00143] As mentioned above, an agent, such as agent 112, can be deployed in a container (e.g., a Docker container), and can also be used to collect information about containers. Collection about a container can be performed by an agent irrespective of whether the agent is itself deployed in a container or not (as the agent can be deployed in a container running in a privileged mode that allows for monitoring).

[00144] Agents can discover containers (e.g., for monitoring) by listening for container create events (e.g., provided by Docker), and can also perform periodic ordered discovery scans to determine whether containers are running on a node. When a container is discovered, the agent can obtain attributes of the container, e.g., using standard Docker API calls (e.g., to obtain IP addresses associated with the container, whether there's a server running inside, what port it is listening on, associated PIDs, etc.). Information such as the parent process that started the container can also be collected, as can information about the image (which comes from the Docker repository).

[00145] In various embodiments, agents may use namespaces to determine whether a process is associated with a container. Namespaces are a feature of the Linux kernel that can be used to isolate resources of a collection of processes. Examples of namespaces include process ID (PID) namespaces, network namespaces, and user namespaces. Given a process, the agent can perform

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00610

a fast lookup to determine whether the process is part of the namespace the container claims to be its namespace.

[00146] As mentioned, agents can be configured to report certain types of information (e.g., attribute information) once, when the agent first becomes aware of a process. In various embodiments, such static information is not reported again (or is reported once a day, every twelve hours, etc.), unless it changes (e.g., a process changes its parent, changes its owner, or a SHA-1 of the binary associated with the process changes).

[00147] In contrast to static/attribute information, certain types of data change constantly (e.g., network-related data). In various embodiments, agents are configured to report a list of current connections every minute (or other appropriate time interval). In that connection list will be connections that started in that minute interval, connections that ended in that minute interval, and connections that were ongoing throughout the minute interval (e.g., a one minute slice of a one hour connection).

[00148] In various embodiments, agents are configured to collect/compute statistical information about connections (e.g., at the one minute level of granularity and or at any other time interval). Examples of such information include, for the time interval, the number of bytes transferred, and in which direction. Another example of information collected by an agent about a connection is the length of time between packets. For connections that span multiple time intervals (e.g., a seven minute connection), statistics may be calculated for each minute of the connection. Such statistical information (for all connections) can be reported (e.g., to a data aggregator) once a minute.

[00149] In various embodiments, agents are also configured to maintain histogram data for a given network connection, and provide the histogram data (e.g., in the Apache Avro data exchange format) under the Connection event type data. Examples of such histograms include: 1. a packet length histogram (packet_len_hist), which characterizes network packet distribution; 2. a session length histogram (session_len_hist), which characterizes a network session length; 3. a session time histogram (session_time_hist), which characterizes a network session time; and 4. a session switch time histogram (session_switch_time_hist), which characterizes network session switch time (i.e., incoming->outgoing and vice versa). For example, histogram data may include one or more of the following fields: 1. count, which provides a count of the elements in the sampling; 2. sum, which provides a sum of elements in the sampling; 3. max, which provides the highest value element in the sampling; 4. std_dev, which provides the standard deviation of elements in the sampling; and 5. buckets, which provides a discrete sample bucket distribution of sampling data (if applicable).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00611

[00150] For some protocols (e.g., HTTP), typically, a connection is opened, a string is sent, a string is received, and the connection is closed. For other protocols (e.g., NFS), both sides of the connection engage in a constant chatter. Histograms allow data platform 12 to model application behavior (e.g., using machine learning techniques), for establishing baselines, and for detecting deviations. As one example, suppose that a given HTTP server typically sends/receives 1,000 bytes (in each direction) whenever a connection is made with it. If a connection generates 500 bytes of traffic, or 2,000 bytes of traffic, such connections would be considered within the typical usage pattern of the server. Suppose, however, that a connection is made that results in 10G of traffic. Such a connection is anomalous and can be flagged accordingly.

[00151] Returning to Fig. 1D, as previously mentioned, data aggregator 114 may be configured to provide information (e.g., collected from nodes 108 by agents) to data platform 12. Data aggregator 128 may be similarly configured to provide information to data platform 12. As shown in Fig. 1D, both aggregator 114 and aggregator 128 may connect to a load balancer 130, which accepts connections from aggregators (and/or as applicable, agents), as well as other devices, such as computer 126 (e.g., when it communicates with web app 120), and supports fair balancing. In various embodiments, load balancer 130 is a reverse proxy that load balances accepted connections internally to various microservices (described in more detail below), allowing for services provided by data platform 12 to scale up as more agents are added to the environment and/or as more entities subscribe to services provided by data platform 12. Example ways to implement load balancer 130 include, but are not limited to, using HaProxy, using nginx, and using elastic load balancing (ELB) services made available by Amazon.

[00152] Agent service 132 is a microservice that is responsible for accepting data collected from agents (e.g., provided by aggregator 114). In various embodiments, agent service 132 uses a standard secure protocol, such as HTTPS to communicate with aggregators (and as applicable agents), and receives data in an appropriate format such as Apache Avro. When agent service 132 receives an incoming connection, it can perform a variety of checks, such as to see whether the data is being provided by a current customer, and whether the data is being provided in an appropriate format. If the data is not appropriately formatted (and/or is not provided by a current customer), it may be rejected.

[00153] If the data is appropriately formatted, agent service 132 may facilitate copying the received data to a streaming data stable storage using a streaming service (e.g., Amazon Kinesis and/or any other suitable streaming service. Once the ingesting into the streaming service is complete, service 132 may send an acknowledgement to the data provider (e.g., data aggregator 114). If the agent does not receive such an acknowledgement, it is configured to retry sending the data to data platform 12. One way to implement agent service 132 is as a REST API server

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00612

framework (e.g., Java DropWizard), configured to communicate with Kinesis (e.g., using a Kinesis library).

[00154] In various embodiments, data platform 12 uses one or more streams (e.g., Kinesis streams) for all incoming customer data (e.g., including data provided by data aggregator 114 and data aggregator 128), and the data is sharded based on the node (also referred to herein as a "machine") that originated the data (e.g., node 116 vs. node 122), with each node having a globally unique identifier within data platform 12. Multiple instances of agent service 132 can write to multiple shards.

[00155] Kinesis is a streaming service with a limited period (e.g., 1-7 days). To persist data longer than a day, the data may be copied to long term storage 42 (e.g., S3). Data loader 136 is a microservice that is responsible for picking up data from a data stream (e.g., a Kinesis stream) and persisting it in long term storage 42. In one example embodiment, files collected by data loader 136 from the Kinesis stream are placed into one or more buckets, and segmented using a combination of a customer identifier and time slice. Given a particular time segment, and a given customer identifier, the corresponding file (stored in long term storage) contains five minutes (or another appropriate time slice) of data collected at that specific customer from all of the customer's nodes. Data loader 136 can be implemented in any appropriate programming language, such as Java or C, and can be configured to use a Kinesis library to interface with Kinesis. In various embodiments, data loader 136 uses the Amazon Simple Queue Service (SQS) (e.g., to alert DB loader 140 that there is work for it to do).

[00156] DB loader 140 is a microservice that is responsible for loading data into an appropriate data store 30, such as SnowflakeDB or Amazon Redshift, using individual per-customer databases. In particular, DB loader 140 is configured to periodically load data into a set of raw tables from files created by data loader 136 as per above. DB loader 140 manages throughput, errors, etc., to make sure that data is loaded consistently and continuously. Further, DB loader 140 can read incoming data and load into data store 30 data that is not already present in tables of data store 30 (also referred to herein as a database). DB loader 140 can be implemented in any appropriate programming language, such as Java or C, and an SQL framework such as jOOQ (e.g., to manage SQLs for insertion of data), and SQL/JDBC libraries. In some examples, DB loader 140 may use Amazon S3 and Amazon Simple Queue Service (SQS) to manage files being transferred to and from data store 30.

[00157] Customer data included in data store 30 can be augmented with data from additional data sources, such as AWS CloudTrail and/or other types of external tracking services. To this end, data platform may include a tracking service analyzer 144, which is another microservice. Tracking service analyzer 144 may pull data from an external tracking service (e.g., Amazon

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00613

CloudTrail) for each applicable customer account, as soon as the data is available. Tracking service analyzer 144 may normalize the tracking data as applicable, so that it can be inserted into data store 30 for later querying/analysis. Tracking service analyzer 144 can be written in any appropriate programming language, such as Java or C. Tracking service analyzer 144 also makes use of SQL/JDBC libraries to interact with data store 30 to insert/query data.

[00158] As described herein, data platform 12 can model activities that occur within datacenters, such as datacenters 104 and 106. The model may be stable over time, and differences, even subtle ones (e.g., between a current state of the datacenter and the model) can be surfaced. The ability to surface such anomalies can be particularly beneficial in datacenter environments where rogue employees and/or external attackers may operate slowly (e.g., over a period of months), hoping that the elastic nature of typical resource use (e.g., virtualized servers) will help conceal their nefarious activities.

[00159] Using techniques described herein, data platform 12 can automatically discover entities (which may implement compute assets 16) deployed in a given datacenter. Examples of entities include workloads, applications, processes, machines, virtual machines, containers, files, IP addresses, domain names, and users. The entities may be grouped together logically (into analysis groups) based on behaviors, and temporal behavior baselines can be established. In particular, using techniques described herein, periodic graphs can be constructed (also referred to herein as polygraphs), in which the nodes are applicable logical entities, and the edges represent behavioral relationships between the logical entities in the graph. Baselines can be created for every node and edge.

[00160] Communication (e.g., between applications/nodes) is one example of a behavior. A model of communications between processes is an example of a behavioral model. As another example, the launching of applications is another example of a behavior that can be modeled. The baselines may be periodically updated (e.g., hourly) for every entity. Additionally or alternatively, the baselines may be continuously updated in substantially real-time as data is collected by agents. Deviations from the expected normal behavior can then be detected and automatically reported (e.g., as anomalies or threats detected). Such deviations may be due to a desired change, a misconfiguration, or malicious activity. As applicable, data platform 12 can score the detected deviations (e.g., based on severity and threat posed). Additional examples of analysis groups include models of machine communications, models of privilege changes, and models of insider behaviors (monitoring the interactive behavior of human users as they operate within the datacenter).

[00161] Two example types of information collected by agents are network level information and process level information. As previously mentioned, agents may collect information about every

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00614

connection involving their respective nodes. And, for each connection, information about both the server and the client may be collected (e.g., using the connection-to-process identification techniques described above). DNS queries and responses may also be collected. The DNS query information can be used in logical entity graphing (e.g., collapsing many different IP addresses to a single service – e.g., s3.amazon.com). Examples of process level information collected by agents include attributes (user ID, effective user ID, and command line). Information such as what user/application is responsible for launching a given process and the binary being executed (and its SHA-256 values) may also be provided by agents.

[00162] The dataset collected by agents across a datacenter can be very large, and many resources (e.g., virtual machines, IP addresses, etc.) are recycled very quickly. For example, an IP address and port number used at a first point in time by a first process on a first virtual machine may very rapidly be used (e.g., an hour later) by a different process/virtual machine.

[00163] A dataset (and elements within it) can be considered at both a physical level, and a logical level, as illustrated in Fig. 2B. In particular, Fig. 2B illustrates an example 5-tuple of data 210 collected by an agent, represented physically (216) and logically (217). The 5-tuple includes a source address 211, a source port 212, a destination address 213, a destination port 214, and a protocol 215. In some cases, port numbers (e.g., 212, 214) may be indicative of the nature of a connection (e.g., with certain port usage standardized). However, in many cases, and in particular in datacenters, port usage is ephemeral. For example, a Docker container can listen on an ephemeral port, which is unrelated to the service it will run. When another Docker container starts (for the same service), the port may well be different. Similarly, particularly in a virtualized environment, IP addresses may be recycled frequently (and are thus also potentially ephemeral) or could be NATed, which makes identification difficult.

[00164] A physical representation of the 5-tuple is depicted in region 216. A process 218 (executing on machine 219) has opened a connection to machine 220. In particular, process 218 is in communication with process 221. Information such as the number of packets exchanged between the two machines over the respective ports can be recorded.

[00165] As previously mentioned, in a datacenter environment, portions of the 5-tuple may change – potentially frequently – but still be associated with the same behavior. Namely, one application (e.g., Apache) may frequently be in communication with another application (e.g., Oracle), using ephemeral datacenter resources. Further, either/both of Apache and Oracle may be multi-homed. This can lead to potentially thousands of 5-tuples (or more) that all correspond to Apache communicating with Oracle within a datacenter. For example, Apache could be executed on a single machine, and could also be executed across fifty machines, which are variously spun up and down (with different IP addresses each time). An alternate representation of the 5-tuple of

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00615

data 210 is depicted in region 217, and is logical. The logical representation of the 5-tuple aggregates the 5-tuple (along with other connections between Apache and Oracle having other 5-tuples) as logically representing the same connection. By aggregating data from raw physical connection information into logical connection information, using techniques described herein, a size reduction of six orders of magnitude in the data set can be achieved.

[00166] Fig. 2C depicts a portion of a logical polygraph. Suppose a datacenter has seven instances of the application update_engine 225, executing as seven different processes on seven different machines, having seven different IP addresses, and using seven different ports. The instances of update_engine variously communicate with update.core-os.net 226, which may have a single IP address or many IP addresses itself, over the one hour time period represented in the polygraph. In the example shown in Fig. 2C, update_engine is a client, connecting to the server update.core-os.net, as indicated by arrow 228.

[00167] Behaviors of the seven processes are clustered together, into a single summary. As indicated in region 227, statistical information about the connections is also maintained (e.g., number of connections, histogram information, etc.). A polygraph such as is depicted in Fig. 2C can be used to establish a baseline of behavior (e.g., at the one-hour level), allowing for the future detection of deviations from that baseline. As one example, suppose that statistically an update_engine instance transmits data at 11 bytes per second. If an instance were instead to transmit data at 1000 bytes per second, such behavior would represent a deviation from the baseline and could be flagged accordingly. Similarly, changes that are within the baseline (e.g., an eighth instance of update_engine appears, but otherwise behaves as the other instances; or one of the seven instances disappears) are not flagged as anomalous. Further, datacenter events, such as failover, autobalancing, and A-B refresh are unlikely to trigger false alarms in a polygraph, as at the logical level, the behaviors remain the same.

[00168] In various embodiments, polygraph data is maintained for every application in a datacenter, and such polygraph data can be combined to make a single datacenter view across all such applications. Fig. 2D illustrates a portion of a polygraph for a service that evidences more complex behaviors than are depicted in Fig. 2C. In particular, Fig. 2D illustrates the behaviors of S3 as a service (as used by a particular customer datacenter). Clients within the datacenter variously connect to the S3 service using one of five fully qualified domains (listed in region 230). Contact with any of the domains is aggregated as contact with S3 (as indicated in region 231). Depicted in region 232 are various containers which (as clients) connect with S3. Other containers (which do not connect with S3) are not included. As with the polygraph portion depicted in Fig. 2C, statistical information about the connections is known and summarized, such as the number of bytes transferred, histogram information, etc.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00616

[00169] Fig. 2E illustrates a communication polygraph for a datacenter. In particular, the polygraph indicates a one hour summary of approximately 500 virtual machines, which collectively run one million processes, and make 100 million connections in that hour. As illustrated in Fig. 2E, a polygraph represents a drastic reduction in size (e.g., from tracking information on 100 million connections in an hour, to a few hundred nodes and a few hundred edges). Further, as a datacenter scales up (e.g., from using 10 virtual machines to 100 virtual machines as the datacenter uses more workers to support existing applications), the polygraph for the datacenter will tend to stay the same size (with the 100 virtual machines clustering into the same nodes that the 10 virtual machines previously clustered into). As new applications are added into the datacenter, the polygraph may automatically scale to include behaviors involving those applications.

[00170] In the particular polygraph shown in Fig. 2E, nodes generally correspond to workers, and edges correspond to communications the workers engage in (with connection activity being the behavior modeled in polygraph 235). Another example polygraph could model other behavior, such as application launching. The communications graphed in Fig. 2E include traffic entering the datacenter, traffic exiting the datacenter, and traffic that stays wholly within the datacenter (e.g., traffic between workers). One example of a node included in polygraph 235 is the sshd application, depicted as node 236. As indicated in Fig. 2E, 421 instances of sshd were executing during the one hour time period of data represented in polygraph 235. As indicated in region 237, nodes within the datacenter communicated with a total of 1349 IP addresses outside of the datacenter (and not otherwise accounted for, e.g., as belonging to a service such as Amazon AWS 238 or Slack 239).

[00171] In the following examples, suppose that user B, an administrator of datacenter 106, is interacting with data platform 12 to view visualizations of polygraphs in a web browser (e.g., as served to user B via web app 120). One type of polygraph user B can view is an application-communication polygraph, which indicates, for a given one hour window (or any other suitable time interval), which applications communicated with which other applications. Another type of polygraph user B can view is an application launch polygraph. User B can also view graphs related to user behavior, such as an insider behavior graph which tracks user connections (e.g., to internal and external applications, including chains of such behavior), a privilege change graph which tracks how privileges change between processes, and a user login graph, which tracks which (logical) machines a user logs into.

[00172] Fig. 2F illustrates an example of an application-communication polygraph for a datacenter (e.g., datacenter 106) for the one hour period of 9am-10am on June 5. The time slice currently being viewed is indicated in region 240. If user B clicks his mouse in region 241, user

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00617

B will be shown a representation of the application-communication polygraph as generated for the following hour (10am-11am on June 5).

[00173] Fig. 2G depicts what is shown in user B's browser after he has clicked on region 241, and has further clicked on region 242. The selection in region 242 turns on and off the ability to compare two time intervals to one another. User B can select from a variety of options when comparing the 9am-10am and 10am-11am time intervals. By clicking region 248, user B will be shown the union of both graphs (i.e., any connections that were present in either time interval). By clicking region 249, user B will be shown the intersection of both graphs (i.e., only those connections that were present in both time intervals).

[00174] As shown in Fig. 2G, user B has elected to click on region 250, which depicts connections that are only present in the 9am-10am polygraph in a first color 251, and depicts connections that are only present in the 10am-11am polygraph in a second color 252. Connections present in both polygraphs are omitted from display. As one example, in the 9am-10am polygraph (corresponding to connections made during the 9am-10am time period at datacenter 106), a connection was made by a server to sshd (253) and also to systemd (254). Both of those connections ended prior to 10am and are thus depicted in the first color. As another example, in the 10am-11am polygraph (corresponding to connections made during the 10am-11am time period at datacenter 106), a connection was made from a known bad external IP to nginx (255). The connection was not present during the 9am-10am time slice and thus is depicted in the second color. As yet another example, two different connections were made to a Slack service between 9am and 11am. However, the first was made by a first client during the 9am-10am time slice (256) and the second was made by a different client during the 10am-11am slice (257), and so the two connections are depicted respectively in the first and second colors and blue.

[00175] Returning to the polygraph depicted in Fig. 2F, suppose user B enters "etcd" into the search box located in region 244. User B will then be presented with the interface illustrated in Fig. 2H. As shown in Fig. 2H, three applications containing the term "etcd" were engaged in communications during the 9am-10am window. One application is etcdctl, a command line client for etcd. As shown in Fig. 2H, a total of three different etcdctl processes were executed during the 9am-10am window, and were clustered together (260). Fig. 2H also depicts two different clusters that are both named etcd2. The first cluster includes (for the 9am-10am window) five members (261) and the second cluster includes (for the same window) eight members (262). The reason for these two distinct clusters is that the two groups of applications behave differently (e.g., they exhibit two distinct sets of communication patterns). Specifically, the instances of etcd2 in cluster 261 only communicate with locksmithctl (263) and other etcd2 instances (in both

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00618

clusters 261 and 262). The instances of etcd2 in cluster 262 communicate with additional entities, such as etcdctl and Docker containers. As desired, user B can click on one of the clusters (e.g., cluster 261) and be presented with summary information about the applications included in the cluster, as is shown in Fig. 2I (e.g., in region 265). User B can also double click on a given cluster (e.g., cluster 261) to see details on each of the individual members of the cluster broken out.

[00176] Suppose user B now clicks on region 245 of the interface shown in Fig. 2F. User B will then be shown an application launch polygraph. Launching an application is another example of a behavior. The launch polygraph models how applications are launched by other applications. Fig. 2J illustrates an example of a portion of a launch polygraph. In particular, user B has typed "find" into region 266, to see how the "find" application is being launched. As shown in Fig. 2J, in the launch polygraph for the 10am-11am time period, find applications (267) are always launched by bash (268), which is in turn always launched by systemd (269). If find is launched by a different application, this would be anomalous behavior.

[00177] Fig. 2K illustrates another example of a portion of an application launch polygraph. In Fig. 2K, user B has searched (270) for "python ma" to see how "python marathon_lb" (271) is launched. As shown in Fig. 2K, in each case (during the one hour time slice of 10am-11am), python marathon_lb is launched as a result of a chain of the same seven applications each time. If python marathon_lb is ever launched in a different manner, this indicates anomalous behavior. The behavior could be indicative of malicious activities, but could also be due to other reasons, such as a misconfiguration, a performance-related issue, and/or a failure, etc.

[00178] Suppose user B now clicks on region 246 of the interface shown in Fig. 2F. User B will then be shown an insider behavior graph. The insider behavior graph tracks information about behaviors such as processes started by a user interactively using protocols such as ssh or telnet, and any processes started by those processes. As one example, suppose an administrator logs into a first virtual machine in datacenter 106 (e.g., using sshd via an external connection he makes from a hotel), using a first set of credentials (e.g., first.last@example.com and an appropriate password). From the first virtual machine, the administrator connects to a second virtual machine (e.g., using the same credentials), then uses the sudo command to change identities to those of another user, and then launches a program. graphs built by data platform 12 can be used to associate the administrator with each of his actions, including launching the program using the identity of another user.

[00179] Fig. 2L illustrates an example of a portion of an insider behavior graph. In particular, in Fig. 2L, user B is viewing a graph that corresponds to the time slice of 3pm-4pm on June 1. Fig. 2L illustrates the internal/external applications that users connected to during the one hour time

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00619

slice. If a user typically communicates with particular applications, that information will become part of a baseline. If the user deviates from his baseline behavior (e.g., using new applications, or changing privilege in anomalous ways), such anomalies can be surfaced.

[00180] Fig. 2M illustrates an example of a portion of a privilege change graph, which identifies how privileges are changed between processes. Typically, when a user launches a process (e.g., "ls"), the process inherits the same privileges that the user has. And, while a process can have fewer privileges than the user (i.e., go down in privilege), it is rare (and generally undesirable) for a user to escalate in privilege. Information included in the privilege change graph can be determined by examining the parent of each running process, and determining whether there is a match in privilege between the parent and the child. If the privileges are different, a privilege change has occurred (whether a change up or a change down). The application ntpd is one rare example of a scenario in which a process escalates (272) to root, and then returns back (273). The sudo command is another example (e.g., used by an administrator to temporarily have a higher privilege). As with the other examples, ntpd's privilege change actions, and the legitimate actions of various administrators (e.g., using sudo) will be incorporated into a baseline model by data platform 12. When deviations occur, such as where a new application that is not ntpd escalates privilege, or where an individual that has not previously/does not routinely use sudo does so, such behaviors can be identified as anomalous.

[00181] Fig. 2N illustrates an example of a portion of a user login graph, which identifies which users log into which logical nodes. Physical nodes (whether bare metal or virtualized) are clustered into a logical machine cluster, for example, using yet another graph, a machine-server graph, an example of which is shown in Fig. 2O. For each machine, a determination is made as to what type of machine it is, based on what kind(s) of workflows it runs. As one example, some machines run as master nodes (having a typical set of workflows they run, as master nodes) and can thus be clustered as master nodes. Worker nodes are different from master nodes, for example, because they run Docker containers, and frequently change as containers move around. Worker nodes can similarly be clustered.

[00182] As previously mentioned, the polygraph depicted in Fig. 2E corresponds to activities in a datacenter in which, in a given hour, approximately 500 virtual machines collectively run one million processes, and make 100 million connections in that hour. The polygraph represents a drastic reduction in size (e.g., from tracking information on 100 million connections in an hour, to a few hundred nodes and a few hundred edges). Using techniques described herein, such a polygraph can be constructed (e.g., using commercially available computing infrastructure) in less than an hour (e.g., within a few minutes). Thus, ongoing hourly snapshots of a datacenter can be created within a two hour moving window (i.e., collecting data for the time period 8am-9am,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00620

while also generating a snapshot for the time previous time period 7am-8am). The following describes various example infrastructure that can be used in polygraph construction, and also describes various techniques that can be used to construct polygraphs.

[00183] Returning to Fig. 1D, embodiments of data platform 12 may be built using any suitable infrastructure as a service (IaaS) (e.g., AWS). For example, data platform 12 can use Simple Storage Service (S3) for data storage, Key Management Service (KMS) for managing secrets, Simple Queue Service (SQS) for managing messaging between applications, Simple Email Service (SES) for sending emails, and Route 53 for managing DNS. Other infrastructure tools can also be used. Examples include: orchestration tools (e.g., Kubernetes or Mesos/Marathon), service discovery tools (e.g., Mesos-DNS), service load balancing tools (e.g., marathon-LB), container tools (e.g., Docker or rkt), log/metric tools (e.g., collectd, fluentd, kibana, etc.), big data processing systems (e.g., Spark, Hadoop, AWS Redshift, Snowflake etc.), and distributed key value stores (e.g., Apache Zookeeper or etcd2).

[00184] As previously mentioned, in various embodiments, data platform 12 may make use of a collection of microservices. Each microservice can have multiple instances, and may be configured to recover from failure, scale, and distribute work amongst various such instances, as applicable. For example, microservices are auto-balancing for new instances, and can distribute workload if new instances are started or existing instances are terminated. In various embodiments, microservices may be deployed as self-contained Docker containers. A Mesos-Marathon or Spark framework can be used to deploy the microservices (e.g., with Marathon monitoring and restarting failed instances of microservices as needed). The service etcd2 can be used by microservice instances to discover how many peer instances are running, and used for calculating a hash-based scheme for workload distribution. Microservices may be configured to publish various health/status metrics to either an SQS queue, or etcd2, as applicable. In some examples, Amazon DynamoDB can be used for state management.

[00185] Additional information on various microservices used in embodiments of data platform 12 is provided below.

[00186] Graph generator 146 is a microservice that may be responsible for generating raw behavior graphs on a per customer basis periodically (e.g., once an hour). In particular, graph generator 146 may generate graphs of entities (as the nodes in the graph) and activities between entities (as the edges). In various embodiments, graph generator 146 also performs other functions, such as aggregation, enrichment (e.g., geolocation and threat), reverse DNS resolution, TF-IDF based command line analysis for command type extraction, parent process tracking, etc.

[00187] Graph generator 146 may perform joins on data collected by the agents, so that both sides of a behavior are linked. For example, suppose a first process on a first virtual machine

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00621

(e.g., having a first IP address) communicates with a second process on a second virtual machine (e.g., having a second IP address). Respective agents on the first and second virtual machines may each report information on their view of the communication (e.g., the PID of their respective processes, the amount of data exchanged and in which direction, etc.). When graph generator performs a join on the data provided by both agents, the graph will include a node for each of the processes, and an edge indicating communication between them (as well as other information, such as the directionality of the communication – i.e., which process acted as the server and which as the client in the communication).

[00188] In some cases, connections are process to process (e.g., from a process on one virtual machine within the cloud environment associated with entity A to another process on a virtual machine within the cloud environment associated with entity A). In other cases, a process may be in communication with a node (e.g., outside of entity A) which does not have an agent deployed upon it. As one example, a node within entity A might be in communication with node 172, outside of entity A. In such a scenario, communications with node 172 are modeled (e.g., by graph generator 146) using the IP address of node 172. Similarly, where a node within entity A does not have an agent deployed upon it, the IP address of the node can be used by graph generator in modeling.

[00189] Graphs created by graph generator 146 may be written to data store 30 and cached for further processing. A graph may be a summary of all activity that happened in a particular time interval. As each graph corresponds to a distinct period of time, different rows can be aggregated to find summary information over a larger timestamp. In some examples, picking two different graphs from two different timestamps can be used to compare different periods. If necessary, graph generator can parallelize its workload (e.g., where its backlog cannot otherwise be handled within a particular time period, such as an hour, or if is required to process a graph spanning a long time period).

[00190] Graph generator 146 can be implemented in any appropriate programming language, such as Java or C, and machine learning libraries, such as Spark's MLLib. Example ways that graph generator computations can be implemented include using SQL or Map-R, using Spark or Hadoop.

[00191] SSH tracker 148 is a microservice that may be responsible for following ssh connections and process parent hierarchies to determine trails of user ssh activity. Identified ssh trails are placed by the SSH tracker 148 into data store 30 and cached for further processing.

[00192] SSH tracker 148 can be implemented in any appropriate programming language, such as Java or C, and machine libraries, such as Spark's MLLib. Example ways that SSH tracker computations can be implemented include using SQL or Map-R, using Spark or Hadoop.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00622

[00193] Threat aggregator 150 is a microservice that may be responsible for obtaining third party threat information from various applicable sources, and making it available to other micro-services. Examples of such information include reverse DNS information, GeoIP information, lists of known bad domains/IP addresses, lists of known bad files etc. As applicable, the threat information is normalized before insertion into data store 30. Threat aggregator 150 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 (e.g., for insertions and queries).

[00194] Scheduler 152 is a microservice that may act as a scheduler and that may run arbitrary jobs organized as a directed graph. In some examples, scheduler 152 ensures that all jobs for all customers are able to run during at a given time interval (e.g., every hour). Scheduler 152 may handle errors and retrying for failed jobs, track dependencies, manage appropriate resource levels, and/or scale jobs as needed. Scheduler 152 can be implemented in any appropriate programming language, such as Java or C. A variety of components can also be used, such as open source scheduler frameworks (e.g., Airflow), or AWS services (e.g., the AWS Data pipeline) which can be used for managing schedules.

[00195] Graph Behavior Modeler (GBM) 154 is a microservice that may compute polygraphs. In particular, GBM 154 can be used to find clusters of nodes in a graph that should be considered similar based on some set of their properties and relationships to other nodes. As described herein, the clusters and their relationships can be used to provide visibility into a datacenter environment without requiring user specified labels. GBM 154 may track such clusters over time persistently, allowing for changes to be detected and alerts to be generated.

[00196] GBM 154 may take as input a raw graph (e.g., as generated by graph generator 146). Nodes are actors of a behavior, and edges are the behavior relationship itself. For example, in the case of communication, example actors include processes, which communicate with other processes. The GBM 154 clusters the raw graph based on behaviors of actors and produces a summary (the polygraph). The polygraph summarizes behavior at a datacenter level. The GBM also produces "observations" that represent changes detected in the datacenter. Such observations may be based on differences in cumulative behavior (e.g., the baseline) of the datacenter with its current behavior. The GBM 154 can be implemented in any appropriate programming language, such as Java, C, or Golang, using appropriate libraries (as applicable) to handle distributed graph computations (handling large amounts of data analysis in a short amount of time). Apache Spark is another example tool that can be used to compute polygraphs. The GBM can also take feedback from users and adjust the model according to that feedback. For example, if a given user is interested in relearning behavior for a particular entity, the GBM can be instructed to "forget" the implicated part of the polygraph.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00623

[00197] GBM runner 156 is a microservice that may be responsible for interfacing with GBM 154 and providing GBM 154 with raw graphs (e.g., using a query language, such as SQL, to push any computations it can to data store 30). GBM runner 156 may also insert polygraph output from GBM 154 to data store 30. GBM runner 156 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 to insert and query data.

[00198] Alert generator 158 is a microservice that may be responsible for generating alerts. Alert generator 158 may examine observations (e.g., produced by GBM 154) in aggregate, deduplicate them, and score them. Alerts may be generated for observations with a score exceeding a threshold. Alert generator 158 may also compute (or retrieves, as applicable) data that a customer (e.g., user A or user B) might need when reviewing the alert. Examples of events that can be detected by data platform 12 (and alerted on by alert generator 158) include, but are not limited to the following:

[00199] new user: This event may be created the first time a user (e.g., of node 116) is first observed by an agent within a datacenter.

[00200] user launched new binary: This event may be generated when an interactive user launches an application for the first time.

[00201] new privilege escalation: This event may be generated when user privileges are escalated and a new application is run.

[00202] new application or container: This event may be generated when an application or container is seen for the first time.

[00203] new external connection: This event may be generated when a connection to an external IP/domain is made from a new application.

[00204] new external host or IP: This event may be generated when a new external host or IP is involved in a connection with a datacenter.

[00205] new internal connection: This event may be generated when a connection between internal-only applications is seen for the first time.

[00206] new external client: This event may be generated when a new external connection is seen for an application which typically does not have external connections.

[00207] new parent: This event may be generated when an application is launched by a different parent.

[00208] connection to known bad IP/domain: Data platform 12 maintains (or can otherwise access) one or more reputation feeds. If an environment makes a connection to a known bad IP or domain, an event will be generated.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00624

[00209] login from a known bad IP/domain: An event may be generated when a successful connection to a datacenter from a known bad IP is observed by data platform 12.

[00210] Alert generator 158 can be implemented in any appropriate programming language, such as Java or C, using SQL/JDBC libraries to interact with data store 30 to insert and query data. In various embodiments, alert generator 158 also uses one or more machine learning libraries, such as Spark's MLLib (e.g., to compute scoring of various observations). Alert generator 158 can also take feedback from users about which kinds of events are of interest and which to suppress.

[00211] QsJobServer 160 is a microservice that may look at all the data produced by data platform 12 for an hour, and compile a materialized view (MV) out of the data to make queries faster. The MV helps make sure that the queries customers most frequently run, and data that they search for, can be easily queried and answered. QsJobServer 160 may also precompute and cache a variety of different metrics so that they can quickly be provided as answers at query time. QsJobServer 160 can be implemented using any appropriate programming language, such as Java or C, using SQL/JDBC libraries. In some examples, QsJobServer 160 is able to compute an MV efficiently at scale, where there could be a large number of joins. An SQL engine, such as Oracle, can be used to efficiently execute the SQL, as applicable.

[00212] Alert notifier 162 is a microservice that may take alerts produced by alert generator 158 and send them to customers' integrated Security Information and Event Management (SIEM) products (e.g., Splunk, Slack, etc.). Alert notifier 162 can be implemented using any appropriate programming language, such as Java or C. Alert notifier 162 can be configured to use an email service (e.g., AWS SES or pagerduty) to send emails. Alert notifier 162 may also provide templating support (e.g., Velocity or Moustache) to manage templates and structured notifications to SIEM products.

[00213] Reporting module 164 is a microservice that may be responsible for creating reports out of customer data (e.g., daily summaries of events, etc.) and providing those reports to customers (e.g., via email). Reporting module 164 can be implemented using any appropriate programming language, such as Java or C. Reporting module 164 can be configured to use an email service (e.g., AWS SES or pagerduty) to send emails. Reporting module 164 may also provide templating support (e.g., Velocity or Moustache) to manage templates (e.g., for constructing HTML-based email).

[00214] Web app 120 is a microservice that provides a user interface to data collected and processed on data platform 12. Web app 120 may provide login, authentication, query, data visualization, etc. features. Web app 120 may, in some embodiments, include both client and server elements. Example ways the server elements can be implemented are using Java DropWizard or Node.Js to serve business logic, and a combination of JSON/HTTP to manage the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00625

service. Example ways the client elements can be implemented are using frameworks such as React, Angular, or Backbone. JSON, jQuery, and JavaScript libraries (e.g., underscore) can also be used.

[00215] Query service 166 is a microservice that may manage all database access for web app 120. Query service 166 abstracts out data obtained from data store 30 and provides a JSON-based REST API service to web app 120. Query service 166 may generate SQL queries for the REST APIs that it receives at run time. Query service 166 can be implemented using any appropriate programming language, such as Java or C and SQL/JDBC libraries, or an SQL framework such as jOOQ. Query service 166 can internally make use of a variety of types of databases, including a relational database engine 168 (e.g., AWS Aurora) and/or data store 30 to manage data for clients. Examples of tables that query service 166 manages are OLTP tables and data warehousing tables.

[00216] Cache 170 may be implemented by Redis and/or any other service that provides a key-value store. Data platform 12 can use cache 170 to keep information for frontend services about users. Examples of such information include valid tokens for a customer, valid cookies of customers, the last time a customer tried to login, etc.

[00217] Fig. 3A illustrates an example of a process for detecting anomalies in a network environment. In various embodiments, process 300 is performed by data platform 12. The process begins at 301 when data associated with activities occurring in a network environment (such as entity A's datacenter) is received. One example of such data that can be received at 301 is agent-collected data described above (e.g., in conjunction with process 200).

[00218] At 302, a logical graph model is generated, using at least a portion of the monitored activities. A variety of approaches can be used to generate such logical graph models, and a variety of logical graphs can be generated (whether using the same, or different approaches). The following is one example of how data received at 301 can be used to generate and maintain a model.

[00219] During bootstrap, data platform 12 creates an aggregate graph of physical connections (also referred to herein as an aggregated physical graph) by matching connections that occurred in the first hour into communication pairs. Clustering is then performed on the communication pairs. Examples of such clustering, described in more detail below, include performing Matching Neighbor clustering and similarity (e.g., SimRank) clustering. Additional processing can also be performed (and is described in more detail below), such as by splitting clusters based on application type, and annotating nodes with DNS query information. The resulting graph (also referred to herein as a base graph or common graph) can be used to generate a variety of models, where a subset of node and edge types (described in more detail below) and their properties are

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00626

considered in a given model. One example of a model is a UID to UID model (also referred to herein as a Uid2Uid model) which clusters together processes that share a username and show similar privilege change behavior. Another example of a model is a CType model, which clusters together processes that share command line similarity. Yet another example of a model is a PType model, which clusters together processes that share behaviors over time.

[00220] Each hour (or any other predetermined time interval) after bootstrap, a new snapshot is taken (i.e., data collected about a datacenter in the last hour is processed) and information from the new snapshot is merged with existing data to create and (as additional data is collected/processed) maintain a cumulative graph. The cumulative graph (also referred to herein as a cumulative PType graph and a polygraph) is a running model of how processes behave over time. Nodes in the cumulative graph are PType nodes, and provide information such as a list of all active processes and PIDs in the last hour, the number of historic total processes, the average number of active processes per hour, the application type of the process (e.g., the CType of the PType), and historic CType information/frequency. Edges in the cumulative graph can represent connectivity and provide information such as connectivity frequency. The edges can be weighted (e.g., based on number of connections, number of bytes exchanged, etc.). Edges in the cumulative graph (and snapshots) can also represent transitions.

[00221] One approach to merging a snapshot of the activity of the last hour into a cumulative graph is as follows. An aggregate graph of physical connections is made for the connections included in the snapshot (as was previously done for the original snapshot used during bootstrap). And, clustering/splitting is similarly performed on the snapshot's aggregate graph. Next, PType clusters in the snapshot's graph are compared against PType clusters in the cumulative graph to identify commonality.

[00222] One approach to determining commonality is, for any two nodes that are members of a given CmdType (described in more detail below), comparing internal neighbors and calculating a set membership Jaccard distance. The pairs of nodes are then ordered by decreasing similarity (i.e., with the most similar sets first). For nodes with a threshold amount of commonality (e.g., at least 66% members in common), any new nodes (i.e., appearing in the snapshot's graph but not the cumulative graph) are assigned the same PType identifier as is assigned to the corresponding node in the cumulative graph. For each node that is not classified (i.e., has not been assigned a PType identifier), a network signature is generated (i.e., indicative of the kinds of network connections the node makes, who the node communicates with, etc.). The following processing is then performed until convergence. If a match of the network signature is found in the cumulative graph, the unclassified node is assigned the PType identifier of the corresponding node in the cumulative graph. Any nodes which remain unclassified after convergence are new PTypes and

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00627

are assigned new identifiers and added to the cumulative graph as new. As applicable, the detection of a new PType can be used to generate an alert. If the new PType has a new CmdType, a severity of the alert can be increased. If any surviving nodes (i.e., present in both the cumulative graph and the snapshot graph) change PTypes, such change is noted as a transition, and an alert can be generated. Further, if a surviving node changes PType and also changes CmdType, a severity of the alert can be increased.

[00223] Changes to the cumulative graph (e.g., a new PType or a new edge between two PTypes) can be used (e.g., at 303) to detect anomalies (described in more detail below). Two example kinds of anomalies that can be detected by data platform 12 include security anomalies (e.g., a user or process behaving in an unexpected manner) and devops/root cause anomalies (e.g., network congestion, application failure, etc.). Detected anomalies can be recorded and surfaced (e.g., to administrators, auditors, etc.), such as through alerts which are generated at 304 based on anomaly detection.

[00224] Additional detail regarding processing performed, by various components depicted in Fig. 1D (whether performed individually or in combination), in conjunction with model/polygraph construction (e.g., as performed at 302) are provided below.

[00225] As explained above, an aggregated physical graph can be generated on a per customer basis periodically (e.g., once an hour) from raw physical graph information, by matching connections (e.g., between two processes on two virtual machines). In various embodiments, a deterministic fixed approach is used to cluster nodes in the aggregated physical graph (e.g., representing processes and their communications). As one example, Matching Neighbors Clustering (MNC) can be performed on the aggregated physical graph to determine which entities exhibit identical behavior and cluster such entities together.

[00226] Fig. 3B depicts a set of example processes (p1, p2, p3, and p4) communicating with other processes (p10 and p11). Fig. 3B is a graphical representation of a small portion of an aggregated physical graph showing (for a given time period, such as an hour) which processes in a datacenter communicate with which other processes. Using MNC, processes p1, p2, and p3 will be clustered together (305), as they exhibit identical behavior (they communicate with p10 and only p10). Process p4, which communicates with both p10 and p11, will be clustered separately.

[00227] In MNC, only those processes exhibiting identical (communication) behavior will be clustered. In various embodiments, an alternate clustering approach can also/instead be used, which uses a similarity measure (e.g., constrained by a threshold value, such as a 60% similarity) to cluster items. In some embodiments, the output of MNC is used as input to SimRank, in other embodiments, MNC is omitted.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00628

[00228] Fig. 3C depicts a set of example processes (p4, p5, p6) communicating with other processes (p7, p8, p9). As illustrated, most of nodes p4, p5, and p6 communicate with most of nodes p7, p8, and p9 (as indicated in Fig. 3C with solid connection lines). As one example, process p4 communicates with process p7 (310), process p8 (311), and process p9 (312). An exception is process p6, which communicates with processes p7 and p8, but does not communicate with process p9 (as indicated by dashed line 313). If MNC were applied to the nodes depicted in Fig. 3C, nodes p4 and p5 would be clustered (and node p6 would not be included in their cluster).

[00229] One approach to similarity clustering is to use SimRank. In an embodiment of the SimRank approach, for a given node v in a directed graph, $I(v)$ and $O(v)$ denote the respective set of in-neighbors and out-neighbors of v. Individual in-neighbors are denoted as $I_i(v)$, for $1 \leq i \leq |I(v)|$, and individual out-neighbors are denoted as $O_i(v)$, for $1 \leq i \leq |O(v)|$. The similarity between two objects a and b can be denoted by $s(a,b) \in [1,0]$. A recursive equation (hereinafter "the SimRank equation") can be written for $s(a,b)$, where, if a=b, then $s(a,b)$ is defined as 1, otherwise, $s(a,b)=\frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$ where C is a constant between 0 and 1. One example value for the decay factor C is 0.8 (and a fixed number of iterations such as five). Another example value for the decay factor C is 0.6 (and/or a different number of iterations). In the event that a or b has no in-neighbors, similarity is set to $s(a,b)=0$, so the summation is defined to be 0 when $I(a)=\emptyset$ or $I(b)=\emptyset$.

[00230] The SimRank equations for a graph G can be solved by iteration to a fixed point. Suppose n is the number of nodes in G. For each iteration k, n^2 entries $s_k(*,*)$ are kept, where $s_k(a,b)$ gives the score between a and b on iteration k. Successive computations of $s_{k+1}(*,*)$ are made based on $s_k(*,*)$. Starting with $s_0(*,*)$, where each $s_0(a,b)$ is a lower bound on the actual SimRank score $s(a,b)$: $s_0(a,b) = \begin{cases} 1, \text{if } a = b, \\ 0, \text{if } a \neq b. \end{cases}$

[00231] The SimRank equation can be used to compute $s_{k+1}(a,b)$ from $s_k(*,*)$ with $s_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|} s_k(I_i(a), I_j(b))$ for $a \neq b$, and $s_{k+1}(a,b)=1$ for $a=b$. On each iteration k+1, the similarity of (a,b) is updated using the similarity scores of the neighbors of (a,b) from the previous iteration k according to the SimRank equation. The values $s_k(*,*)$ are nondecreasing as k increases.

[00232] Returning to Fig. 3C, while MNC would cluster nodes p4 and p5 together (and not include node p6 in their cluster), application of SimRank would cluster nodes p4-p6 into one cluster (314) and also cluster nodes p7-p9 into another cluster (315).

[00233] Fig. 3D depicts a set of processes, and in particular server processes s1 and s2, and client processes c1, c2, c3, c4, c5, and c6. Suppose only nodes s1, s2, c1, and c2 are present in the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00629

graph depicted in Fig. 3D (and the other nodes depicted are omitted from consideration). Using MNC, nodes s1 and s2 would be clustered together, as would nodes c1 and c2. Performing SimRank clustering as described above would also result in those two clusters (s1 and s2, and c1 and c2). As previously mentioned, in MNC, identical behavior is required. Thus, if node c3 were now also present in the graph, MNC would not include c3 in a cluster with c2 and c1 because node c3 only communicates with node s2 and not node s1. In contrast, a SimRank clustering of a graph that includes nodes s1, s2, c1, c2, and c3 would result (based, e.g., on an applicable selected decay value and number of iterations) in a first cluster comprising nodes s1 and s2, and a second cluster of c1, c2, and c3. As an increasing number of nodes which communicate with server process s2, and do not also communicate with server process s1, are included in the graph (e.g., as c4, c5, and c6 are added), under SimRank, nodes s1 and s2 will become decreasingly similar (i.e., their intersection is reduced).

[00234] In various embodiments, SimRank is modified (from what is described above) to accommodate differences between the asymmetry of client and server connections. As one example, SimRank can be modified to use different thresholds for client communications (e.g., an 80% match among nodes c1-c6) and for server communications (e.g., a 60% match among nodes s1 and s2). Such modification can also help achieve convergence in situations such as where a server process dies on one node and restarts on another node.

[00235] The application of MNC/SimRank to an aggregated physical graph results in a smaller graph, in which processes which are determined to be sufficiently similar are clustered together. Typically, clusters generated as output of MNC will be underinclusive. For example, for the nodes depicted in Fig. 3C, process p6 will not be included in a cluster with processes p4 and p5, despite substantial similarity in their communication behaviors. The application of SimRank (e.g., to the output of MNC) helps mitigate the underinclusiveness of MNC, but can result in overly inclusive clusters. As one example, suppose (returning to the nodes depicted in Fig. 3B) that as a result of applying SimRank to the depicted nodes, nodes p1-p4 are all included in a single cluster. Both MNC and SimRank operate agnostically of which application a given process belongs to. Suppose processes p1-p3 each correspond to a first application (e.g., an update engine), and process p4 corresponds to a second application (e.g., sshd). Further suppose process p10 corresponds to contact with AWS. Clustering all four of the processes together (e.g., as a result of SimRank) could be problematic, particularly in a security context (e.g., where granular information useful in detecting threats would be lost).

[00236] As previously mentioned, data platform 12 may maintain a mapping between processes and the applications to which they belong. In various embodiments, the output of SimRank (e.g., SimRank clusters) is split based on the applications to which cluster members belong (such a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00630

split is also referred to herein as a "CmdType split"). If all cluster members share a common application, the cluster remains. If different cluster members originate from different applications, the cluster members are split along application-type (CmdType) lines. Using the nodes depicted in Fig. 3D as an example, suppose that nodes c1, c2, c3, and c5 all share "update engine" as the type of application to which they belong (sharing a CmdType). Suppose that node c4 belongs to "ssh," and suppose that node c6 belongs to "bash." As a result of SimRank, all six nodes (c1-c6) might be clustered into a single cluster. After a CmdType split is performed on the cluster, however, the single cluster will be broken into three clusters (c1, c2, c3, c5; c4; and c6). Specifically, the resulting clusters comprise processes associated with the same type of application, which exhibit similar behaviors (e.g., communication behaviors). Each of the three clusters resulting from the CmdType split represents, respectively, a node (also referred to herein as a PType) of a particular CmdType. Each PType is given a persistent identifier and stored persistently as a cumulative graph.

[00237] A variety of approaches can be used to determine a CmdType for a given process. As one example, for some applications (e.g., sshd), a one-to-one mapping exists between the CmdType and the application/binary name. Thus, processes corresponding to the execution of sshd will be classified using a CmdType of sshd. In various embodiments, a list of common application/binary names (e.g., sshd, apache, etc.) is maintained by data platform 12 and manually curated as applicable. Other types of applications (e.g., Java, Python, and Ruby) are multi-homed, meaning that several very different applications may all execute using the binary name, "java." For these types of applications, information such as command line / execution path information can be used in determining a CmdType. In particular, the subapplication can be used as the CmdType of the application, and/or term frequency analysis (e.g., TF/IDF) can be used on command line information to group, for example, any marathon related applications together (e.g., as a python.marathon CmdType) and separately from other Python applications (e.g., as a python.airflow CmdType).

[00238] In various embodiments, machine learning techniques are used to determine a CmdType. The CmdType model is constrained such that the execution path for each CmdType is unique. One example approach to making a CmdType model is a random forest based approach. An initial CmdType model is bootstrapped using process parameters (e.g., available within one minute of process startup) obtained using one hour of information for a given customer (e.g., entity A). Examples of such parameters include the command line of the process, the command line of the process's parent(s) (if applicable), the uptime of the process, UID/EUID and any change information, TTY and any change information, listening ports, and children (if any).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00631

Another approach is to perform term frequency clustering over command line information to convert command lines into cluster identifiers.

[00239] The random forest model can be used (e.g., in subsequent hours) to predict a CmdType for a process (e.g., based on features of the process). If a match is found, the process can be assigned the matching CmdType. If a match is not found, a comparison between features of the process and its nearest CmdType (e.g., as determined using a Levenstein distance) can be performed. The existing CmdType can be expanded to include the process, or, as applicable, a new CmdType can be created (and other actions taken, such as generating an alert). Another approach to handling processes which do not match an existing CmdType is to designate such processes as unclassified, and once an hour, create a new random forest seeded with process information from a sampling of classified processes (e.g., 10 or 100 processes per CmdType) and the new processes. If a given new process winds up in an existing set, the process is given the corresponding CmdType. If a new cluster is created, a new CmdType can be created.

[00240] Conceptually, a polygraph represents the smallest possible graph of clusters that preserve a set of rules (e.g., in which nodes included in the cluster must share a CmdType and behavior). As a result of performing MNC, SimRank, and cluster splitting (e.g., CmdType splitting) many processes are clustered together based on commonality of behavior (e.g., communication behavior) and commonality of application type. Such clustering represents a significant reduction in graph size (e.g., compared to the original raw physical graph). Nonetheless, further clustering can be performed (e.g., by iterating on the graph data using the GBM to achieve such a polygraph). As more information within the graph is correlated, more nodes can be clustered together, reducing the size of the graph, until convergence is reached and no further clustering is possible.

[00241] Fig. 3E depicts two pairs of clusters. In particular, cluster 320 represents a set of client processes sharing the same CmdType ("a1"), communicating (collectively) with a server process having a CmdType ("a2"). Cluster 322 also represents a set of client processes having a CmdType a1 communicating with a server process having a CmdType a2. The nodes in clusters 320 and 322 (and similarly nodes in 321 and 323) remain separately clustered (as depicted) after MNC/SimRank/CmdType splitting – isolated islands. One reason this could occur is where server process 321 corresponds to processes executing on a first machine (having an IP address of 1.1.1.1). The machine fails and a new server process 323 starts, on a second machine (having an IP address of 2.2.2.2) and takes over for process 321.

[00242] Communications between a cluster of nodes (e.g., nodes of cluster 320) and the first IP address can be considered different behavior from communications between the same set of nodes and the second IP address, and thus communications 324 and 325 will not be combined by

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00632

MNC/SimRank in various embodiments. Nonetheless, it could be desirable for nodes of clusters 320/322 to be combined (into cluster 326), and for nodes of clusters 321/323 to be combined (into cluster 327), as representing (collectively) communications between a1 and a2. One task that can be performed by data platform 12 is to use DNS query information to map IP addresses to logical entities. As will be described in more detail below, GBM 154 can make use of the DNS query information to determine that graph nodes of cluster 320 and graph nodes of cluster 322 both made DNS queries for "appserverabc.example.com," which first resolved to 1.1.1.1 and then to 2.2.2.2, and to combine nodes 320/322 and 321/323 together into a single pair of nodes (326 communicating with 327).

[00243] In various embodiments, GBM 154 operates in a batch manner in which it receives as input the nodes and edges of a graph for a particular time period along with its previous state, and generates as output clustered nodes, cluster membership edges, cluster-to-cluster edges, events, and its next state.

[00244] GBM 154 may not try to consider all types of entities and their relationships that may be available in a conceptual common graph all at once. Instead, GBM uses a concept of models where a subset of node and edge types and their properties are considered in a given model. Such an approach is helpful for scalability, and also to help preserve detailed information (of particular importance in a security context) – as clustering entities in a more complex and larger graph could result in less useful results. In particular, such an approach allows for different types of relationships between entities to be preserved/more easily analyzed.

[00245] While GBM 154 can be used with different models corresponding to different subgraphs, core abstractions remain the same across types of models.

[00246] For example, each node type in a GBM model is considered to belong to a class. The class can be thought of as a way for the GBM to split nodes based on the criteria it uses for the model. The class for a node is represented as a string whose value is derived from the node's key and properties depending on the GBM Model. Note that different GBM models may create different class values for the same node. For each node type in a given GBM model, GBM 154 can generate clusters of nodes for that type. A GBM generated cluster for a given member node type cannot span more than one class for that node type. GBM 154 generates edges between clusters that have the same types as the edges between source and destination cluster node types.

[00247] Additionally or alternatively, the processes described herein as being used for a particular model can be used (can be the same) across models, and different models can also be configured with different settings.

[00248] Additionally or alternatively, the node types and the edge types may correspond to existing types in the common graph node and edge tables but this is not necessary. Even when

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00633

there is a correspondence, the properties provided to GBM 154 are not limited to the properties that are stored in the corresponding graph table entries. They can be enriched with additional information before being passed to GBM 154.

[00249] Logically, the input for a GBM model can be characterized in a manner that is similar to other graphs. Edge triplets can be expressed, for example, as an array of source node type, edge type, and destination node type. And, each node type is associated with node properties, and each edge type is associated with edge properties. Other edge triplets can also be used (and/or edge triplets can be extended) in accordance with various embodiments.

[00250] Note that the physical input to the GBM model need not (and does not, in various embodiments) conform to the logical input. For example, the edges in the PtypeConn model correspond to edges between Matching Neighbors (MN) clusters, where each process node has an MN cluster identifier property. In the User ID to User ID model (also referred to herein as the Uid2Uid model), edges are not explicitly provided separately from nodes (as the euid array in the node properties serves the same purpose). In both cases, however, the physical information provides the applicable information necessary for the logical input.

[00251] The state input for a particular GBM model can be stored in a file, a database, or other appropriate storage. The state file (from a previous run) is provided, along with graph data, except for when the first run for a given model is performed, or the model is reset. In some cases, no data may be available for a particular model in a given time period, and GBM may not be run for that time period. As data becomes available at a future time, GBM can run using the latest state file as input.

[00252] GBM 154 outputs cluster nodes, cluster membership edges, and inter-cluster relationship edges that are stored (in some embodiments) in the graph node tables: node_c, node_cm, and node_icr, respectively. The type names of nodes and edges may conform to the following rules:

[00253] * A given node type can be used in multiple different GBM models. The type names of the cluster nodes generated by two such models for that node type will be different. For instance, process type nodes will appear in both PtypeConn and Uid2Uid models, but their cluster nodes will have different type names.

[00254] * The membership edge type name is "MemberOf."

[00255] * The edge type names for cluster-to-cluster edges will be the same as the edge type names in the underlying node-to-node edges in the input.

[00256] The following are example events GBM 154 can generate: new class, new cluster, new edge from class to class, split class (the notion that GBM 154 considers all nodes of a given type and class to be in the same cluster initially and if GBM 154 splits them into multiple clusters, it is

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00634

splitting a class), new edge from cluster and class, new edge between cluster and cluster, and/or new edge from class to cluster.

[00257] One underlying node or edge in the logical input can cause multiple types of events to be generated. Conversely, one event can correspond to multiple nodes or edges in the input. Not every model generates every event type.

[00258] Additional information regarding examples of data structures/models that can be used in conjunction with models used by data platform 12 is now provided.

[00259] In some examples, a PTypeConn Model clusters nodes of the same class that have similar connectivity relationships. For example, if two processes had similar incoming neighbors of the same class and outgoing neighbors of the same class, they could be clustered.

[00260] The node input to the PTypeConn model for a given time period includes non-interactive (i.e., not associated with tty) process nodes that had connections in the time period and the base graph nodes of other types (IP Service Endpoint (IPSep) comprising an IP address and a port), DNS Service Endpoint (DNSSep) and IPAddress) that have been involved in those connections. The base relationship is the connectivity relationship for the following type triplets:

[00261] * Process, ConnectedTo, Process

[00262] * Process, ConnectedTo, IP Service Endpoint (IPSep)

[00263] * Process, ConnectedTo, DNS Service Endpoint (DNSSep)

[00264] * IPAddress, ConnectedTo, ProcessProcess, DNS, ConnectedTo, Process

[00265] The edge inputs to this model are the ConnectedTo edges from the MN cluster, instead of individual node-to-node ConnectedTo edges from the base graph. The membership edges created by this model refer to the base graph node type provided in the input.

[00266] Class Values:

[00267] The class values of nodes are determined as follows depending on the node type (e.g., Process nodes, IPSep nodes, DNSSep nodes, and IP Address nodes).

[00268] Process nodes:

[00269] if exe_path contains java then "java <cmdline_term_1> ..."

[00270] else if exe_path contains python then "python <cmdline_term_1> ..."

[00271] else "last_part_of_exe_path"

[00272] IPSep nodes:

[00273] if IP_internal then "IntIPS"

[00274] else if severity = 0 then "<IP_addr>:<protocol>:<port>"

[00275] else "<IP_addr>:<port>_BadIP"

[00276] DNSSep nodes:

[00277] if IP_internal = 1 then "<hostname>"

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00635

[00278] else if severity = 0 then "<hostname>:<protocol>:port"

[00279] else "<hostname>:<port>_BadIP"

[00280] IPAddress nodes (will appear only on client side):

[00281] if IP_internal = 1 then "IPIntC"

[00282] else if severity = 0 then "ExtIPC"

[00283] else "ExtBadIPC"

[00284] Events:

[00285] A new class event in this model for a process node is equivalent to seeing a new CType being involved in a connection for the first time. Note that this does not mean the CType was not seen before. It is possible that it was previously seen but did not make a connection at that time.

[00286] A new class event in this model for an IPSep node with IP_internal = 0 is equivalent to seeing a connection to a new external IP address for the first time.

[00287] A new class event in this model for a DNSSep node is equivalent to seeing a connection to a new domain for the first time.

[00288] A new class event in this model for an IPAddress node with IP_internal = 0 and severity = 0 is equivalent to seeing a connection from any external IP address for the first time.

[00289] A new class event in this model for an IPAddress node with IP_internal = 0 and severity > 0 is equivalent to seeing a connection from any bad external IP address for the first time.

[00290] A new class to class to edge from a class for a process node to a class for a process node is equivalent to seeing a communication from the source CType making a connection to the destination CType for the first time.

[00291] A new class to class to edge from a class for a process node to a class for a DNSSep node is equivalent to seeing a communication from the source CType making a connection to the destination domain name for the first time.

[00292] An IntPConn Model may be similar to the PtypeConn Model, except that connection edges between parent/child processes and connections between processes where both sides are not interactive are filtered out.

[00293] A Uid2Uid Model may cluster processes with the same username that show similar privilege change behavior. For instance, if two processes with the same username had similar effective user values, launched processes with similar usernames, and were launched by processes with similar usernames, then they could be clustered.

[00294] An edge between a source cluster and destination cluster generated by this model means that all of the processes in the source cluster had a privilege change relationship to at least one process in the destination cluster.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00636

[00295] The node input to this model for a given time period includes process nodes that are running in that period. The value of a class of process nodes is "<username>".

[00296] The base relationship that is used for clustering is privilege change, either by the process changing its effective user ID, or by launching a child process which runs with a different user.

[00297] The physical input for this model includes process nodes (only), with the caveat that the complete ancestor hierarchy of process nodes active (i.e., running) for a given time period is provided as input even if an ancestor is not active in that time period. Note that effective user IDs of a process are represented as an array in the process node properties, and launch relationships are available from ppid_hash fields in the properties as well.

[00298] A new class event in this model is equivalent to seeing a user for the first time.

[00299] A new class to class edge event is equivalent to seeing the source user making a privilege change to the destination user for the first time.

[00300] A Ct2Ct Model may cluster processes with the same CType that show similar launch behavior. For instance, if two processes with the same CType have launched processes with similar CTypes, then they could be clustered.

[00301] The node input to this model for a given time period includes process nodes that are running in that period. The value class of process nodes is CType (similar to how it is created for the PtypeConn Model).

[00302] The base relationship that is used for clustering is a parent process with a given CType launching a child process with another given destination CType.

[00303] The physical input for this model includes process nodes (only) with the caveat that the complete ancestor hierarchy active process nodes (i.e., that are running) for a given time period is provided as input even if an ancestor is not active in that time period. Note that launch relationships are available from ppid_hash fields in the process node properties.

[00304] An edge between a source cluster and destination cluster generated by this model means that all of the processes in the source cluster launched at least one process in the destination cluster.

[00305] A new class event in this model is equivalent to seeing a CType for the first time. Note that the same type of event will be generated by the PtypeConn Model as well.

[00306] A new class to class edge event is equivalent to seeing the source CType launching the destination CType for the first time.

[00307] An MTypeConn Model may cluster nodes of the same class that have similar connectivity relationships. For example, if two machines had similar incoming neighbors of the same class and outgoing neighbors of the same class, they could be clustered.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00637

[00308] A new class event in this model will be generated for external IP addresses or (as applicable) domain names seen for the first time. Note that a new class to class to edge Machine, class to class for an IPSep or DNSName node will also be generated at the same time.

[00309] The membership edges generated by this model will refer to Machine, IPAddress, DNSName, and IPSep nodes in the base graph. Though the nodes provided to this model are IPAddress nodes instead of IPSep nodes, the membership edges it generates will refer to IPSep type nodes. Alternatively, the base graph can generate edges between Machine and IPSep node types. Note that the Machine to IPAddress edges have tcp_dst_ports/udp_dst_ports properties that can be used for this purpose.

[00310] The node input to this model for a given time period includes machine nodes that had connections in the time period and the base graph nodes of other types (IPAddress and DNSName) that were involved in those connections.

[00311] The base relationship is the connectivity relationship for the following type triplets:

[00312] * Machine, ConnectedTo, Machine

[00313] * Machine, ConnectedTo, IPAddress

[00314] * Machine, ConnectedTo, DNSName

[00315] * IPAddress, ConnectedTo, Machine, DNS, ConnectedTo, Machine

[00316] The edge inputs to this model are the corresponding ConnectedTo edges in the base graph.

[00317] Class Values:

[00318] Machine:

[00319] The class value for all Machine nodes is "Machine."

[00320] The machine_terms property in the Machine nodes is used, in various embodiments, for labeling machines that are clustered together. If a majority of the machines clustered together share a term in the machine_terms, that term can be used for labeling the cluster.

[00321] IPSep:

[00322] The class value for IPSep nodes is determined as follows:

[00323] if IP_internal then "IntIPS"

[00324] else

[00325] if severity = 0 then "<ip_addr>:<protocol>:<port>"

[00326] else "<IP_addr_BadIP>"

[00327] IPAddress:

[00328] The class value for IpAddress nodes is determined as follows:

[00329] if IP_internal then "IntIPC"

[00330] else

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00638

[00331] if severity = 0 then “ExtIPC”

[00332] else “ExtBadIPC”

[00333] DNSName:

[00334] The class value for DNSName nodes is determined as follows:

[00335] if severity = 0 then “<hostname>”

[00336] else then “<hostname>_BadIP”

[00337] An example structure for a New Class Event is now described.

[00338] The key field for this event type looks as follows (using the PtypeConn model as an example):

[00339] {

[00340] "node": {

[00341] "class": {

[00342] "cid": "httpd"

[00343] },

[00344] "key": {

[00345] "cid": "29654"

[00346] },

[00347] "type": "PtypeConn"

[00348] }

[00349] }

[00350] It contains the class value and also the ID of the cluster where that class value is observed. Multiple clusters can be observed with the same value in a given time period. It contains the class value and also the ID of the cluster where that class value is observed. Multiple clusters can be observed with the same value in a given time period. Accordingly, in some embodiments, GBM 154 generates multiple events of this type for the same class value.

[00351] The properties field looks as follows:

[00352] {

[00353] "set_size": 5

[00354] }

[00355] The set_size indicates the size of the cluster referenced in the keys field.

[00356] Conditions:

[00357] For a given model and time period, multiple NewClass events can be generated if there is more than one cluster in that class. NewNode events will not be generated separately in this case.

[00358] Example New Class to Class Edge Event structure:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00639

[00359] The key field for this event type looks as follows (using the PtypeConn model as an example):

[00360] "edge": {

[00361] "dst_node": {

[00362] "class": {

[00363] "cid": "java war"

[00364] },

[00365] "key": {

[00366] "cid": "27635"

[00367] },

[00368] "type": "PtypeConn"

[00369] },

[00370] "src_node": {

[00371] "class": {

[00372] "cid": "IntIPC"

[00373] },

[00374] "key": {

[00375] "cid": "20881"

[00376] },

[00377] "type": "PtypeConn"

[00378] },

[00379] "type": "ConnectedTo"

[00380] }

[00381] }

[00382] The key field contains source and destination class values and also source and destination cluster identifiers (i.e., the src/dst_node:key.cid represents the src/dst cluster identifier).

[00383] In a given time period for a given model, an event of this type could involve multiple edges between different cluster pairs that have the same source and destination class values. GBM 154 can generate multiple events in this case with different source and destination cluster identifiers.

[00384] The props fields look as follows for this event type:

[00385] {

[00386] "dst_set_size": 2,

[00387] "src_set_size": 1

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00640

[00388] }

[00389] The source and destination sizes represent the sizes of the clusters given in the keys field.

[00390] Conditions:

[00391] For a given model and time period, multiple NewClassToClass events can be generated if there are more than one pair of clusters in that class pair. NewNodeToNode events are not generated separately in this case.

[00392] Combining Events at the Class Level: for a given model and time period, the following example types of events can represent multiple changes in the underlying GBM cluster level graph in terms of multiple new clusters or multiple new edges between clusters:

[00393] * NewClass

[00394] * NewEdgeClassToClass

[00395] * NewEdgeNodeToClass

[00396] * NewEdgeClassToNode

[00397] Multiple NewClass events with the same model and class can be output if there are multiple clusters in that new class.

[00398] Multiple NewEdgeClassToClass events with the same model and class pair can be output if there are multiple new cluster edges within that class pair.

[00399] Multiple NewEdgeNodeToClass events with the same model and destination class can be output if there are multiple new edges from the source cluster to the destination clusters in that destination class (the first time seeing this class as a destination cluster class for the source cluster).

[00400] Multiple NewEdgeClassToNode events with the same model and source class can be output if there are multiple new edges from source clusters to the destination clusters in that source class (the first time seeing this class as a source cluster class for the destination cluster).

[00401] These events may be combined at the class level and treated as a single event when it is desirable to view changes at the class level, e.g., when one wants to know when there is a new CType.

[00402] In some examples, different models may have partial overlap in the types of nodes they use from the base graph. Therefore, they can generate NewClass type events for the same class. NewClass events can also be combined across models when it is desirable to view changes at the class level.

[00403] Using techniques herein, actions can be associated with processes and (e.g., by associating processes with users) actions can thus also be associated with extended user sessions. Such information can be used to track user behavior correctly, even where a malicious user

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00641

attempts to hide his trail by changing user identities (e.g., through lateral movement). Extended user session tracking can also be useful in operational use cases without malicious intent, e.g., where users make original logins with distinct usernames (e.g., "charlie" or "dave") but then perform actions under a common username (e.g., "admin" or "support"). One such example is where multiple users with administrator privileges exist, and they need to gain superuser privilege to perform a particular type of maintenance. It may be desirable to know which operations are performed (as the superuser) by which original user when debugging issues. In the following examples describing extended user session tracking, reference is generally made to using the secure shell (ssh) protocol as implemented by openssh (on the server side) as the mechanism for logins. However, extended user session tracking is not limited to the ssh protocol or a particular limitation and the techniques described herein can be extended to other login mechanisms.

[00404] On any given machine, there will be a process that listens for and accepts ssh connections on a given port. This process can run the openssh server program running in daemon mode or it could be running another program (e.g., initd on a Linux system). In either case, a new process running openssh will be created for every new ssh login session and this process can be used to identify an ssh session on that machine. This process is called the "privileged" process in openssh.

[00405] After authentication of the ssh session, when an ssh client requests a shell or any other program to be run under that ssh session, a new process that runs that program will be created under (i.e., as a child of) the associated privileged process. If an ssh client requests port forwarding to be performed, the connections will be associated with the privileged process.

[00406] In modern operating systems such as Linux and Windows, each process has a parent process (except for the very first process) and when a new process is created the parent process is known. By tracking the parent-child hierarchy of processes, one can determine if a particular process is a descendant of a privileged openssh process and thus if it is associated with an ssh login session.

[00407] For user session tracking across machines (or on a single machine with multiple logins) in a distributed environment, it is established when two login sessions have a parent-child relationship. After that, the "original" login session, if any, for any given login session can be determined by following the parent relationship recursively.

[00408] Fig. 3F is a representation of a user logging into a first machine and then into a second machine from the first machine, as well as information associated with such actions. In the example of Fig. 3F, a user, Charlie, logs into Machine A (331) from a first IP address (332). As part of the login process, he provides a username (333). Once connected to Machine A, an

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00642

openssh privileged process (334) is created to handle the connection for the user, and a terminal session is created and a bash process (335) is created as a child. Charlie launches an ssh client (336) from the shell, and uses it to connect (337) to Machine B (338). As with the connection he makes to Machine A, Charlie's connection to Machine B will have an associated incoming IP address (339), in this case, the IP address of Machine A. And, as part of the login process with Machine B, Charlie will provide a username (340) which need not be the same as username 333. An openssh privileged process (341) is created to handle the connection, and a terminal session and child bash process (342) will be created. From the command line of Machine B, Charlie launches a curl command (343), which opens an HTTP connection (2028) to an external Machine C (345).

[00409] Fig. 3G is an alternate representation of actions occurring in Fig. 3F, where events occurring on Machine A are indicated along line 350, and events occurring on Machine B are indicated along line 351. As shown in Fig. 3G, an incoming ssh connection is received at Machine A (352). Charlie logs in (as user "x") and an ssh privileged process is created to handle Charlie's connection (353). A terminal session is created and a bash process is created (354) as a child of process 353. Charlie wants to ssh to Machine B, and so executes an ssh client on Machine A (355), providing credentials (as user "y") at 356. Charlie logs into Machine B, and an ssh privileged process is created to handle Charlie's connection (357). A terminal session is created and a bash process is created (358) as a child of process 357. Charlie then executes curl (359) to download content from an external domain (via connection 360).

[00410] The external domain could be a malicious domain, or it could be benign. Suppose the external domain is malicious (and, e.g., Charlie has malicious intent). It would be advantageous (e.g., for security reasons) to be able to trace the contact with the external domain back to Machine A, and then back to Charlie's IP address. Using techniques described herein (e.g., by correlating process information collected by various agents), such tracking of Charlie's activities back to his original login (330) can be accomplished. In particular, an extended user session can be tracked that associates Charlie's ssh processes together with a single original login and thus original user.

[00411] As described herein, software agents (such as agent 112) may run on machines (such as a machine that implements one of nodes 116) and detect new connections, processes, and/or logins. As also previously explained, such agents send associated records to data platform 12 which includes one or more datastores (e.g., data store 30) for persistently storing such data. Such data can be modeled using logical tables, also persisted in datastores (e.g., in a relational database that provides an SQL interface), allowing for querying of the data. Other datastores such as graph oriented databases and/or hybrid schemes can also be used.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00643

[00412] The following identifiers are commonly used in the tables:

[00413] * MID

[00414] * PID_hash

[00415] An ssh login session can be identified uniquely by an (MID, PID_hash) tuple. The MID is a machine identifier that is unique to each machine, whether physical or virtual, across time and space. Operating systems use numbers called process identifiers (PIDs) to identify processes running at a given time. Over time processes may die and new processes may be started on a machine or the machine itself may restart. The PID is not necessarily unique across time in that the same PID value can be reused for different processes at different times. In order to track process descendants across time, one should therefore account for time as well. In order to be able to identify a process on a machine uniquely across time, another number called a PID_hash is generated for the process. In various embodiments, the PID_hash is generated using a collision-resistant hash function that takes the PID, start time, and (in various embodiments, as applicable) other properties of a process.

[00416] Input data collected by agents comprises the input data model and is represented by the following logical tables:

[00417] * connections

[00418] * processes

[00419] * logins

[00420] A connections table may maintain records of TCP/IP connections observed on each machine. Example columns included in a connections table are as follows:

Column Name	Description
MID	Identifier of the machine that the connection was observed on.
start_time	Connection start time.
PID_hash	Identifier of the process that was associated with the connection.
src_IP_addr	Source IP address (the connection was initiated from this IP address).
src_port	Source port.
dst_IP_addr	Destination IP address (the connection was made to this IP address).
dst_port	Destination port.
Prot	Protocol (TCP or UDP).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00644

Dir	Direction of the connection (incoming or outgoing) with respect to this machine.

[00421] The source fields (IP address and port) correspond to the side from which the connection was initiated. On the destination side, the agent associates an ssh connection with the privileged ssh process that is created for that connection.

[00422] For each connection in the system, there will be two records in the table, assuming that the machines on both sides of the connection capture the connection. These records can be matched based on equality of the tuple (src_IP_addr, src_port, dst_IP_addr, dst_port, Prot) and proximity of the start_time fields (e.g., with a one minute upper threshold between the start_time fields).

[00423] A processes table maintains records of processes observed on each machine. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the process was observed on.
PID_hash	Identifier of the process.
start_time	Start time of the process.
exe_path	The executable path of the process.
PPID_hash	Identifier of the parent process.

[00424] A logins table may maintain records of logins to machines. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with login.
login_time	Time of login.
login_username	Username used in login.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00645

[00425] Output data generated by session tracking is represented with the following logical tables:

[00426] * login-local-descendant

[00427] * login-connection

[00428] * login-lineage

[00429] Using data in these tables, it is possible to determine descendant processes of a given ssh login session across the environment (i.e., spanning machines). Conversely, given a process, it is possible to determine if it is an ssh login descendant as well as the original ssh login session for it if so.

[00430] A login-local-descendant table maintains the local (i.e., on the same machine) descendant processes of each ssh login session. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with login.
login_time	Time of login.
login_username	Username used in login.

[00431] A login-connections table may maintain the connections associated with ssh logins. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the process was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with the login.
login_time	Time of login.
login_username	The username used in the login.
src_IP_addr	Source IP address (connection was initiated from this IP address).
src_port	Source port.
dst_IP_addr	Destination IP address (connection was made to this IP address).

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00646

dst_port	Destination port.

[00432] A login-lineage table may maintain the lineage of ssh login sessions. It may have the following columns:

Column Name	Description
MID	Identifier of the machine that the ssh login was observed on.
sshd_PID_hash	Identifier of the sshd privileged process associated with the login.
parent_MID	Identifier of the machine that the parent ssh login was observed on.
parent_sshd_PID_hash	Identifier of the sshd privileged process associated with the parent login.
origin_MID	Identifier of the machine that the origin ssh login was observed on.
origin_sshd_PID_hash	Identifier of the sshd privileged process associated with the origin login.

[00433] The parent_MID and parent_sshd_PID_hash columns can be null if there is no parent ssh login. In that case, the (MID, sshd_PID_hash) tuple will be the same as the (origin_MID, origin_sshd_PID_hash) tuple.

[00434] Fig. 3H illustrates an example of a process for performing extended user tracking. In various embodiments, process 361 is performed by data platform 12. The process begins at 362 when data associated with activities occurring in a network environment (such as entity A's datacenter) is received. One example of such data that can be received at 362 is agent-collected data described above (e.g., in conjunction with process 200). At 363, the received network activity is used to identify user login activity. And, at 364, a logical graph that links the user login activity to at least one user and at least one process is generated (or updated, as applicable). Additional detail regarding process 361, and in particular, portions 363 and 364 of process 361 are described in more detail below (e.g., in conjunction with discussion of Fig. 3J).

[00435] Fig. 3I depicts a representation of a user logging into a first machine, then into a second machine from the first machine, and then making an external connection. The scenario depicted in Fig. 3I is used to describe an example of processing that can be performed on data collected by agents to generate extended user session tracking information. Fig. 3I is an alternate depiction of the information shown in Figs. 3F and 3G.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00647

[00436] At time t1 (365), a first ssh connection is made to Machine A (366) from an external source (367) by a user having a username of "X." In the following example, suppose the external source has an IP address of 1.1.1.10 and uses source port 10000 to connect to Machine A (which has an IP address of 2.2.2.20 and a destination port 22). External source 367 is considered an external source because its IP address is outside of the environment being monitored (e.g., is a node outside of entity A's datacenter, connecting to a node inside of entity A's datacenter).

[00437] A first ssh login session LS1 is created on machine A for user X. The privileged openssh process for this login is A1 (368). Under the login session LS1, the user creates a bash shell process with PID_hash A2 (369).

[00438] At time t2 (370), inside the bash shell process A2, the user runs an ssh program under a new process A3 (371) to log in to machine B (372) with a different username ("Y"). In particular, an ssh connection is made from source IP address 2.2.2.20 and source port 10001 (Machine A's source information) to destination IP address 2.2.2.21 and destination port 22 (Machine B's destination information).

[00439] A second ssh login session LS2 is created on machine B for user Y. The privileged openssh process for this login is B1 (373). Under the login session LS2, the user creates a bash shell process with PID_hash B2 (374).

[00440] At time t3 (376), inside the bash shell process B2, the user runs a curl command under a new process B3 (377) to download a file from an external destination (378). In particular, an HTTPS connection is made from source IP address 2.2.2.21 and source port 10002 (Machine B's source information) to external destination IP address 3.3.3.30 and destination port 443 (the external destination's information).

[00441] Using techniques described herein, it is possible to determine the original user who initiated the connection to external destination 378, which in this example is a user having the username X on machine A (where the extended user session can be determined to start with ssh login session LS1).

[00442] Based on local descendant tracking, the following determinations can be on machine A and B without yet having performed additional processing (described in more detail below):

[00443] * A3 is a descendant of A1 and thus associated with LS1.

[00444] * The connection to the external domain from machine B is initiated by B3.

[00445] * B3 is a descendant of B1 and is thus associated with LS2.

[00446] * Connection to the external domain is thus associated with LS2.

[00447] An association between A3 and LS2 can be established based on the fact that LS2 was created based on an ssh connection initiated from A3. Accordingly, it can be determined that LS2 is a child of LS1.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00648

[00448] To determine the user responsible for making the connection to the external destination (e.g., if it were a known bad destination), first, the process that made the connection would be traced, i.e., from B3 to LS2. Then LS2 would be traced to LS1 (i.e., LS1 is the origin login session for LS2). Thus the user for this connection is the user for LS1, i.e., X. As represented in Fig. 3I, one can visualize the tracing by following the links (in the reverse direction of arrows) from external destination 378 to A1 (368).

[00449] In the example scenario, it is assumed that both ssh connections occur in the same analysis period. However, the approaches described herein will also work for connections and processes that are created in different time periods.

[00450] Fig. 3J illustrates an example of a process for performing extended user tracking. In various embodiments, process 380 is performed periodically (e.g., once an hour in a batch fashion) by ssh tracker 148 to generate new output data. In general, batch processing allows for efficient analysis of large volumes of data. However, the approach can be adapted, as applicable, to process input data on a record-by-record fashion while maintaining the same logical data processing flow. As applicable the results of a given portion of process 380 are stored for use in a subsequent portion.

[00451] The process begins at 381 when new ssh connection records are identified. In particular, new ssh connections started during the current time period are identified by querying the connections table. The query uses filters on the start_time and dst_port columns. The values of the range filter on the start_time column are based on the current time period. The dst_port column is checked against ssh listening port(s). By default, the ssh listening port number is 22. However, as this could vary across environments, the port(s) that openssh servers are listening to in the environment can be determined by data collection agents dynamically and used as the filter value for the dst_port as applicable. In the scenario depicted in Fig. 3I, the query result will generate the records shown in Fig. 3K. Note that for the connection between machine A and B, the two machines are likely to report start_time values that are not exactly the same but close enough to be considered matching (e.g., within one minute or another appropriate amount of time). In the above table, they are shown to be the same for simplicity.

[00452] At 382, ssh connection records reported from source and destination sides of the same connection are matched. The ssh connection records (e.g., returned from the query at 381) are matched based on the following criteria:

[00453] * The five tuples (src_IP, dst_IP, IP_prot, src_port, dst_port) of the connection records must match.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00649

[00454] * The delta between the start times of the connections must be within a limit that would account for the worst case clock difference expected between two machines in the environment and typical connection setup latency.

[00455] * If there are multiple matches possible, then the match with the smallest time delta is chosen.

[00456] Note that record 390 from machine A for the incoming connection from the external source cannot be matched with another record as there is an agent only on the destination side for this connection. Example output of portion 382 of process 380 is shown in Fig. 3L. The values in the dst_PID_hash column (391) are that of the sshd privileged process associated with ssh logins.

[00457] At 383, new logins during the current time period are identified by querying the logins table. The query uses a range filter on the login_time column with values based on the current time period. In the example depicted in Fig. 3I, the query result will generate the records depicted in Fig. 3M.

[00458] At 384, matched ssh connection records created at 382 and new login records created at 383 are joined to create new records that will eventually be stored in the login-connection table. The join condition is that dst_MID of the matched connection record is equal to the MID of the login record and the dst_PID_hash of the matched connection record is equal to the sshd_PID_hash of the login record. In the example depicted in Fig. 3I, the processing performed at 384 will generate the records depicted in Fig. 3N.

[00459] At 385, login-local-descendant records in the lookback time period are identified. It is possible that a process that is created in a previous time period makes an ssh connection in the current analysis batch period. Although not depicted in the example illustrated in Fig. 3I, consider a case where bash process A2 does not create ssh process A3 right away but instead that the ssh connection A3 later makes to machine B is processed in a subsequent time period than the one where A2 was processed. While processing this subsequent time period in which processes A3 and B1 are seen, knowledge of A2 would be useful in establishing that B1 is associated with A3 (via ssh connection) which is associated with A2 (via process parentage) which in turn would be useful in establishing that the parent of the second ssh login is the first ssh login. The time period for which look back is performed can be limited to reduce the amount of historical data that is considered. However, this is not a requirement (and the amount of look back can be determined, e.g., based on available processing resources). The login local descendants in the lookback time period can be identified by querying the login-local-descendant table. The query uses a range filter on the login_time column where the range is from start_time_of_current_period - lookback_time to start_time_of_current_period. (No records as a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00650

result of performing 385 on the scenario depicted in Fig. 3I are obtained, as only a single time period is applicable in the example scenario.)

[00460] At 386, new processes that are started in the current time period are identified by querying the processes table. The query uses a range filter on the start_time column with values based on the current time period. In the example depicted in Fig. 3I, the processing performed at 386 will generate the records depicted in Fig. 3O.

[00461] At 387, new login-local-descendant records are identified. The purpose is to determine whether any of the new processes in the current time period are descendants of an ssh login process and if so to create records that will be stored in the login-local-descendant table for them. In order to do so, the parent-child relationships between the processes are recursively followed. Either a top down or bottom up approach can be used. In a top down approach, the ssh local descendants in the lookback period identified at 385, along with new ssh login processes in the current period identified at 384 are considered as possible ancestors for the new processes in the current period identified at 386.

[00462] Conceptually, the recursive approach can be considered to include multiple sub-steps where new processes that are identified to be ssh local descendants in the current sub-step are considered as ancestors for the next step. In the example scenario depicted in Fig. 3I, the following descendancy relationships will be established in two sub-steps:

[00463] Sub-step 1:

[00464] Process A2 is a local descendant of LS1 (i.e., MID = A, sshd_PID_hash = A1) because it is a child of process A1 which is the login process for LS1.

[00465] Process B2 is a local descendant of LS2 (i.e., MID = B, sshd_PID_hash = B1) because it is a child of process B1 which is the login process for LS2.

[00466] Sub-step 2:

[00467] Process A3 is a local descendant of LS1 because it is a child of process A2 which is associated to LS1 in sub-step 1.

[00468] Process B3 is a local descendant of LS2 because it is a child of process B1 which is associated to LS2 in sub-step 1.

[00469] Implementation portion 387 can use a datastore that supports recursive query capabilities, or, queries can be constructed to process multiple conceptual sub-steps at once. In the example depicted in Fig. 3I, the processing performed at 387 will generate the records depicted in Fig. 3P. Note that the ssh privileged processes associated with the logins are also included as they are part of the login session.

[00470] At 388, the lineage of new ssh logins created in the current time period is determined by associating their ssh connections to source processes that may be descendants of other ssh logins

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00651

(which may have been created in the current period or previous time periods). In order to do so, first an attempt is made to join the new ssh login connections in the current period (identified at 384) with the combination of the login local descendants in the lookback period (identified at 385) and the login local descendants in the current time period (identified at 386). This will create adjacency relationships between child and parent logins. In the example depicted in Fig. 3I, the second ssh login connection will be associated with process A3 and an adjacency relationship between the two login sessions will be created (as illustrated in Fig. 3Q).

[00471] Next, the adjacency relationships are used to find the original login sessions. While not shown in the sample scenario, there could be multiple ssh logins in a chain in the current time period, in which case a recursive approach (as in 387) could be used. At the conclusion of portion 388, the login lineage records depicted in Fig. 3R will be generated.

[00472] Finally, at 389, output data is generated. In particular, the new login-connection, login-local-descendant, and login-lineage records generated at 384, 387, and 388 are inserted into their respective output tables (e.g., in a transaction manner).

[00473] An alternate approach to matching TCP connections between machines running an agent is for the client to generate a connection GUID and send it in the connection request (e.g., the SYN packet) it sends and for the server to extract the GUID from the request. If two connection records from two machines have the same GUID, they are for the same connection. Both the client and server will store the GUID (if it exists) in the connection records they maintain and report. On the client side, the agent can configure the network stack (e.g. using IP tables functionality on Linux) to intercept an outgoing TCP SYN packet and modify it to add the generated GUID as a TCP option. On the server side, the agent already extracts TCP SYN packets and thus can look for this option and extract the GUID if it exists.

[00474] Example graph-based user tracking and threat detection embodiments associated with data platform 12 will now be described. Administrators and other users of network environments (e.g., entity A's datacenter 104) often change roles to perform tasks. As one example, suppose that at the start of a workday, an administrator (hereinafter "Joe Smith") logs in to a console, using an individualized account (e.g., username=joe.smith). Joe performs various tasks as himself (e.g., answering emails, generating status reports, writing code, etc.). For other tasks (e.g., performing updates), Joe may require different/additional permission than his individual account has (e.g., root privileges). One way Joe can gain access to such permissions is by using sudo, which will allow Joe to run a single command with root privileges. Another way Joe can gain access to such permissions is by su or otherwise logging into a shell as root. After gaining root privileges, another thing that Joe can do is switch identities. As one example, to perform administrative tasks, Joe may use "su help" or "su database-admin" to become (respectively) the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00652

help user or the database-admin user on a system. He may also connect from one machine to another, potentially changing identities along the way (e.g., logging in as joe.smith at a first console, and connecting to a database server as database-admin). When he's completed various administrative tasks, Joe can relinquish his root privileges by closing out of any additional shells created, reverting back to a shell created for user joe.smith.

[00475] While there are many legitimate reasons for Joe to change his identity throughout the day, such changes may also correspond to nefarious activity. Joe himself may be nefarious, or Joe's account (joe.smith) may have been compromised by a third party (whether an "outsider" outside of entity A's network, or an "insider"). Using techniques described herein, the behavior of users of the environment can be tracked (including across multiple accounts and/or multiple machines) and modeled (e.g., using various graphs described herein). Such models can be used to generate alerts (e.g., to anomalous user behavior). Such models can also be used forensically, e.g., helping an investigator visualize various aspects of a network and activities that have occurred, and to attribute particular types of actions (e.g., network connections or file accesses) to specific users.

[00476] In a typical day in a datacenter, a user (e.g., Joe Smith) will log in, run various processes, and (optionally) log out. The user will typically log in from the same set of IP addresses, from IP addresses within the same geographical area (e.g., city or country), or from historically known IP addresses/geographical areas (i.e., ones the user has previously/occasionally used). A deviation from the user's typical (or historical) behavior indicates a change in login behavior. However, it does not necessarily mean that a breach has occurred. Once logged into a datacenter, a user may take a variety of actions. As a first example, a user might execute a binary/script. Such binary/script might communicate with other nodes in the datacenter, or outside of the datacenter, and transfer data to the user (e.g., executing "curl" to obtain data from a service external to the datacenter). As a second example, the user can similarly transfer data (e.g., out of the datacenter), such as by using POST. As a third example, a user might change privilege (one or more times), at which point the user can send/receive data as per above. As a fourth example, a user might connect to a different machine within the datacenter (one or more times), at which point the user can send/receive data as per the above.

[00477] In various embodiments, the above information associated with user behavior is broken into four tiers. The tiers represent example types of information that data platform 12 can use in modeling user behavior:

[00478] 1. The user's entry point (e.g., domains, IP addresses, and/or geolocation information such as country/city) from which a user logs in.

[00479] 2. The login user and machine class.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00653

[00480] 3. Binaries, executables, processes, etc. a user launches.

[00481] 4. Internal servers with which the user (or any of the user's processes, child processes, etc.) communicates, and external contacts (e.g., domains, IP addresses, and/or geolocation information such as country/city) with which the user communicates (i.e., transfers data).

[00482] In the event of a security breach, being able to concretely answer questions about such information can be very important. And, collectively, such information is useful in providing an end-to-end path (e.g., for performing investigations).

[00483] In the following example, suppose a user ("UserA") logs into a machine ("Machine01") from a first IP address ("IP01"). Machine01 is inside a datacenter. UserA then launches a script ("runnable.sh") on Machine01. From Machine01, UserA next logs into a second machine ("Machine02") via ssh, also as UserA, also within the datacenter. On Machine02, UserA again launches a script ("new_runnable.sh"). On Machine02, UserA then changes privilege, becoming root on Machine02. From Machine02, UserA (now as root) logs into a third machine ("Machine03") in the datacenter via ssh, as root on Machine03. As root on Machine03, the user executes a script ("collect_data.sh") on Machine03. The script internally communicates (as root) to a MySQL-based service internal to the datacenter, and downloads data from the MySQL-based service. Finally, as root on Machine03, the user externally communicates with a server outside the datacenter ("External01"), using a POST command. To summarize what has occurred, in this example, the source/entry point is IP01. Data is transferred to an external server External01. The machine performing the transfer to External01 is Machine03. The user transferring the data is "root" (on Machine03), while the actual user (hiding behind root) is UserA.

[00484] In the above scenario, the "original user" (ultimately responsible for transmitting data to External01) is UserA, who logged in from IP01. Each of the processes ultimately started by UserA, whether started at the command line (tty) such as "runnable.sh" or started after an ssh connection such as "new_runnable.sh," and whether as UserA, or as a subsequent identity, are all examples of child processes which can be arranged into a process hierarchy.

[00485] As previously mentioned, machines can be clustered together logically into machine clusters. One approach to clustering is to classify machines based on information such as the types of services they provide/binaries they have installed upon them/processes they execute. Machines sharing a given machine class (as they share common binaries/services/etc.) will behave similarly to one another. Each machine in a datacenter can be assigned to a machine cluster, and each machine cluster can be assigned an identifier (also referred to herein as a machine class). One or more tags can also be assigned to a given machine class (e.g., database_servers_west or prod_web_frontend). One approach to assigning a tag to a machine class is to apply term frequency analysis (e.g., TF/IDF) to the applications run by a given

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00654

machine class, selecting as tags those most unique to the class. Data platform 12 can use behavioral baselines taken for a class of machines to identify deviations from the baseline (e.g., by a particular machine in the class).

[00486] Fig. 3S illustrates an example of a process for detecting anomalies. In various embodiments, process 392 is performed by data platform 12. As explained above, a given session will have an original user. And, each action taken by the original user can be tied back to the original user, despite privilege changes and/or lateral movement throughout a datacenter. Process 392 begins at 393 when log data associated with a user session (and thus an original user) is received. At 394, a logical graph is generated, using at least a portion of the collected data. When an anomaly is detected (395), it can be recorded, and as applicable, an alert is generated (396). The following are examples of graphs that can be generated (e.g., at 394), with corresponding examples of anomalies that can be detected (e.g., at 395) and alerted upon (e.g., at 396).

[00487] Fig. 4A illustrates a representation of an embodiment of an insider behavior graph. In the example of Fig. 4A, each node in the graph can be: (1) a cluster of users; (2) a cluster of launched processes; (3) a cluster of processes/servers running on a machine class; (4) a cluster of external IP addresses (of incoming clients); or (5) a cluster of external servers based on DNS/IP/etc. As depicted in Fig. 4A, graph data is vertically tiered into four tiers. Tier 0 (400) corresponds to entry point information (e.g., domains, IP addresses, and/or geolocation information) associated with a client entering the datacenter from an external entry point. Entry points are clustered together based on such information. Tier 1 (401) corresponds to a user on a machine class, with a given user on a given machine class represented as a node. Tier 2 (402) corresponds to launched processes, child processes, and/or interactive processes. Processes for a given user and having similar connectivity (e.g., sharing the processes they launch and the machines with which they communicate) are grouped into nodes. Finally, Tier 3 (403) corresponds to the services/servers/domains/IP addresses with which processes communicate. A relationship between the tiers can be stated as follows: Tier 0 nodes log in to tier 1 nodes. Tier 1 nodes launch tier 2 nodes. Tier 2 nodes connect to tier 3 nodes.

[00488] The inclusion of an original user in both Tier 1 and Tier 2 allows for horizontal tiering. Such horizontal tiering ensures that there is no overlap between any two users in Tier 1 and Tier 2. Such lack of overlap provides for faster searching of an end-to-end path (e.g., one starting with a Tier 0 node and terminating at a Tier 3 node). Horizontal tiering also helps in establishing baseline insider behavior. For example, by building an hourly insider behavior graph, new edges/changes in edges between nodes in Tier 1 and Tier 2 can be identified. Any such changes correspond to a change associated with the original user. And, any such changes can be surfaced as anomalous and alerts can be generated.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00655

[00489] As explained above, Tier 1 corresponds to a user (e.g., user "U") logging into a machine having a particular machine class (e.g., machine class "M"). Tier 2 is a cluster of processes having command line similarity (e.g., CType "C"), having an original user "U," and running as a particular effective user (e.g., user "U1"). The value of U1 may be the same as U (e.g., joe.smith in both cases), or the value of U1 may be different (e.g., U=joe.smith and U1=root). Thus, while an edge may be present from a Tier 1 node to a Tier 2 node, the effective user in the Tier 2 node may or may not match the original user (while the original user in the Tier 2 node will match the original user in the Tier 1 node).

[00490] A change from a user U into a user U1 can take place in a variety of ways. Examples include where U becomes U1 on the same machine (e.g., via su), and also where U sshes to other machine(s). In both situations, U can perform multiple changes, and can combine approaches. For example, U can become U1 on a first machine, ssh to a second machine (as U1), become U2 on the second machine, and ssh to a third machine (whether as user U2 or user U3). In various embodiments, the complexity of how user U ultimately becomes U3 (or U5, etc.) is hidden from a viewer of an insider behavior graph, and only an original user (e.g., U) and the effective user of a given node (e.g., U5) are depicted. As applicable (e.g., if desired by a viewer of the insider behavior graph), additional detail about the path (e.g., an end-to-end path of edges from user U to user U5) can be surfaced (e.g., via user interactions with nodes).

[00491] Fig. 4B illustrates an example of a portion of an insider behavior graph (e.g., as rendered in a web browser). In the example shown, node 405 (the external IP address, 52.32.40.231) is an example of a Tier 0 node, and represents an entry point into a datacenter. As indicated by directional arrows 406 and 407, two users, "aruneli_prod" and "harish_prod," both made use of the source IP 52.32.40.231 when logging in between 5pm and 6pm on Sunday July 30 (408). Nodes 409 and 410 are examples of Tier 1 nodes, having aruneli_prod and harish_prod as associated respective original users. As previously mentioned, Tier 1 nodes correspond to a combination of a user and a machine class. In the example depicted in Fig. 4B, the machine class associated with nodes 409 and 410 is hidden from view to simplify visualization, but can be surfaced to a viewer of interface 404 (e.g., when the user clicks on node 409 or 410).

[00492] Nodes 414-423 are examples of Tier 2 nodes – processes that are launched by users in Tier 1 and their child, grandchild, etc. processes. Note that also depicted in Fig. 4B is a Tier 1 node 411 that corresponds to a user, "root," that logged in to a machine cluster from within the datacenter (i.e., has an entry point within the datacenter). Nodes 425-1 and 425-2 are examples of Tier 3 nodes – internal/external IP addresses, servers, etc., with which Tier 2 nodes communicate.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00656

[00493] In the example shown in Fig. 4B, a viewer of interface 404 has clicked on node 423. As indicated in region 426, the user running the marathon container is "root." However, by following the directional arrows in the graph backwards from node 423 (i.e. from right to left), the viewer can determine that the original user, responsible for node 423, is "aruneli_prod," who logged into the datacenter from IP 52.32.40.231.

[00494] The following are examples of changes that can be tracked using an insider behavior graph model:

[00495] * A user logs in from a new IP address.

[00496] * A user logs in from a geolocation not previously used by that user.

[00497] * A user logs into a new machine class.

[00498] * A user launches a process not previously used by that user.

[00499] * A user connects to an internal server to which the user has not previously connected.

[00500] * An original user communicates with an external server (or external server known to be malicious) with which that user has not previously communicated.

[00501] * A user communicates with an external server which has a geolocation not previously used by that user.

[00502] Such changes can be surfaced as alerts, e.g., to help an administrator determine when/what anomalous behavior occurs within a datacenter. Further, the behavior graph model can be used (e.g., during forensic analysis) to answer questions helpful during an investigation. Examples of such questions include:

[00503] * Was there any new login activity (Tier 0) in the timeframe being investigated? As one example, has a user logged in from an IP address with unknown geolocation information? Similarly, has a user started communicating externally with a new Tier 3 node (e.g., one with unknown geolocation information).

[00504] * Has there been any suspicious login activity (Tier 0) in the timeframe being investigated? As one example, has a user logged in from an IP address that corresponds to a known bad IP address as maintained by Threat aggregator 150? Similarly, has there been any suspicious Tier 3 activity?

[00505] * Were any anomalous connections made within the datacenter during the timeframe being investigated? As one example, suppose a given user ("Frank") typically enters a datacenter from a particular IP address (or range of IP addresses), and then connects to a first machine type (e.g., bastion), and then to a second machine type (e.g., database_prod). If Frank has directly connected to database_prod (instead of first going through bastion) during the timeframe, this can be surfaced using the insider graph.

[00506] * Who is (the original user) responsible for running a particular process?

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00657

[00507] An example of an insider behavior graph being used in an investigation is depicted in Figs. 4C and 4D. Fig. 4C depicts a baseline of behavior for a user, "Bill." As shown in Fig. 4C, Bill typically logs into a datacenter from the IP address, 71.198.44.40 (427). He typically makes use of ssh (428), and sudo (429), makes use of a set of typical applications (430) and connects (as root) with the external service, api.lacework.net (431).

[00508] Suppose Bill's credentials are compromised by a nefarious outsider ("Eve"). Fig. 4D depicts an embodiment of how the graph depicted in Fig. 4C would appear once Eve begins exfiltrating data from the datacenter. Eve logs into the datacenter (using Bill's credentials) from 52.5.66.8 (432). As Bill, Eve escalates her privilege to root (e.g., via su), and then becomes a different user, Alex (e.g., via su alex). As Alex, Eve executes a script, "sneak.sh" (433), which launches another script, "post.sh" (434), which contacts external server 435 which has an IP address of 52.5.66.7, and transmits data to it. Edges 436-439 each represent changes in Bill's behavior. As previously mentioned, such changes can be detected as anomalies and associated alerts can be generated. As a first example, Bill logging in from an IP address he has not previously logged in from (436) can generate an alert. As a second example, while Bill does typically make use of sudo (429), he has not previously executed sneak.sh (433) or post.sh (434) and the execution of those scripts can generate alerts as well. As a third example, Bill has not previously communicated with server 435, and an alert can be generated when he does so (439). Considered individually, each of edges 436-439 may indicate nefarious behavior, or may be benign. As an example of a benign edge, suppose Bill begins working from a home office two days a week. The first time he logs in from his home office (i.e., from an IP address that is not 71.198.44.40), an alert can be generated that he has logged in from a new location. Over time, however, as Bill continues to log in from his home office but otherwise engages in typical activities, Bill's graph will evolve to include logins from both 71.198.44.40 and his home office as baseline behavior. Similarly, if Bill begins using a new tool in his job, an alert can be generated the first time he executes the tool, but over time will become part of his baseline.

[00509] In some cases, a single edge can indicate a serious threat. For example, if server 432 (or 435) is included in a known bad IP listing, edge 436 (or 439) indicates compromise. An alert that includes an appropriate severity level (e.g., "threat level high") can be generated. In other cases, a combination of edges could indicate a threat (where a single edge might otherwise result in a lesser warning). In the example shown in Fig. 4D, the presence of multiple new edges is indicative of a serious threat. Of note, even though "sneak.sh" and "post.sh" were executed by Alex, because data platform 12 also keeps track of an original user, the compromise of user B's account will be discovered.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00658

[00510] Fig. 4E illustrates a representation of an embodiment of a user login graph. In the example of Fig. 4E, tier 0 (440) clusters source IP addresses as belonging to a particular country (including an "unknown" country) or as a known bad IP. Tier 1 (441) clusters user logins, and tier 2 (442) clusters type of machine class into which a user is logging in. The user login graph tracks the typical login behavior of users. By interacting with a representation of the graph, answers to questions such as the following can be obtained:

[00511] * Where is a user logging in from?

[00512] * Have any users logged in from a known bad address?

[00513] * Have any non-developer users accessed development machines?

[00514] * Which machines does a particular user access?

[00515] Examples of alerts that can be generated using the user login graph include:

[00516] * A user logs in from a known bad IP address.

[00517] * A user logs in from a new country for the first time.

[00518] * A new user logs into the datacenter for the first time.

[00519] * A user accesses a machine class that the user has not previously accessed.

[00520] One way to track privilege changes in a datacenter is by monitoring a process hierarchy of processes. To help filter out noisy commands/processes such as "su –u," the hierarchy of processes can be constrained to those associated with network activity. In a *nix system, each process has two identifiers assigned to it, a process identifier (PID) and a parent process identifier (PPID). When such a system starts, the initial process is assigned a PID 0. Each user process has a corresponding parent process.

[00521] Using techniques described herein, a graph can be constructed (also referred to herein as a privilege change graph) which models privilege changes. In particular, a graph can be constructed which identifies where a process P1 launches a process P2, where P1 and P2 each have an associated user U1 and U2, with U1 being an original user, and U2 being an effective user. In the graph, each node is a cluster of processes (sharing a CType) executed by a particular (original) user. As all the processes in the cluster belong to the same user, a label that can be used for the cluster is the user's username. An edge in the graph, from a first node to a second node, indicates that a user of the first node changed its privilege to the user of the second node.

[00522] Fig. 4F illustrates an example of a privilege change graph. In the example shown in Fig. 4F, each node (e.g., nodes 444 and 445) represents a user. Privilege changes are indicated by edges, such as edge 446.

[00523] As with other graphs, anomalies in graph 443 can be used to generate alerts. Three examples of such alerts are as follows:

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00659

[00524] * New user entering the datacenter. Any time a new user enters the datacenter and runs a process, the graph will show a new node, with a new CType. This indicates a new user has been detected within the datacenter. Fig. 4F is a representation of an example of an interface that depicts such an alert. Specifically, as indicated in region 447, an alert for the time period 1pm-2pm on June 8 was generated. The alert identifies that a new user, Bill (448) executed a process.

[00525] *Privilege change. As explained above, a new edge, from a first node (user A) to a second node (user B) indicates that user A has changed privilege to user B.

[00526] * Privilege escalation. Privilege escalation is a particular case of privilege change, in which the first user becomes root.

[00527] An example of an anomalous privilege change and an example of an anomalous privilege escalation are each depicted in graph 450 of Fig. 4G. In particular, as indicated in region 451, two alerts for the time period 2pm-3pm on June 8 were generated (corresponding to the detection of the two anomalous events). In region 452, root has changed privilege to the user "daemon," which root has not previously done. This anomaly is indicated to the user by highlighting the daemon node (e.g., outlining it in a particular color, e.g., red). As indicated by edge 453, Bill has escalated his privilege to the user root (which can similarly be highlighted in region 454). This action by Bill represents a privilege escalation.

[00528] An Extensible query interface for dynamic data compositions and filter applications will now be described.

[00529] As described herein, datacenters are highly dynamic environments. And, different customers of data platform 12 (e.g., entity A vs. entity B) may have different/disparate needs/requirements of data platform 12, e.g., due to having different types of assets, different applications, etc. Further, as time progresses, new software tools will be developed, new types of anomalous behavior will be possible (and should be detectable), etc. In various embodiments, data platform 12 makes use of predefined relational schema (including by having different predefined relational schema for different customers). However, the complexity and cost of maintaining/updating such predefined relational schema can rapidly become problematic – particularly where the schema includes a mix of relational, nested, and hierarchical (graph) datasets. In other embodiments, the data models and filtering applications used by data platform 12 are extensible. As will be described in more detail below, in various embodiments, data platform 12 supports dynamic query generation by automatic discovery of join relations via static or dynamic filtering key specifications among composable data sets. This allows a user of data platform 12 to be agnostic to modifications made to existing data sets as well as creation of new data sets. The extensible query interface also provides a declarative and configurable specification for optimizing internal data generation and derivations.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00660

[00530] As will also be described in more detail below, data platform 12 is configured to dynamically translate user interactions (e.g., received via web app 120) into SQL queries (and without the user needing to know how to write queries). Such queries can then be performed (e.g., by query service 166) against any compatible backend (e.g., data store 30).

[00531] Fig. 4H illustrates an example of a user interacting with a portion of an interface. When a user visits data platform 12 (e.g., via web app 120 using a browser), data is extracted from data store 30 as needed (e.g., by query service 166), to provide the user with information, such as the visualizations depicted variously herein). As the user continues to interact with such visualizations (e.g., clicking on graph nodes, entering text into search boxes, navigating between tabs (e.g., tab 455 vs. 465)), such interactions act as triggers that cause query service 166 to continue to obtain information from data store 30 as needed (and as described in more detail below).

[00532] In the example shown in Fig. 4H, user A is viewing a dashboard that provides various information about entity A users (455), during the time period March 2 at midnight – March 25 at 7pm (which she selected by interacting with region 456). Various statistical information is presented to user A in region 457. Region 458 presents a timeline of events that occurred during the selected time period. User A has opted to list only the critical, high, and medium events during the time period by clicking on the associated boxes (459-461). A total of 55 low severity, and 155 info-only events also occurred during the time period. Each time user A interacts with an element in Fig. 4H (e.g., clicks on box 461, clicks on link 464-1, or clicks on tab 465), her actions are translated/formalized into filters on the data set and used to dynamically generate SQL queries. The SQL queries are generated transparently to user A (and also to a designer of the user interface shown in Fig. 4H).

[00533] User A notes in the timeline (462) that a user, Harish, connected to a known bad server (examplebad.com) using wget, an event that has a critical severity level. User A can click on region 463 to expand details about the event inline (which will display, for example, the text "External connection made to known bad host examplebad.com at port 80 from application 'wget' running on host dev1.lacework.internal as user harish") directly below line 462. User A can also click on link 464-1, which will take her to a dossier for the event (depicted in Fig. 4I). As will be described in more detail below, a dossier is a template for a collection of visualizations.

[00534] As shown in interface 466, the event of Harish using wget to contact examplebad.com on March 16 was assigned an event ID of 9291 by data platform 12 (467). For convenience to user A, the event is also added to her dashboard in region 476 as a bookmark (468). A summary of the event is depicted in region 469. By interacting with boxes shown in region 470, user A can

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00661

see a timeline of related events. In this case, user A has indicated that she would like to see other events involving the wget application (by clicking box 471). Events of critical and medium security involving wget occurred during the one hour window selected in region 472.

[00535] Region 473 automatically provides user A with answers to questions that may be helpful to have answers to while investigating event 9291. If user A clicks on any of the links in the event description (474), she will be taken to a corresponding dossier for the link. As one example, suppose user A clicks on link 475. She will then be presented with interface 477 shown in Fig. 4J.

[00536] Interface 477 is an embodiment of a dossier for a domain. In this example, the domain is "examplebad.com," as shown in region 478. Suppose user A would like to track down more information about interactions entity A resources have made with examplebad.com between January 1 and March 20. She selects the appropriate time period in region 479 and information in the other portions of interface 477 automatically update to provide various information corresponding to the selected time frame. As one example, user A can see that contact was made with examplebad.com a total of 17 times during the time period (480), as well as a list of each contact (481). Various statistical information is also included in the dossier for the time period (482). If she scrolls down in interface 477, user A will be able to view various polygraphs associated with examplebad.com, such as an application-communication polygraph (483).

[00537] Data stored in data store 30 can be internally organized as an activity graph. In the activity graph, nodes are also referred to as Entities. Activities generated by Entities are modeled as directional edges between nodes. Thus, each edge is an activity between two Entities. One example of an Activity is a "login" Activity, in which a user Entity logs into a machine Entity (with a directed edge from the user to the machine). A second example of an Activity is a "launch" Activity, in which a parent process launches a child process (with a directed edge from the parent to the child). A third example of an Activity is a "DNS query" Activity, in which either a process or a machine performs a query (with a directed edge from the requestor to the answer, e.g., an edge from a process to www.example.com). A fourth example of an Activity is a network "connected to" Activity, in which processes, IP addresses, and listen ports can connect to each other (with a directed edge from the initiator to the server).

[00538] As will be described in more detail below, query service 166 provides either relational views or graph views on top of data stored in data store 30. Typically, a user will want to see data filtered using the activity graph. For example, if an entity was not involved in an activity in a given time period, that entity should be filtered out of query results. Thus, a request to show "all machines" in a given time frame will be interpreted as "show distinct machines that were active" during the time frame.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00662

[00539] Query service 166 relies on three main data model elements: fields, entities, and filters. As used herein, a field is a collection of values with the same type (logical and physical). A field can be represented in a variety of ways, including: 1. a column of relations (table/view), 2. a return field from another entity, 3. an SQL aggregation (e.g., COUNT, SUM, etc.), 4. an SQL expression with the references of other fields specified, and 5. a nested field of a JSON object. As viewed by query service 166, an entity is a collection of fields that describe a data set. The data set can be composed in a variety of ways, including: 1. a relational table, 2. a parameterized SQL statement, 3. DynamicSQL created by a Java function, and 4. join/project/aggregate/subclass of other entities. Some fields are common for all entities. One example of such a field is a "first observed" timestamp (when first use of the entity was detected). A second example of such a field is the entity classification type (e.g., one of: 1. Machine (on which an agent is installed), 2. Process, 3. Binary, 4. UID, 5. IP, 6. DNS Information, 7. ListenPort, and 8. PType). A third example of such a field is a "last observed" timestamp.

[00540] A filter is an operator that: 1. takes an entity and field values as inputs, 2. a valid SQL expression with specific reference(s) of entity fields, or 3. is a conjunct/disjunct of filters. As will be described in more detail below, filters can be used to filter data in various ways, and limit data returned by query service 166 without changing the associated data set.

[00541] As mentioned above, a dossier is a template for a collection of visualizations. Each visualization (e.g., the box including chart 484) has a corresponding card, which identifies particular target information needed (e.g., from data store 30) to generate the visualization. In various embodiments, data platform 12 maintains a global set of dossiers/cards. Users of data platform 12 such as user A can build their own dashboard interfaces using preexisting dossiers/cards as components, and/or they can make use of a default dashboard (which incorporates various of such dossiers/cards).

[00542] A JSON file can be used to store multiple cards (e.g., as part of a query service catalog). A particular card is represented by a single JSON object with a unique name as a field name.

[00543] Each card may be described by the following named fields:

[00544] TYPE: the type of the card. Example values include:

[00545] * Entity (the default type)

[00546] * SQL

[00547] * Filters

[00548] * DynamicSQL

[00549] * graphFilter

[00550] * graph

[00551] * Function

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00663

[00552] * Template

[00553] PARAMETERS: a JSON array object that contains an array of parameter objects with the following fields:

[00554] * name (the name of the parameter)

[00555] * required (a Boolean flag indicating whether the parameter is required or not)

[00556] * default (a default value of the parameter)

[00557] * props (a generic JSON object for properties of the parameter. Possible values are: "utype" (a user defined type), and "scope" (an optional property to configure a namespace of the parameter))

[00558] * value (a value for the parameter - non-null to override the default value defined in nested source entities)

[00559] SOURCES: a JSON array object explicitly specifying references of input entities. Each source reference has the following attributes:

[00560] * name (the card/entity name or fully-qualified Table name)

[00561] * type (required for base Table entity)

[00562] * alias (an alias to access this source entity in other fields (e.g., returns, filters, groups, etc))

[00563] RETURNS: a required JSON array object of a return field object. A return field object can be described by the following attributes:

[00564] * field (a valid field name from a source entity)

[00565] * expr (a valid SQL scalar expression. References to input fields of source entities are specified in the format of #{Entity.Field}. Parameters can also be used in the expression in the format of ${ParameterName})

[00566] * type (the type of field, which is required for return fields specified by expr. It is also required for all return fields of an Entity with an SQL type)

[00567] * alias (the unique alias for return field)

[00568] * aggr (possible aggregations are: COUNT, COUNT_DISTINCT, DISTINCT, MAX, MIN, AVG, SUM, FIRST_VALUE, LAST_VALUE)

[00569] * case (JSON array object represents conditional expressions "when" and "expr")

[00570] * fieldsFrom, and, except (specification for projections from a source entity with excluded fields)

[00571] * props (general JSON object for properties of the return field. Possible properties include: "filterGroup," "title," "format," and "utype")

[00572] PROPS: generic JSON objects for other entity properties

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00664

[00573] SQL: a JSON array of string literals for SQL statements. Each string literal can contain parameterized expressions ${ParameterName} and/or composable entity by #{EntityName}

[00574] GRAPH: required for graph entity. Has the following required fields:

[00575] * source (including "type," "props," and "keys")

[00576] * target (including "type," "props," and "keys")

[00577] * edge (including "type" and "props")

[00578] JOINS: a JSON array of join operators. Possible fields for a join operator include:

[00579] * type (possible join types include: "loj" - Left Outer Join, "join" - Inner Join, "in" - Semi Join, "implicit" - Implicit Join)

[00580] * left (a left hand side field of join)

[00581] * right (a right hand side field of join)

[00582] * keys (key columns for multi-way joins)

[00583] * order (a join order of multi-way joins)

[00584] FKEYS: a JSON array of FilterKey(s). The fields for a FilterKey are:

[00585] * type (type of FilterKey)

[00586] * fieldRefs (reference(s) to return fields of an entity defined in the sources field)

[00587] * alias (an alias of the FilterKey, used in implicit join specification)

[00588] FILTERS: a JSON array of filters (conjunct). Possible fields for a filter include:

[00589] * type (types of filters, including: "eq" - equivalent to SQL =, "ne" - equivalent to SQL <>, "ge" - equivalent to SQL >=, "gt" - equivalent to SQL >, "le" - equivalent to SQL <=, "lt" - equivalent to SQL <, "like" - equivalent to SQL LIKE, "not_like" - equivalent to SQL NOT LIKE, "rlike" - equivalent to SQL RLIKE (Snowflake specific), "not_rlike" - equivalent to SQL NOT RLIKE (Snowflake specific), "in" - equivalent to SQL IN, "not_in" - equivalent to SQL NOT IN)

[00590] * expr (generic SQL expression)

[00591] * field (field name)

[00592] * value (single value)

[00593] * values (for both IN and NOT IN)

[00594] ORDERS: a JSON array of ORDER BY for returning fields. Possible attributes for the ORDER BY clause include:

[00595] * field (field ordinal index (1 based) or field alias)

[00596] * order (asc/desc, default is ascending order)

[00597] GROUPS: a JSON array of GROUP BY for returning fields. Field attributes are:

[00598] * field (ordinal index (1 based) or alias from the return fields)

[00599] LIMIT: a limit for the number of records to be returned

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00665

[00600] OFFSET: an offset of starting position of returned data. Used in combination with limit for pagination.

[00601] Suppose customers of data platform 12 (e.g., entity A and entity B) request new data transformations or a new aggregation of data from an existing data set (as well as a corresponding visualization for the newly defined data set). As mentioned above, the data models and filtering applications used by data platform 12 are extensible. Thus, two example scenarios of extensibility are (1) extending the filter data set, and (2) extending a FilterKey in the filter data set.

[00602] Data platform 12 includes a query service catalog that enumerates cards available to users of data platform 12. New cards can be included for use in data platform 12 by being added to the query service catalog (e.g., by an operator of data platform 12). For reusability and maintainability, a single external-facing card (e.g., available for use in a dossier) can be composed of multiple (nested) internal cards. Each newly added card (whether external or internal) will also have associated FilterKey(s) defined. A user interface (UI) developer can then develop a visualization for the new data set in one or more dossier templates. The same external card can be used in multiple dossier templates, and a given external card can be used multiple times in the same dossier (e.g., after customization). Examples of external card customization include customization via parameters, ordering, and/or various mappings of external data fields (columns).

[00603] As mentioned above, a second extensibility scenario is one in which a FilterKey in the filter data set is extended (i.e., existing template functions are used to define a new data set). As also mentioned above, data sets used by data platform 12 are composable/reusable/extensible, irrespective of whether the data sets are relational or graph data sets. One example data set is the User Tracking polygraph, which is generated as a graph data set (comprising nodes and edges). Like other polygraphs, User Tracking is an external data set that can be visualized both as a graph (via the nodes and edges) and can also be used as a filter data set for other cards, via the cluster identifier (CID) field.

[00604] As mentioned above, as users such as user A navigate through/interact with interfaces provided by data platform 12 (e.g., as shown in Fig. 4H), such interactions trigger query service 166 to generate and perform queries against data store 30. Dynamic composition of filter datasets can be implemented using FilterKeys and FilterKey Types. A FilterKey can be defined as a list of columns and/or fields in a nested structure (e.g., JSON). Instances of the same FilterKey Type can be formed as an Implicit Join Group. The same instance of a FilterKey can participate in different Implicit Join Groups. A list of relationships among all possible Implicit Join Groups is

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00666

represented as a Join graph for the entire search space to create a final data filter set by traversing edges and producing Join Path(s).

[00605] Each card (e.g., as stored in the query service catalog and used in a dossier) can be introspected by a /card/describe/CardID REST request.

[00606] At runtime (e.g., whenever it receives a request from web app 120), query service 166 parses the list of implicit joins and creates a Join graph to manifest relationships of FilterKeys among Entities. A Join graph (an example of which is depicted in Fig. 4K) comprises a list of Join Link(s). A Join Link represents each implicit join group by the same FilterKey type. A Join Link maintains a reverse map (Entity-to-FilterKey) of FilterKeys and their Entities. As previously mentioned, Entities can have more than one FilterKey defined. The reverse map guarantees one FilterKey per Entity can be used for each JoinLink. Each JoinLink also maintains a list of entities for the priority order of joins. Each JoinLink is also responsible for creating and adding directional edge(s) to graphs. An edge represents a possible join between two Entities.

[00607] At runtime, each Implicit Join uses the Join graph to find all possible join paths. The search of possible join paths starts with the outer FilterKey of an implicit join. One approach is to use a shortest path approach, with breadth first traversal and subject to the following criteria:

[00608] * Use the priority order list of Join Links for all entities in the same implicit join group.

[00609] * Stop when a node (Entity) is reached which has local filter(s).

[00610] * Include all join paths at the same level (depth).

[00611] * Exclude join paths based on the predefined rules (path of edges).

[00612] Fig. 4L illustrates an example of a process for dynamically generating and executing a query. In various embodiments, process 485 is performed by data platform 12. The process begins at 486 when a request is received to filter information associated with activities within a network environment. One example of such a request occurs in response to user A clicking on tab 465. Another example of such a request occurs in response to user A clicking on link 464-1. Yet another example of such a request occurs in response to user A clicking on link 464-2 and selecting (e.g., from a dropdown) an option to filter (e.g., include, exclude) based on specific criteria that she provides (e.g., an IP address, a username, a range of criteria, etc.).

[00613] At 487, a query is generated based on an implicit join. One example of processing that can be performed at 487 is as follows. As explained above, one way dynamic composition of filter datasets can be implemented is by using FilterKeys and FilterKey Types. And, instances of the same FilterKey Type can be formed as an Implicit Join Group. A Join graph for the entire search space can be constructed from a list of all relationships among all possible Join Groups. And, a final data filter set can be created by traversing edges and producing one or more Join Paths. Finally, the shortest path in the join paths is used to generate an SQL query string.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00667

[00614] One approach to generating an SQL query string is to use a query building library (authored in an appropriate language such as Java). For example, a common interface "sqlGen" may be used in conjunction with process 485 is as follows. First, a card/entity is composed by a list of input cards/entities, where each input card recursively is composed by its own list of input cards. This nested structure can be visualized as a tree of query blocks(SELECT) in standard SQL constructs. SQL generation can be performed as the traversal of the tree from root to leaf entities (top-down), calling the sqlGen of each entity. Each entity can be treated as a subclass of the Java class(Entity). An implicit join filter (EntityFilter) is implemented as a subclass of Entity, similar to the right hand side of a SQL semi-join operator. Unlike the static SQL semi-join construct, it is conditionally and recursively generated even if it is specified in the input sources of the JSON specification. Another recursive interface can also be used in conjunction with process 485, preSQLGen, which is primarily the entry point for EntityFilter to run a search and generate nested implicit join filters. During preSQLGen recursive invocations, the applicability of implicit join filters is examined and pushed down to its input subquery list. Another top-down traversal, pullUpCachable, can be used to pull up common sub-query blocks, including those dynamically generated by preSQLGen, such that SELECT statements of those cacheable blocks are generated only once at top-level WITH clauses. A recursive interface, sqlWith, is used to generate nested subqueries inside WITH clauses. The recursive calls of a sqlWith function can generate nested WITH clauses as well. An sqlFrom function can be used to generate SQL FROM clauses by referencing those subquery blocks in the WITH clauses. It also produces INNER/OUTER join operators based on the joins in the specification. Another recursive interface, sqlWhere, can be used to generate conjuncts and disjuncts of local predicates and semi-join predicates based on implicit join transformations. Further, sqlProject, sqlGroupBy, sqlOrderBy, and sqlLimitOffset can respectively be used to translate the corresponding directives in JSON spec to SQL SELECT list, GROUP BY, ORDER BY, and LIMIT/OFFSET clauses.

[00615] Returning to process 485, at 488, the query (generated at 487) is used to respond to the request. As one example of the processing performed at 488, the generated query is used to query data store 30 and provide (e.g., to web app 120) fact data formatted in accordance with a schema (e.g., as associated with a card associated with the request received at 486).

[00616] Although the examples described herein largely relate to embodiments where data is collected from agents and ultimately stored in a data store such as those provided by Snowflake, in other embodiments data that is collected from agents and other sources may be stored in different ways. For example, data that is collected from agents and other sources may be stored in a data warehouse, data lake, data mart, and/or any other data store.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00668

[00617] A data warehouse may be embodied as an analytic database (e.g., a relational database) that is created from two or more data sources. Such a data warehouse may be leveraged to store historical data, often on the scale of petabytes. Data warehouses may have compute and memory resources for running complicated queries and generating reports. Data warehouses may be the data sources for business intelligence ('BI') systems, machine learning applications, and/or other applications. By leveraging a data warehouse, data that has been copied into the data warehouse may be indexed for good analytic query performance, without affecting the write performance of a database (e.g., an Online Transaction Processing ('OLTP') database). Data warehouses also enable the joining data from multiple sources for analysis. For example, a sales OLTP application probably has no need to know about the weather at various sales locations, but sales predictions could take advantage of that data. By adding historical weather data to a data warehouse, it would be possible to factor it into models of historical sales data.

[00618] Data lakes, which store files of data in their native format, may be considered as "schema on read" resources. As such, any application that reads data from the lake may impose its own types and relationships on the data. Data warehouses, on the other hand, are "schema on write," meaning that data types, indexes, and relationships are imposed on the data as it is stored in the EDW. "Schema on read" resources may be beneficial for data that may be used in several contexts and poses little risk of losing data. "Schema on write" resources may be beneficial for data that has a specific purpose, and good for data that must relate properly to data from other sources. Such data stores may include data that is encrypted using homomorphic encryption, data encrypted using privacy-preserving encryption, smart contracts, non-fungible tokens, decentralized finance, and other techniques.

[00619] Data marts may contain data oriented towards a specific business line whereas data warehouses contain enterprise-wide data. Data marts may be dependent on a data warehouse, independent of the data warehouse (e.g., drawn from an operational database or external source), or a hybrid of the two. In embodiments described herein, different types of data stores (including combinations thereof) may be leveraged. Such data stores may be proprietary or may be embodied as vendor provided products or services such as, for example, Google BigQuery, Druid, Amazon Redshift, IBM Db2, Dremio, Databricks Lakehouse Platform, Cloudera, Azure Synapse Analytics, and others.

[00620] The deployments (e.g., a customer's cloud deployment) that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may be provisioned, deployed, and/or managed using infrastructure as code ('IaC'). IaC involves the managing and/or provisioning of infrastructure through code

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00669

instead of through manual processes. With IaC, configuration files may be created that include infrastructure specifications. IaC can be beneficial as configurations may be edited and distributed, while also ensuring that environments are provisioned in a consistent manner. IaC approaches may be enabled in a variety of ways including, for example, using IaC software tools such as Terraform by HashiCorp. Through the usage of such tools, users may define and provide data center infrastructure using JavaScript Object Notation ('JSON'), YAML, proprietary formats, or some other format. In some embodiments, the configuration files may be used to emulate a cloud deployment for the purposes of analyzing the emulated cloud deployment using the systems described herein. Likewise, the configuration files themselves may be used as inputs to the systems described herein, such that the configuration files may be inspected to identify vulnerabilities, misconfigurations, violations of regulatory requirements, or other issues. In fact, configuration files for multiple cloud deployments may even be used by the systems described herein to identify best practices, to identify configuration files that deviate from typical configuration files, to identify configuration files with similarities to deployments that have been determined to be deficient in some way, or the configuration files may be leveraged in some other ways to detect vulnerabilities, misconfigurations, violations of regulatory requirements, or other issues prior to deploying an infrastructure that is described in the configuration files. In some embodiments the techniques described herein may be use in multi-cloud, multi-tenant, cross-cloud, cross-tenant, cross-user, industry cloud, digital platform, and other scenarios depending on specific need or situation.

[00621] In some embodiments, the deployments that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may be monitored to determine the extent to which a particular component has experienced "drift" relative to its associated IaC configuration. Discrepancies between how cloud resources were defined in an IaC configuration file and how they are currently configured in runtime may be identified and remediation workflows may be initiated to generate an alert, reconfigure the deployment, or take some other action. Such discrepancies may occur for a variety of reasons. Such discrepancies may occur, for example, due to maintenance operations being performed, due to incident response tasks being carried out, or for some other reason. Readers will appreciate that while IaC helps avoid initial misconfigurations of a deployment by codifying and enforcing resource creation, resource configuration, security policies, and so on, the systems described herein may prevent unwanted drift from occurring during runtime and after a deployment has been created in accordance with an IaC configuration.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00670

[00622] In some embodiments, the deployments (e.g., a customer's cloud deployment) that are analyzed, monitored, evaluated, or otherwise observed by the systems described herein (e.g., systems that include components such as the platform 12 of Fig. 1D, the data collection agents described herein, and/or other components) may also be provisioned, deployed, and/or managed using security as code ('SaC'). SaC extends IaC concepts by defining cybersecurity policies and/or standards programmatically, so that the policies and/or standards can be referenced automatically in the configuration scripts used to provision cloud deployments. Stated differently, SaC can automate policy implementation and cloud deployments may even be compared with the policies to prevent "drift." For example, if a policy is created where all personally identifiable information ('PII') or personal health information ('PHI') must be encrypted when it is stored, that policy is translated into a process that is automatically launched whenever a developer submits code, and code that violates the policy may be automatically rejected.

[00623] In some embodiments, SaC may be implemented by initially classifying workloads (e.g., by sensitivity, by criticality, by deployment model, by segment). Policies that can be instantiated as code may subsequently be designed. For example, compute-related policies may be designed, access-related policies may be designed, application-related policies may be designed, network-related policies may be designed, data-related policies may be designed, and so on. Security as code may then be instantiated through architecture and automation, as successful implementation of SaC can benefit from making key architectural-design decisions and executing the right automation capabilities. Next, operating model protections may be built and supported. For example, an operating model may "shift left" to maximize self-service and achieve full-life-cycle security automation (e.g., by standardizing common development toolchains, CI/CD pipelines, and the like). In such an example, security policies and access controls may be part of the pipeline, automatic code review and bug/defect detection may be performed, automated build processes may be performed, vulnerability scanning may be performed, checks against a risk-control framework may be made, and other tasks may be performed all before deploying an infrastructure or components thereof.

[00624] The systems described herein may be useful in analyzing, monitoring, evaluating, or otherwise observing a GitOps environment. In a GitOps environment, Git may be viewed as the one and only source of truth. As such, GitOps may require that the desired state of infrastructure (e.g., a customer's cloud deployment) be stored in version control such that the entire audit trail of changes to such infrastructure can be viewed or audited. In a GitOps environment, all changes to infrastructure are embodied as fully traceable commits that are associated with committer information, commit IDs, time stamps, and/or other information. In such an embodiment, both an

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00671

application and the infrastructure (e.g., a customer's cloud deployment) that supports the execution of the application are therefore versioned artifacts and can be audited using the gold standards of software development and delivery. Readers will appreciate that while the systems described herein are described as analyzing, monitoring, evaluating, or otherwise observing a GitOps environment, in other embodiments other source control mechanisms may be utilized for creating infrastructure, making changes to infrastructure, and so on. In these embodiments, the systems described herein may similarly be used for analyzing, monitoring, evaluating, or otherwise observing such environments.

[00625] As described in other portions of the present disclosure, the systems described herein may be used to analyze, monitor, evaluate, or otherwise observe a customer's cloud deployment. While securing traditional datacenters requires managing and securing an IP-based perimeter with networks and firewalls, hardware security modules ('HSMs'), security information and event management ('SIEM') technologies, and other physical access restrictions, such solutions are not particularly useful when applied to cloud deployments. As such, the systems described herein may be configured to interact with and even monitor other solutions that are appropriate for cloud deployments such as, for example, "zero trust" solutions.

[00626] A zero trust security model (a.k.a., zero trust architecture) describes an approach to the design and implementation of IT systems. A primary concept behind zero trust is that devices should not be trusted by default, even if they are connected to a managed corporate network such as the corporate LAN and even if they were previously verified. Zero trust security models help prevent successful breaches by eliminating the concept of trust from an organization's network architecture. Zero trust security models can include multiple forms of authentication and authorization (e.g., machine authentication and authorization, human/user authentication and authorization) and can also be used to control multiple types of accesses or interactions (e.g., machine-to-machine access, human-to-machine access).

[00627] In some embodiments, the systems described herein may be configured to interact with zero trust solutions in a variety of ways. For example, agents that collect input data for the systems described herein (or other components of such systems) may be configured to access various machines, applications, data sources, or other entity through a zero trust solution, especially where local instances of the systems described herein are deployed at edge locations. Likewise, given that zero trust solutions may be part of a customer's cloud deployment, the zero trust solution itself may be monitored to identify vulnerabilities, anomalies, and so on. For example, network traffic to and from the zero trust solution may be analyzed, the zero trust solution may be monitored to detect unusual interactions, log files generated by the zero trust solution may be gathered and analyzed, and so on.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00672

[00628] In some embodiments, the systems described herein may leverage various tools and mechanisms in the process of performing its primary tasks (e.g., monitoring a cloud deployment). For example, Linux eBPF is mechanism for writing code to be executed in the Linux kernel space. Through the usage of eBPF, user mode processes can hook into specific trace points in the kernel and access data structures and other information. For example, eBPF may be used to gather information that enables the systems described herein to attribute the utilization of networking resources or network traffic to specific processes. This may be useful in analyzing the behavior of a particular process, which may be important for observability/SIEM.

[00629] The systems described may be configured to collect security event logs (or any other type of log or similar record of activity) and telemetry in real time for threat detection, for analyzing compliance requirements, or for other purposes. In such embodiments, the systems described herein may analyze telemetry in real time (or near real time), as well as historical telemetry, to detect attacks or other activities of interest. The attacks or activities of interest may be analyzed to determine their potential severity and impact on an organization. In fact, the attacks or activities of interest may be reported, and relevant events, logs, or other information may be stored for subsequent examination.

[00630] In one embodiment, systems described herein may be configured to collect security event logs (or any other type of log or similar record of activity) and telemetry in real time to provide customers with a SIEM or SIEM-like solution. SIEM technology aggregates event data produced by security devices, network infrastructure, systems, applications, or other source. Centralizing all of the data that may be generated by a cloud deployment may be challenging for a traditional SIEM, however, as each component in a cloud deployment may generate log data or other forms of machine data, such that the collective amount of data that can be used to monitor the cloud deployment can grow to be quite large. A traditional SIEM architecture, where data is centralized and aggregated, can quickly result in large amounts of data that may be expensive to store, process, retain, and so on. As such, SIEM technologies may frequently be implemented such that silos are created to separate the data.

[00631] In some embodiments of the present disclosure, data that is ingested by the systems described herein may be stored in a cloud-based data warehouse such as those provided by Snowflake and others. Given that companies like Snowflake offer data analytics and other services to operate on data that is stored in their data warehouses, in some embodiments one or more of the components of the systems described herein may be deployed in or near Snowflake as part of a secure data lake architecture (a.k.a., a security data lake architecture, a security data lake/warehouse). In such an embodiment, components of the systems described herein may be deployed in or near Snowflake to collect data, transform data, analyze data for the purposes of

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00673

detecting threats or vulnerabilities, initiate remediation workflows, generate alerts, or perform any of the other functions that can be performed by the systems described herein. In such embodiments, data may be received from a variety of sources (e.g., EDR or EDR-like tools that handle endpoint data, cloud access security broker ('CASB') or CASB-like tools that handle data describing interactions with cloud applications, Identity and Access Management ('IAM') or IAM-like tools, and many others), normalized for storage in a data warehouse, and such normalized data may be used by the systems described herein. In fact, the systems described herein may actually implement the data sources (e.g., an EDR tool, a CASB tool, an IAM tool) described above.

[00632] In some embodiments one data source that is ingested by the systems described herein is log data, although other forms of data such as network telemetry data (flows and packets) and/or many other forms of data may also be utilized. In some embodiments, event data can be combined with contextual information about users, assets, threats, vulnerabilities, and so on, for the purposes of scoring, prioritization and expediting investigations. In some embodiments, input data may be normalized, so that events, data, contextual information, or other information from disparate sources can be analyzed more efficiently for specific purposes (e.g., network security event monitoring, user activity monitoring, compliance reporting). The embodiments described here offer real-time analysis of events for security monitoring, advanced analysis of user and entity behaviors, querying and long-range analytics for historical analysis, other support for incident investigation and management, reporting (for compliance requirements, for example), and other functionality.

[00633] In some embodiments, the systems described herein may be part of an application performance monitoring ('APM') solution. APM software and tools enable the observation of application behavior, observation of its infrastructure dependencies, observation of users and business key performance indicators ('KPIs') throughout the application's life cycle, and more. The applications being observed may be developed internally, as packaged applications, as software as a service ('SaaS'), or embodied in some other ways. In such embodiments, the systems described herein may provide one or more of the following capabilities:

[00634] The ability to operate as an analytics platform that ingests, analyzes, and builds context from traces, metrics, logs, and other sources.

[00635] Automated discovery and mapping of an application and its infrastructure components.

[00636] Observation of an application's complete transactional behavior, including interactions over a data communications network.

[00637] Monitoring of applications running on mobile (native and browser) and desktop devices.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00674

[00638] Identification of probable root causes of an application's performance problems and their impact on business outcomes.

[00639] Integration capabilities with automation and service management tools.

[00640] Analysis of business KPIs and user journeys (for example, login to check-out).

[00641] Domain-agnostic analytics capabilities for integrating data from third-party sources.

[00642] Endpoint monitoring to understand the user experience and its impact on business outcomes.

[00643] Support for virtual desktop infrastructure ('VDI') monitoring.

[00644] In embodiments where the systems described herein are used for APM, some components of the system may be modified, other components may be added, some components may be removed, and other components may remain the same. In such an example, similar mechanisms as described elsewhere in this disclosure may be used to collect information from the applications, network resources used by the application, and so on. The graph based modelling techniques may also be leveraged to perform some of the functions mentioned above, or other functions as needed.

[00645] In some embodiments, the systems described herein may be part of a solution for developing and/or managing artificial intelligence ('AI') or machine learning ('ML') applications. For example, the systems described herein may be part of an AutoML tool that automate the tasks associated with developing and deploying ML models. In such an example, the systems described herein may perform various functions as part of an AutoML tool such as, for example, monitoring the performance of a series of processes, microservices, and so on that are used to collectively form the AutoML tool. In other embodiments, the systems described herein may perform other functions as part of an AutoML tool or may be used to monitor, analyze, or otherwise observe an environment that the AutoML tool is deployed within.

[00646] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include other forms of AI/ML tools. For example, the systems described herein may manage, analyze, or otherwise observe deployments that include AI services. AI services are, like other resources in an as-a-service model, ready-made models and AI applications that are consumable as services and made available through APIs. In such an example, rather than using their own data to build and train models for common activities, organizations may access pre-trained models that accomplish specific tasks. Whether an organization needs natural language processing ('NLP'), automatic speech recognition ('ASR'), image recognition, or some other capability, AI services simply plug-and-play into an application through an API. Likewise, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include other forms of AI/ML tools such as Amazon

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00675

Sagemaker (or other cloud machine-learning platform that enables developers to create, train, and deploy ML models) and related services such as Data Wrangler (a service to accelerate data prep for ML) and Pipelines (a CI/CD service for ML).

[00647] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include various data services. For example, data services may include secure data sharing services, data marketplace services, private data exchanges services, and others. Secure data sharing services can allow access to live data from its original location, where those who are granted access to the data simply reference the data in a controlled and secure manner, without latency or contention from concurrent users. Because changes to data are made to a single version, data remains up-to-date for all consumers, which ensures data models are always using the latest version of such data. Data marketplace services operate as a single location to access live, ready-to-query data (or data that is otherwise ready for some other use). A data marketplace can even include a "feature stores," which can allow data scientists to repurpose existing work. For example, once a data scientist has converted raw data into a metric (e.g., costs of goods sold), this universal metric can be found quickly and used by other data scientists for quick analysis against that data.

[00648] In some embodiments, the systems described herein may be used to manage, analyze, or otherwise observe deployments that include distributed training engines or similar mechanisms such as, for example, such as tools built on Dask. Dask is an open source library for parallel computing that is written in Python. Dask is designed to enable data scientists to improve model accuracy faster, as Dask enables data scientists can do everything in Python end-to-end, which means that they no longer need to convert their code to execute in environments like Apache Spark. The result is reduced complexity and increased efficiency. The systems described herein may also be used to manage, analyze, or otherwise observe deployments that include technologies such as RAPIDS (an open source Python framework which is built on top of Dask). RAPIDS optimizes compute time and speed by providing data pipelines and executing data science code entirely on graphics processing units (GPUs) rather than CPUs. Multi-cluster, shared data architecture, DataFrames, Java user-defined functions (UDF) are supported to enable trained models to run within a data warehouse.

[00649] In some embodiments, the systems described herein may be leveraged for the specific use case of detecting and/or remediating ransomware attacks and/or other malicious action taken with respect to data, systems, and/or other resources associated with one or more entities. Ransomware is a type of malware from cryptovirology that threatens to publish the victim's data or perpetually block access to such data unless a ransom is paid. In such embodiments, ransomware attacks may be carried out in a manner such that patterns (e.g., specific process-to-

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00676

process communications, specific data access patterns, unusual amounts of encryption/re-encryption activities) emerge, where the systems described herein may monitor for such patterns. Alternatively, ransomware attacks may involve behavior that deviates from normal behavior of a cloud deployment that is not experiencing a ransomware attack, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a ransomware attack.

[00650] In some embodiments, particular policies may be put in place the systems described herein may be configured to enforce such policies as part of an effort to thwart ransomware attacks. For example, particular network sharing protocols (e.g., Common Internet File System ('CIFS'), Network File System ('NFS')) may be avoided when implementing storage for backup data, policies that protect backup systems may be implemented and enforced to ensure that usable backups are always available, multifactor authentication for particular accounts may be utilized and accounts may be configured with the minimum privilege required to function, isolated recovery environments may be created and isolation may be monitored and enforced to ensure the integrity of the recovery environment, and so on. As described in the present disclosure, the systems described herein may be configured to explicitly enforce such policies or may be configured to detect unusual activity that represents a violation of such policies, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a violation of a particular policy.

[00651] Readers will appreciate that ransomware attacks are often deployed as part of a larger attack that may involve, for example:

[00652] Penetration of the network through means such as, for example, stolen credentials and remote access malware.

[00653] Stealing of credentials for critical system accounts, including subverting critical administrative accounts that control systems such as backup, Active Directory ('AD'), DNS, storage admin consoles, and/or other key systems.

[00654] Attacks on a backup administration console to turn off or modify backup jobs, change retention policies, or even provide a roadmap to where sensitive application data is stored.

[00655] Data theft attacks.

[00656] As a result of the many aspects that are part of a ransomware attack, embodiments of the present disclosure may be configured as follows:

[00657] The systems may include one or more components that detect malicious activity based on the behavior of a process.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00677

[00658] The systems may include one or more components that store indicator of compromise ('IOC') or indicator of attack ('IOA') data for retrospective analysis.

[00659] The systems may include one or more components that detect and block fileless malware attacks.

[00660] The systems may include one or more components that remove malware automatically when detected.

[00661] The systems may include a cloud-based, SaaS-style, multitenant infrastructure.

[00662] The systems may include one or more components that identify changes made by malware and provide the recommended remediation steps or a rollback capability.

[00663] The systems may include one or more components that detect various application vulnerabilities and memory exploit techniques.

[00664] The systems may include one or more components that continue to collect suspicious event data even when a managed endpoint is outside of an organization's network.

[00665] The systems may include one or more components that perform static, on-demand malware detection scans of folders, drives, devices, or other entities.

[00666] The systems may include data loss prevention (DLP) functionality.

[00667] In some embodiments, the systems described herein may manage, analyze, or otherwise observe deployments that include deception technologies. Deception technologies allow for the use of decoys that may be generated based on scans of true network areas and data. Such decoys may be deployed as mock networks running on the same infrastructure as the real networks, but when an intruder attempts to enter the real network, they are directed to the false network and security is immediately notified. Such technologies may be useful for detecting and stopping various types of cyber threats such as, for example, Advanced Persistent Threats ('APTs'), malware, ransomware, credential dumping, lateral movement and malicious insiders. To continue to outsmart increasingly sophisticated attackers, these solutions may continuously deploy, support, refresh and respond to deception alerts.

[00668] In some embodiments, the systems described herein may manage, analyze, or otherwise observe deployments that include various authentication technologies, such as multi-factor authentication and role-based authentication. In fact, the authentication technologies may be included in the set of resources that are managed, analyzed, or otherwise observed as interactions with the authentication technologies may monitored. Likewise, log files or other information retained by the authentication technologies may be gathered by one or more agents and used as input to the systems described herein.

[00669] In some embodiments, the systems described herein may be leveraged for the specific use case of detecting supply chain attacks. More specifically, the systems described herein may

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00678

be used to monitor a deployment that includes software components, virtualized hardware components, and other components of an organization's supply chain such that interactions with an outside partner or provider with access to an organization's systems and data can be monitored. In such embodiments, supply chain attacks may be carried out in a manner such that patterns (e.g., specific interactions between internal and external systems) emerge, where the systems described herein may monitor for such patterns. Alternatively, supply chain attacks may involve behavior that deviates from normal behavior of a cloud deployment that is not experiencing a supply chain attack, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a supply chain attack.

[00670] In some embodiments, the systems described herein may be leveraged for other specific use cases such as, for example, detecting the presence of (or preventing infiltration from) cryptocurrency miners (e.g., bitcoin miners), token miners, hashing activity, non-fungible token activity, other viruses, other malware, and so on. As described in the present disclosure, the systems described herein may monitor for such threats using known patterns or by detecting unusual activity, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a particular type of threat, intrusion, vulnerability, and so on.

[00671] The systems described herein may also be leveraged for endpoint protection, such the systems described herein form all of or part of an endpoint protection platform. In such an embodiment, agents, sensors, or similar mechanisms may be deployed on or near managed endpoints such as computers, servers, virtualized hardware, internet of things ('IotT') devices, mobile devices, phones, tablets, watches, other personal digital devices, storage devices, thumb drives, secure data storage cards, or some other entity. In such an example, the endpoint protection platform may provide functionality such as:

[00672] Prevention and protection against security threats including malware that uses file-based and fileless exploits.

[00673] The ability to apply control (allow/block) to access of software, scripts, processes, microservices, and so on.

[00674] The ability to detect and prevent threats using behavioral analysis of device activity, application activity, user activity, and/or other data.

[00675] The ability for facilities to investigate incidents further and/or obtain guidance for remediation when exploits evade protection controls

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00679

[00676] The ability to collect and report on inventory, configuration and policy management of the endpoints.

[00677] The ability to manage and report on operating system security control status for the monitored endpoints.

[00678] The ability to scan systems for vulnerabilities and report/manage the installation of security patches.

[00679] The ability to report on internet, network and/or application activity to derive additional indications of potentially malicious activity.

[00680] Example embodiments are described in which policy enforcement, threat detection, or some other function is carried out by the systems described herein by detecting unusual activity, such that the mere presence of unusual activity may trigger the systems described herein to generate alerts or take some other action, even without explicit knowledge that the unusual activity is associated with a particular type of threat, intrusion, vulnerability, and so on. Although these examples are largely described in terms of identifying unusual activity, in these examples the systems described herein may be configured to learn what constitutes 'normal activity' – where 'normal activity' is activity observed, modeled, or otherwise identified in the absence of a particular type of threat, intrusion, vulnerability, and so on. As such, detecting 'unusual activity' may alternatively be viewed as detecting a deviation from 'normal activity' such that 'unusual activity' does not need to be identified and sought out. Instead, deviations from 'normal activity' may be assumed to be 'unusual activity'.

[00681] Readers will appreciate that while specific examples of the functionality that the systems described herein can provide are included in the present disclosure, such examples are not to be interpreted as limitations as to the functionality that the systems described herein can provide. Other functionality may be provided by the systems described herein, all of which are within the scope of the present disclosure. For the purposes of illustration and not as a limitation, additional examples can include governance, risk, and compliance ('GRC'), threat detection and incident response, identity and access management, network and infrastructure security, data protection and privacy, identity and access management ('IAM'), and many others.

[00682] In order to provide the functionality described above, the systems described herein or the deployments that are monitored by such systems may implement a variety of techniques. For example, the systems described herein or the deployments that are monitored by such systems may tag data and logs to provide meaning or context, persistent monitoring techniques may be used to monitor a deployment at all times and in real time, custom alerts may be generated based on rules, tags, and/or known baselines from one or more polygraphs, and so on.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00680

[00683] Although examples are described above where data may be collected from one or more agents, in some embodiments other methods and mechanisms for obtaining data may be utilized. For example, some embodiments may utilize agentless deployments where no agent (or similar mechanism) is deployed on one or more customer devices, deployed within a customer's cloud deployment, or deployed at another location that is external to the data platform. In such embodiments, the data platform may acquire data through one or more APIs such as the APIs that are available through various cloud services. For example, one or more APIs that enable a user to access data captured by Amazon CloudTrail may be utilized by the data platform to obtain data from a customer's cloud deployment without the use of an agent that is deployed on the customer's resources. In some embodiments, agents may be deployed as part of a data acquisition service or tool that does not utilize a customer's resources or environment. In some embodiments, agents (deployed on a customer's resources or elsewhere) and mechanisms in the data platform that can be used to obtain data from through one or more APIs such as the APIs that are available through various cloud services may be utilized. In some embodiments, one or more cloud services themselves may be configured to push data to some entity (deployed anywhere), which may or may not be an agent. In some embodiments, other data acquisition techniques may be utilized, including combinations and variations of the techniques described above, each of which is within the scope of the present disclosure.

[00684] Readers will appreciate that while specific examples of the cloud deployments that may be monitored, analyzed, or otherwise observed by the systems described herein have been provided, such examples are not to be interpreted as limitations as to the types of deployments that may be monitored, analyzed, or otherwise observed by the systems described herein. Other deployments may be monitored, analyzed, or otherwise observed by the systems described herein, all of which are within the scope of the present disclosure. For the purposes of illustration and not as a limitation, additional examples can include multi-cloud deployments, on-premises environments, hybrid cloud environments, sovereign cloud environments, heterogeneous environments, DevOps environments, DevSecOps environments, GitOps environments, quantum computing environments, data fabrics, composable applications, composable networks, decentralized applications, and many others.

[00685] Readers will appreciate that while specific examples of the types of data that may be collected, transformed, stored, and/or analyzed by the systems described herein have been provided, such examples are not to be interpreted as limitations as to the types of data that may be collected, transformed, stored, and/or analyzed by the systems described herein. Other types of data can include, for example, data collected from different tools (e.g., DevOps tools, DevSecOps, GitOps tools), different forms of network data (e.g., routing data, network

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00681

translation data, message payload data, Wi-Fi data, Bluetooth data, personal area networking data, payment device data, near field communication data, metadata describing interactions carried out over a network, and many others), data describing processes executing in a container, lambda, EC2 instance, virtual machine, or other execution environment), information describing the execution environment itself, and many other types of data.

[00686] In some embodiments, one or more of the components described above may be deployed using a secure access service edge ('SASE') model or similar model. In a SASE model, the services, functionality, or components described above may be deployed at edge devices (or relatively close to such edge device) such as a user's laptop, tablet, smartphone, or other device. In such a way, network security controls may be delivered on such edge devices. SASE capabilities may be delivered as a service based upon the identity of the entity, real-time context, enterprise security/compliance policies and continuous assessment of risk/trust throughout the sessions, where the identity of entities can be associated with people, groups of people, devices, applications, services, IoT systems or edge computing locations, and so on. In these embodiments, one or more of the components described above may be deployed at or near the edge devices, and the edge devices may even include local applications that are configured to utilize one or more of the components described above where the components are not deployed on the edge devices themselves.

[00687] For further explanation, Fig. 5 sets forth a flowchart illustrating an example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The cloud deployments 508, 514 may be similar to the cloud deployments described above, where a particular cloud deployment can include a variety of components 510, 512 such as one or more applications, one or more data sources, networking resources, processing resources, and other resources. Such components 510, 512 may, in some embodiments, be deployed in the cloud deployments 508, 514 using one or more as-a-service models where software, infrastructure, platforms, databases, and other components as delivered as services. Configuring cloud deployments 508, 514 based on learnings obtained by monitoring other cloud deployments may be carried out using the systems described above. As such, one or more of the steps depicted in Fig. 5 may be performed by the systems described above.

[00688] The example method depicted in Figure 5 includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508. Determining 502 normal behavior for one or more components 510 in a first cloud deployment 508 may be carried out, for example, as described in greater detail above (at times described as identifying 'normal activity') by the systems described above (also referred to herein as a 'data platform').

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00682

[00689] The example method depicted in Figure 5 also includes determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514. Determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514 may also be carried out, for example, as described in greater detail above (at times described as identifying 'normal activity') by the systems described above.

[00690] In some embodiments, a customer-specific data platform may be used to analyze, monitor, or otherwise observe a particular customer's cloud deployment (or some other deployment). Within such a cloud deployment, various clusters may exist. For example, a collection of microservices may form a cluster by virtue of those microservices communicating only (or mostly) with each other. Likewise, one or more cloud computing instances (e.g., one or more EC2 instances) and a database may form a cluster by virtue of the EC2 instances accessing the database as the only source of data utilized by the EC2 instances. Using the techniques and mechanisms described above, such clusters may be identified. Although clusters may be identified and characteristics associated with the cluster may be learned, limited insights may be gained if only a particular customer's cloud deployment is analyzed, monitored, or otherwise observed. In accordance with embodiments of the present disclosure, cross-customer analysis may be leveraged to gain deeper insights than would be gained if only a single customer's cloud deployment is analyzed, monitored, or otherwise observed.

[00691] In some embodiments, cross-customer analysis may be carried out by gathering information related to cloud deployments (or some other deployments) for multiple customers and comparing such information. Using the example described above, information describing clusters identified in a first customer's cloud deployment may be compared to information describing clusters identified in a second customer's cloud deployment for the purposes of identifying similar or identical clusters in each customer's cloud deployment.

[00692] Consider an example in which each customer's deployment included a web server that was deployed in one or more EC2 instances. In such an example, a particular cluster that represents the web server may be identified in each customer's deployment. For example, a first cluster in the first customer's deployment may represent a first web server and a second cluster in the second customer's deployment may represent a second web server. Because the first cluster and the second cluster would have similar characteristics (e.g., each cluster receives data communications using HTTP or HTTPS or any other suitable communication protocol, each cluster communicates with a web browser, each cluster requires similar computing resources, and so on), the first cluster and the second cluster may be identified as being identical clusters by comparing the characteristics of each cluster that each cluster. This process may be repeated

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00683

across the cloud deployments for many customers such that a collection of 'web server' clusters (in this example) may be identified.

[00693] Readers will appreciate that although the example described above relates to an embodiment where a collection of 'web server' clusters are identified in different customer's cloud deployments, identifying the nature or type (e.g., a web server) of the clusters is not required. In fact, by comparing the characteristics of different clusters to each other, similar or identical clusters may be identified even if the exact nature/type of those clusters is not known. For example, a comparison of the characteristics of multiple clusters may only reveal that the cluster are identical, even if such a comparison does not reveal that clusters are 'web server' clusters. Multiple clusters that have been identified as being identical (or sufficiently similar as measured by a threshold) will be referred to throughout the remainder of this document as a "cluster set," where the clusters that are members of the cluster set may be deployed across multiple customer's cloud deployments.

[00694] In some embodiments, information describing each member of the cluster set may be utilized to identify distributions across the cluster set. Distributions may be identified for traditional resource consumption metrics such as, for example, CPU usage, memory usage, network bandwidth usage, and others. A distribution may reveal, for example, that all members of the cluster set utilize between 10-60 Mb/s of network bandwidth, with the vast majority of members of the cluster set utilize between 45-60 Mb/s of network bandwidth. Readers will appreciate that distributions may also be identified for other quantifiable characteristics of each cluster. Such quantifiable characteristics can include, for example, the failure rate of a cluster or particular components thereof, an identification of communication protocols used by a cluster or particular components thereof, an identification of the types of communications endpoints that a cluster or particular components thereof communicate with (e.g., endpoints that reside on the public internet v. endpoints that are in a private network), characteristics that can be classified by a binary value (e.g., does any component in the cluster perform privileged operations), information describing the various privileges that are given to a particular cluster, and many more.

[00695] In some embodiments, the distributions may be used to identify 'normal' behavior for a particular cluster. Consider an example in which the cluster set is identified, where each member of the cluster set represents a payroll system deployed in a particular customer's cloud deployment. In such an example, a distribution may be identified which indicates that all members of the cluster communicate with (and has privileged access to) the same set of cloud services, including: 1) a cloud database service (e.g., Amazon Aurora, Microsoft Azure SQL Database, Amazon Relational Database Service, Google Cloud SQL, Amazon DynamoDB), 2) a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00684

vendor provided SaaS offering that provides bill payment services, and 3) a vendor provided SaaS offering that provides accounting services. In such an example, by looking at each member of the cluster set and identifying that each member communicates with the same set of cloud services, a baseline may be established that identifies 'normal' behavior for each member of the cluster set, at least with respect to the specific characteristic (i.e., what cloud services are utilized by members) that the distribution is based on. As such, if monitoring a particular cluster revealed that some member of the cluster set accessed a source code repository cloud service (e.g., GitHub Enterprise on AWS), this sort of access would be outside of the typical distribution for this cluster set and could serve as the basis for raising an alert, denying access to the service, or initiating some other alerting/remediation workflow. Readers will appreciate that many distributions may be created for each cluster set, where each distribution is based on one or more characteristics of the members of the cluster set.

[00696] The example method depicted in Figure 5 also includes recommending 506, based on the normal behavior for one or more components 512 in one or more other cloud deployments 514, a change to the first cloud deployment 508. Recommending 506 a change to the first cloud deployment 508 may be carried out, for example, in response to determining that the normal behavior in one or more other cloud deployments 514 differs from the normal behavior for one or more components 510 in a first cloud deployment 508. In such an example, changes to the first cloud deployment 508 may be recommended that (if implemented) would cause the first cloud deployment 508 to be more similar to the other cloud deployments 514. For example, if the normal behavior in one or more other cloud deployments 514 indicates that all computing resources (e.g., virtual machines, container, serverless computing resources) communicate with each other using a particular secure data communications protocol and the normal behavior for one or more components 510 in a first cloud deployment 508 is for computing resources to communicate using some other data communications protocol, a change may be recommended that involves reconfiguring the computing resources to communicate using the particular secure data communications protocol.

[00697] Readers will appreciate that in some embodiments the mere fact that normal behavior in a first cloud deployment 508 deviates from normal behavior in one or more other cloud deployments 514 may be sufficient rationale for recommending 506 a change to the first cloud deployment 508. That is, a departure from normality and standard practices alone may result in recommending 506 a change to the first cloud deployment 508. In other embodiments, recommending 506 a change to the first cloud deployment 508 may only occur where the normal behavior in one or more other cloud deployments 514 is determined to be superior to the normal behavior in the first cloud deployment 508. For example, recommending 506 a change to the

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00685

first cloud deployment 508 may only be carried where the normal behavior in one or more other cloud deployments 514 is representative of a stronger security posture than the normal behavior in the first cloud deployment 508.

[00698] Consider an example in which a particular threat was detected (in part by detecting a deviation from normal behavior for one or more components 512) in a particular customer's cloud deployment 514, where the threat turned out to be a ransomware attack, which may in some embodiments include an encryption component and/or a data theft or leakage component. In such an example, if an identical (or sufficiently similar, following a general recognized pattern or 'fingerprint') threat is detected in the first customer's cloud deployment 508 (in part by detecting a similar deviation from deviation from normal behavior for one or more components 510), information describing the remedial actions (e.g., disabling encryption, increasing the frequency of backups, locking down a backup system, blocking transmission of data exernally, etc.) that were taken by the particular customer 514 may even recommended 506 as changes to be made to the first cloud deployment 508. Furthermore, if many customers had experienced the same attack and the data platform could determine with sufficient certainty that the first cloud deployment 508 was experiencing the same attack, workflows may be automatically initiated to carry out various remedial actions.

[00699] For further explanation, Fig. 6 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 6 is similar to the example depicted in Fig. 5, as the example depicted in Fig. 6 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

[00700] In some embodiments, customers (and their corresponding deployments) may be modeled into logical groups such that cross customer learning could be carried out only across customers in the same logical group, or other customers in the same logical group may be given a greater weighting for the purposes of cross customer learning. For example, a logical group may be tied to a customer's business function (e.g., the customer is a financial company, the customer is a health care company, the customer is a services company, the customer has inventory to manage, the customer sells to retail customers, and so on). Likewise, a logical group may be tied to a customer's cloud deployment (e.g., the deployment uses a particular combination of resources, the deployment uses a particular set of cloud-services, the deployment utilizes

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00686

availability zones and regions in a particular way, and so on). In such an example, a particular customer may be associated with multiple logical groups and cross customer learning for the particular customer may only involve other customers that are in the same (or sufficiently similar) logical groups. As such, the particular customer's resources (e.g., people, cloud deployment) may only be included in cluster sets with resources associated with other customers that are in the same (or sufficiently similar) logical groups.

[00701] To that end, the example method depicted in Fig. 6 also includes identifying 604, from the other cloud deployments, cloud deployments that are similar to the first cloud deployment 508. Identifying 604 cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by taking an inventory of the various components in each cloud deployment 508, 514 and identifying cloud deployments that have similar components, portions of the cloud deployment that have similar deployments, and so on. Alternatively, identifying 604 cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by identifying cloud deployments that are associated with customers in the same industry as the customer associated with the first cloud deployment 508, by identifying cloud deployments that are associated with a similar intended functionality as the first cloud deployment 508 (e.g., both cloud deployment are intended to provide an online store for an online retailer), and so on. In the examples in the preceding sentence, it may be assumed that two cloud deployments should be relatively similar even without inspecting the cloud deployments themselves. In such examples, similarity may be judge based on the application of rules, policies, heuristics, or similar mechanism. Alternatively, similarity may be detected via machine learning techniques as where information describing the activity of various components in many cloud deployments are fed as input to one or more machine learning models which subsequently identifies clusters across various cloud deployments that are similar.

[00702] In the example depicted in Fig. 6, recommending 506 the change to the first cloud deployment 508 may be based on the normal behavior for the cloud deployments that are similar to the first cloud deployment. Recommending 606 the change to the first cloud deployment 508 based on the normal behavior for the cloud deployments that are similar to the first cloud deployment 508 may be carried out, for example, by ignoring normal behavior for dissimilar cloud deployments and only taking into consideration those cloud deployments that have been identified 604 as being sufficiently similar to the first cloud deployment 508.

[00703] The example method depicted in Fig. 6 also includes identifying 602, from the other cloud deployments, one or more highly rated cloud deployments. A cloud deployment may be identified 602 as being 'highly rated' in the sense that fewer than a threshold number of vulnerabilities have been detected in the cloud deployment over some period of time, fewer than

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00687

a threshold number of outages (or less than a threshold amount of downtime) has been detected in the cloud deployment over some period of time, fewer than a threshold number of regulatory violations (or attempted violations) have been detected in the cloud deployment over some period of time, or identified 602 based on some other standard. In fact, comparisons may be relative (rather than comparing some quantifiable aspect of the other deployments operation) such that, for example, the 15% of cloud deployments with the fewest detected vulnerabilities over some period of time as identified 602 as being 'highly rated'. In other embodiments, identifying 602 one or more highly rated cloud deployments may be carried out through the use of machine learning techniques where information describing the activity of various components in many cloud deployments are fed as input to one or more machine learning models which subsequently identifies deployments that have the best performance against a combination of one or more metrics (e.g., vulnerabilities, availability, cost). In such a way, the best-of-breed cloud deployments may be identified 602.

[00704] In the example depicted in Fig. 6, recommending 506 the change to the first cloud deployment 508 may be based on the normal behavior for the highly rated cloud deployments. Recommending 608 the change to the first cloud deployment 508 based on the normal behavior for the highly rated cloud deployments may be carried out, for example, by ignoring normal behavior for cloud deployments that are not highly rated and only taking into consideration those cloud deployments are highly rated.

[00705] Readers will appreciate that although the example in Fig. 6 is depicted where identifying 604 cloud deployments that are similar to the first cloud deployment 508 and identifying 602 one or more highly rated cloud deployments are alternatives to each other, in other embodiments both steps may be performed. In such embodiments, recommending 506 the change to the first cloud deployment 508 may be based on the normal behavior other cloud deployments that are both highly rated and similar to the first cloud deployment 508.

[00706] For further explanation, Fig. 7 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 7 is similar to the examples depicted in Fig. 5 and Fig. 6, as the example depicted in Fig. 7 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or more other cloud deployments 514, and recommending 506 a change to the first cloud deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00688

[00707] The example method depicted in Fig. 7 also includes ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514. Ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514 may be carried out, for example, by identifying how well the cloud deployment 508 compares to other cloud deployments 514 with respect to one or more metrics. For example, cloud deployments may be ranked based on a weighted combination of multiple metrics (e.g., reliability, cost, regulatory compliance) such that the best-of-breed deployments may be identified. In such an example, the ranking of the first cloud deployment 508 relative to the other cloud deployments 514 may be presented to an administrator or other user associated with the first cloud deployment 508 for analysis by the administrator or other user.

[00708] The example method depicted in Fig. 7 also includes comparing 704 the trajectory of the first cloud deployment 508 to the trajectory of the other cloud deployments 514. The trajectory of a particular cloud deployment 508, 514 may be determined, for example, by evaluating the particular cloud deployment's 508, 514 performance over time as measured by one or more metrics. For example, the performance of each cloud deployment may be periodically scored based on a weighted combination of multiple metrics (e.g., reliability, cost, regulatory compliance). In such an example, the trajectory for a particular cloud deployment may be determined by determining the extent to which the score associated with a particular cloud deployment is changing over time. For example, if the first cloud deployment's 508 scoring is improving over time while the other cloud deployments' 514 scores are remaining the same over time, the trajectory of the first cloud deployment 508 may be determined to be better than the trajectory of the other cloud deployments 514 (in embodiments where a higher score is considered to be better).

[00709] Readers will appreciate that ranking 702 the first cloud deployment 508 relative to the other cloud deployments 514 and comparing 704 the trajectory of the first cloud deployment 508 to the trajectory of the other cloud deployments 514 are just two examples of the sort of analytics that can be performed by comparing the performance of multiple cloud deployments 508, 514. Readers will appreciate that other analytics may also be put in place due to the availability of information describing the performance of multiple cloud deployments 508, 514.

[00710] For further explanation, Fig. 8 sets forth a flowchart illustrating an additional example method of configuring cloud deployments based on learnings obtained by monitoring other cloud deployments in accordance with some embodiments of the present disclosure. The example depicted in Fig. 8 is similar to the examples depicted in Figs. 5-7, as the example depicted in Fig. 8 also includes determining 502 normal behavior for one or more components 510 in a first cloud deployment 508, determining 504 normal behavior for one or more components 512 in one or

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00689

more other cloud deployments 514, and recommending 506 a change to the first cloud deployment 508 based on the normal behavior for one or more components 512 in one or more other cloud deployments 514.

[00711] The example method depicted in Fig. 8 also includes identifying 802 other cloud deployments 514 to exclude from consideration when recommending changes to the first cloud deployment 508. Readers will appreciate that other cloud deployments 514 may be excluded from consideration when recommending changes to the first cloud deployment 508 for a variety of reasons. For example, another cloud deployment 514 may be so dissimilar to the first cloud deployment 508 that there may be very little to learn that is relevant to the first cloud deployment 508, another cloud deployment 514 may be deficient for a variety of reasons such that there may be very little high quality takeaways from evaluating the other cloud deployment 514 that are relevant to the first cloud deployment 508, another cloud deployment 514 may be a relatively recent deployment such that any takeaways from evaluating the other cloud deployment 514 may not be reliable and proven over time, or for some other reason. In such an example, identifying 802 other cloud deployments 514 to exclude from consideration when recommending changes to the first cloud deployment 508 may be carried out by filtering the set of other cloud deployments 514 according to some criteria, or carried out in some other way. In the example depicted in Fig. 8, recommending 506 the change to the first cloud deployment 508 may therefore include recommending 806 changes based on the other cloud deployments 514 that are not excluded from consideration.

[00712] The example method depicted in Fig. 8 also includes identifying 804 a response to a recommended change in one or more other cloud deployments 514. Identifying 804 a response to a recommended change in one or more other cloud deployments 514 may be carried out, for example, by tracking the status of a recommended change that was recommended for another cloud deployment 514 to determine whether the recommended change was actually implemented. Consider an example in which a recommendation was made to have another cloud deployment 514 switch from using a first IaaS offering to a second IaaS offering in response to detecting some condition in the cloud deployment 514. In such an example, identifying 804 a response to a recommended change in one or more other cloud deployments 514 may be carried out by determining whether the recommended change was actually implemented and a switch was made to utilize the second IaaS offering as part of the cloud deployment 514. Readers will appreciate that by repeatedly identifying 804 responses to recommended changes in one or more other cloud deployments 514 can be used by the systems described above as an indication as to the systems described above are making useful recommendations. For example, if identifying 804 a response to a recommended change in one or more other cloud deployments 514 reveals that a very small

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00690

percentage of administrators actually implement the recommended change, this may indicate that the recommend change is not particularly valuable and may be used as feedback in determining whether to make the recommended change if the condition that triggered the recommendation is encountered again. Alternatively, if identifying 804 a response to a recommended change in one or more other cloud deployments 514 reveals that a very large percentage of administrators actually implement the recommended change, this may indicate that the recommend change is valuable and may be used as feedback in determining whether to make the recommended change if the condition that triggered the recommendation is encountered again. In such a way, the response from other customer's may be used to drive recommendations that are made regarding the first cloud deployment 508.

[00713] In the example depicted in Fig. 8, recommending 506 the change to the first cloud deployment 508 recommending 808 the change to the first cloud deployment 508 based on the response to the recommended change in one or more other cloud deployments 514. Recommending 808 the change to the first cloud deployment 508 based on the response to the recommended change in one or more other cloud deployments 514 may be carried out, for example, only if a predetermined percentage of previous recipients of similar recommendations have actually implemented the recommended change, as described above.

[00714] Readers will appreciate that while the examples described in the preceding two paragraphs relate to an example in which the response to a recommended change in one or more other cloud deployments 514 is expressed in terms of whether the recommended change was or was not implemented, in other embodiments the response to a recommended change in one or more other cloud deployments 514 may be measured in other ways. For example, a response to a recommended change in one or more other cloud deployments 514 may represent whether the recommended change did or did not resolve the condition that caused the recommendation to be generated in the first place. In other embodiments, the response to a recommended change in one or more other cloud deployments 514 may be measured or determined using different criteria.

[00715] Although some of the figures described above depict only a single other cloud deployment 514, a single other cloud deployment 514 is included for ease of illustration but in no way represents a limitation of the embodiments described herein. In fact, most embodiments will include multiple other cloud deployments 514 that may be monitored and learned from.

[00716] Readers will appreciate that while many of the embodiments described above relate to embodiments where cross-customer learnings are used to generate recommendations for changes to the first cloud deployment 508, in other embodiments the cross-customer learning may be used to initiate, based on the normal behavior for the one or more other cloud deployments, a change to the first cloud deployment. Initiating such changes may be carried out, for example, by

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00691

initiating one or more remediation workflows that implement the changes. As was the case with recommendations, initiating a change to the first cloud environment may be based on the normal behavior for similar cloud deployments, based on the normal behavior for the highly rated cloud deployments, based on the cloud deployments to exclude from consideration when recommending changes to the first cloud deployment, and so on.

[00717] For further explanation, Fig. 9 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 9 may be carried out by the systems described above (also referred to as a 'data platform' above). As such, the systems described above may include computer program instructions executing on computer hardware, virtualized hardware, or some other execution environment (e.g., one or more containers, one or more serverless compute instances), where the computer program instructions carry out the steps described in Fig. 9 when the computer program instructions are executed.

[00718] The example depicted in Fig. 9 includes a plurality of cloud deployments, including a first cloud deployment 910 and one or more additional cloud deployments 912, which in this illustration includes three cloud deployments 914a, 914b, 914n, although more or fewer cloud deployments may be included in the set of additional cloud deployments 912 in other embodiments. Such cloud deployments may be embodied as described above and may include a variety of components such as, for example, software applications, storage resources, computing resources, networking resources, and other resources. In each cloud deployment, each of the resources may be delivered as services provided by a public cloud, private cloud, hybrid cloud, and so on.

[00719] The example method depicted in Fig. 9 includes identifying 902, for at least a portion of a first cloud deployment 910, one or more additional cloud deployments 912 to utilize for cross-customer learning. Cross-customer learning, as the phrase is used here, can generally be described as the process of learning about the cloud deployments of different customers, different organizations, or some other entity for the benefit of tailoring another entity's cloud deployment. For example, the cloud deployments of Customer A, Customer B, Customer C, Customer D, and Customer F may be evaluated and the information gathered about their deployments may be leveraged to help shape the cloud deployment of Customer F. In such an example, lessons learned by other customers may be used to help guide a particular customer's design, deployment, and management of their cloud deployments, which may be particularly useful for organizations that are pivoting to the cloud from on-premises based deployments.

[00720] Identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may be carried out, for example, by identifying cloud deployments for other

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00692

customers that are in similar industries as the customer associated with the first cloud deployment 910, by identifying highly rated (as described above) cloud deployments, by identifying cloud deployments (or portions thereof) that are similar to the first cloud deployment 910 (or some portion thereof), or in some other way. In some embodiments, identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may be carried out using machine learning techniques as one or more machine learning models may be used to identify cloud deployments that are similar to the first cloud deployment 910, to identify best-of-breed cloud deployments, and so on. In such a way, identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning may result in a subset of curated, relevant, and/or exemplary cloud deployments being utilized for cross-customer learning rather than using the entire set of available cloud deployments being utilized for cross-customer learning (although some embodiments could utilize all cloud deployments for cross-customer learning).

[00721] The example method depicted in Fig. 9 also includes receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912. The one or more actions associated with the additional cloud deployments 912 may be embodied as actions taken by components within the additional cloud deployments 912. For example, a particular software service accessing a data store may be an example of an action that is associated with the additional cloud deployments 912, especially where the software service and the data store are components of a particular additional cloud deployment 914a. Likewise, a message, request, or other form of data communications that are exchanged between the additional cloud deployments 912 and actors that are external to the additional cloud deployments 912 may be an example of an action that is associated with the additional cloud deployments 912. For example, a server that is external to any of the additional cloud deployments 912 may send requests to access some software service within a particular additional cloud deployment 914a.

[00722] In other embodiments, the one or more actions associated with the additional cloud deployments 912 may include one or more user interactions with the additional cloud deployments 912. Such user interactions can include a description of which resources are accessed and/or utilized by particular users, particular business units (e.g., the finance department of a business, the engineering department of a business), particular personas (e.g., a system administrator, a software developer, a human resources manager), and so on. The one or more actions associated with the additional cloud deployments 912 may also include, for example, one or more interactions involving external applications, external computing devices, or similar external actor and the additional cloud deployments 912. Such user interactions can include, for example, a description of which resources within the additional cloud deployments 912 are accessed and/or utilized by particular external application, what IP address is associated with an

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00693

external entity that is communicating with the additional cloud deployments 912, a description of the specific requests that an external actor is issuing to the resources within the additional cloud deployments 912, and so on.

[00723] In the example method depicted in Fig. 9, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 may be carried out by receiving (directly or indirectly) such information 916 from the additional cloud deployments 912 themselves. In fact, the additional cloud deployments 912 may be queried for such information. In other embodiments, the information 916 describing one or more actions associated with the additional cloud deployments 912 may be retained by the systems described above as part of monitoring the additional cloud deployments 912. In such an example, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 may be carried out by querying a data repository (e.g., a data warehouse) that contains such information that was gathered while monitoring the additional cloud deployments 912. Readers will appreciate that information 916 may be received 904 describing other types of actions associated with the additional cloud deployments 912.

[00724] The example method depicted in Fig. 9 also includes receiving 906 information 918 describing configurations associated with the additional cloud deployments 912. The information 918 describing configurations associated with the additional cloud deployments 912 may be embodied, for example, as information describing how components within a cloud deployment are organized (including which other components they may communication with), information describing what permissions are granted to various users, information describing the manner in which internal (i.e., occurring entirely within the cloud deployment) data communications and/or external (i.e., occurring at least partially external to the cloud deployment) data communications are carried out, and so on. Such information 918 may be received 906 in a similar manner as described above with respect to step 904.

[00725] The example method depicted in Fig. 9 also includes identifying 908, based on the configurations and the one or more actions, one or more configurations to adopt for the first cloud deployment 910. Readers will appreciate that by identifying 908 one or more configurations to adopt for the first cloud deployment 910 in such a way, the configuration of the first cloud deployment 910 may be influenced by the monitoring of the additional cloud deployments 912. In such an example, once the configurations that the first cloud deployment 910 should adopt have been identified 908, the first cloud deployment 910 (or associated entities) may be reconfigured to implement such configurations.

[00726] Identifying 908 one or more configurations to adopt for the first cloud deployment 910 may be carried out, for example, by identifying configurations in the additional cloud

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00694

deployments 912 that were effective in dealing with various threats, securing vulnerabilities, or otherwise contributed to a healthy cloud deployment. Some configurations associated with the additional cloud deployments 912 may have been put in place in response to some action. For example, a configuration to always backup data stored in a first storage service (e.g., AWS S3) to a second, distinct storage service (e.g., AWS Glacier) may have been put in place in response to some attack that was directed to a particular cloud deployment's 914a S3 buckets. In such an embodiment, if the first cloud deployment 910 also leverages S3 as its object store, it may be desirable for the first cloud deployment 910 to adopt a configuration setting that would result in its S3 buckets being backed up to Glacier so that the first cloud deployment 910 can be ready to survive a similar attack that was experienced by the particular cloud deployment 914a.

[00727] For further explanation, Fig. 10 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 10 is similar to the example depicted in Fig. 9, as the example depicted in Fig. 10 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

[00728] In the example method depicted in Fig. 10, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 can include receiving 1002 information describing a security threat to one or more of the additional cloud deployments 912. The security threat may be embodied, for example, as a ransomware attack that was directed to one or more of the additional cloud deployments 912, as a denial of service attack that was directed to one or more of the additional cloud deployments 912, as an SQL injection attack that was directed to one or more of the additional cloud deployments 912, or as some other security threat. The information describing a security threat to one or more of the additional cloud deployments 912 can include, for example, information describing where an attack originated from, information describing the components within the cloud deployment that were targeted by an attack, information describing data access patterns that were associated with an attack, or some other information that would be useful in detecting subsequent security threats.

[00729] In the example method depicted in Fig. 10, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can include receiving 1004 information describing configuration settings used to combat the security threat. The configuration settings used to combat the security threat may be embodied, for example, as configuration settings that caused the additional cloud deployments 912 to blacklist certain IP

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00695

addresses or network domains, as configuration settings that caused the additional cloud deployments 912 to close vulnerabilities that allowed an attack to succeed, or some other configuration that would be useful in detecting/preventing/mitigating a security threat.

[00730] In the example method depicted in Fig. 10, receiving 904 information 916 describing one or more actions associated with the additional cloud deployments 912 can also include receiving 1006 information describing a detected vulnerability associated with one or more of the additional cloud deployments 912. The detected vulnerability may be embodied, for example, as a vulnerability to some known security threat, as a vulnerability to some data loss event, as a data breach vulnerability, or as some other vulnerability. The information describing a detected vulnerability of one or more of the additional cloud deployments 912 can include, for example, information describing vulnerable components, information describing data communications protocols, endpoints, or other data communications components that expose vulnerabilities, or some other information that would be useful in detecting a vulnerability.

[00731] In the example method depicted in Fig. 10, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can also include receiving 1008 information describing configuration settings used to address the vulnerability. The configuration settings used to address the vulnerability may be embodied, for example, as configuration settings that caused the additional cloud deployments 912 to blacklist certain IP addresses or network domains, as configuration settings that caused certain components within the additional cloud deployments 912 to not connect to external data communications networks, as configuration settings that caused certain components to be accessible only by using enhanced authentication protocols, or as some other configuration that would be useful in detecting, preventing, or otherwise mitigating a vulnerability.

[00732] For further explanation, Fig. 11 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 11 is similar to the examples depicted in Fig. 9 and Fig. 10, as the example depicted in Fig. 11 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

[00733] In the example method depicted in Fig. 11, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912 can include receiving 1102 information describing permissions for one or more users of the additional cloud deployments 912. The one or more users of the additional cloud deployments 912 may be embodied, for example, as individual users (e.g., Bob Smith), as a collection of user (e.g., Finance Group,

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00696

Marketing Group), and so on. The one or more users of the additional cloud deployments 912 may also be embodied as different personas within an organization, where each persona is associated with a certain role. For example, there may be 'software developer' personas, 'database administrator' personas, 'customer support' personas, and many others. Receiving 1102 information describing permissions for one or more users of the additional cloud deployments 912 may be carried out, for example, by receiving (directly or indirectly) information from one or more of the additional cloud deployments 912 that described what resources a particular user/persona/group can access, what level of privileges they have with respect to a particular resource, and any other information describing the user/persona/group's privileges. In fact, such information may be retained in a data warehouse during monitoring of the additional cloud deployments 912 and received 1102 by querying such a data warehouse.

[00734] The example method depicted in Fig. 11 also includes determining 1104, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment 910 are over-permissioned. A particular user may be over-permissioned in the sense that the user has privileges or permissions beyond those that should be associated with the user. Consider the example of two distinct personas within an organization: 1) a software engineer, and 2) an accounts payable administrator. In such an example, the software engineer might have sufficient privileges to access a code repository, a dev/test environment, a code documentation tool, and so on. The accounts payable administrator, however, might have sufficient privileges to access bill paying software, a financial ledger, and so. The software developer having access to bill paying software or the accounts payable administrator having access to a code repository may be examples of the users being over-permissioned, as the user has access to resources that they should not be able to access. Such situations can create vulnerabilities as an over-permissioned user (or a malicious actor using the over-permissioned user's credentials) may intentionally or unintentionally perform some harmful act that they would not be capable of performing if they weren't over-permissioned.

[00735] Determining 1104 that one or more users of the first cloud deployment 910 are over-permissioned may be carried out, for example, by examining the information describing permissions for one or more users of the additional cloud deployments 912 to determine whether similar users in the additional cloud deployments 912 have different/fewer privileges. As such, a user in a first cloud deployment 910 may be determined to be over-permissioned if other cloud deployments 912 are giving similar users fewer permissions than such a user has in the first cloud deployment 910.

[00736] In the example method depicted in Fig. 11, identifying 908 one or more configurations to adopt for the first cloud deployment 910 can include identifying 1106 a reduced privilege level

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00697

to give to the one or more users. Identifying 1106 a reduced privilege level to give to the one or more users may be carried out, for example, by giving an over-permissioned user of the first cloud deployment 910 privileges that are similar to those held by similar users in the additional cloud deployments 912.

[00737] For further explanation, Fig. 12 sets forth a flowchart illustrating an example method of learning from similar cloud deployments in accordance with some embodiments of the present disclosure. The example method depicted in Fig. 12 is similar to the examples depicted in Figs. 9-11, as the example depicted in Fig. 12 also includes identifying 902 one or more additional cloud deployments 912 to utilize for cross-customer learning, receiving 906 information 918 describing configurations associated with the additional cloud deployments 912, and identifying 908 one or more configurations to adopt for the first cloud deployment 910.

[00738] The example method depicted in Fig. 12 also includes receiving 1202 information 1204 describing one or more deployment processes associated with the additional cloud deployments 912. The information 1204 describing one or more deployment processes associated with the additional cloud deployments 912 can include, for example, information describing how a particular component within an additional cloud deployment was created (e.g., who requested that the component be created, when was the request to create the component issued). In addition, in some embodiments such information 1204 can include information describing the software development processes that were used to develop software that will be deployed in a particular cloud deployment (e.g., was a code repository used, who has access to the repository, who committed the last change to some code before deployment). In such a way, the information 1204 describing one or more deployment processes associated with the additional cloud deployments 912 can describe the processes that are being implemented in other cloud deployments. Such information 1204 may be used, for example, to determine whether the development and deployment processes associated with the first cloud deployment 910 are abnormal, to use as the basis for making recommended changes to the development and deployment processes associated with the first cloud deployment 910, to initiate remediation workflows to back out changes to the first cloud deployment 910 that deviated from normal activity or violated some policy, and so on.

[00739] The example method depicted in Fig. 12 also includes identifying 1206 abnormally configured components in the first cloud deployment 910. Abnormally configured components in the first cloud deployment 910 may be embodied as components that are configured in a way that deviates from typical configurations observed by monitoring the additional cloud deployments 912. Monitoring the additional cloud deployments 912 may reveal, for example, that a connection to a server that is contained in another cloud deployment 914a should always

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00698

go through an authentication server that is contained in another cloud deployment 914a. In such an example, if an examination of the first cloud deployment 910 reveals that an authentication server has been bypassed by a connection to a server, this may be an indication of a breach or misconfiguration regardless of customer's network topology. In such an example, the component (i.e., the server) may be identified 1206 as being abnormally configured based on its deviation from typical configurations observed in the additional cloud deployments 912.

[00740] Although the examples described above relate to embodiments where one or more configurations to adopt for the first cloud deployment 910 are identified 908 using the information described above, in other embodiments such information may be used for other purposes. For example, information gained by observing additional cloud deployments 912 may be leveraged to identify best-of-breed deployments, similar deployments, and so on.

[00741] In some embodiments, the distributions that identify 'normal' behavior for a particular cluster set may be used for a variety of purposes. For example, anomaly detection may be performed by identify members of the cluster set that are operating outside of a particular typical distribution. Likewise, best practices may be identified using distributions and members of the cluster set that are not adhering to best practices may be identified if they are operating outside of a particular typical distribution. Vulnerabilities may also be identified using distributions, for example, by identifying members of the cluster set that have over-privileged users or components that may be able access things that they should not be able to access. Readers will appreciate that that deviation from an established baseline of normal behavior, normal activity, normal configuration, or other form of normal operation may be indicative of many other things, all of which may be detected through the usage of the distributions described above paired with the monitoring of particular clusters.

[00742] In some embodiments, the distributions that identify 'normal' behavior for a particular cluster set may be used to not only detect threats, vulnerabilities, compromise, and things of that nature, but the distributions that identify 'normal' behavior for a particular cluster set may be used to optimize the operation or configuration of a particular cluster. For example, cloud deployments that utilize fewer resources may be examined and characteristics that are common across such the cloud deployments may be identified as being high efficiency characteristics. Likewise, cloud deployments that are the subject of fewer alerts may be examined and characteristics that are common across such the cloud deployments may be identified as being high efficiency characteristics.

[00743] In some embodiments, historical information may also be retained and utilized to show the trajectory of a particular one or more clusters. For example, historical information may be used to compare the current state of a particular cluster (as measured by one or more quantifiable

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00699

characteristics associated with the cluster) with a previous state of the cluster such that trends and/or trajectories may be identified. Consider an example in which a particular characteristic associated with a cluster identifies how many users are accessing the cluster. Historical information associated with the cluster may be compared to current (or most recent) information associated with the cluster to determine, for example, that more or less users are accessing the cluster, that the number of users that are accessing the cluster is increasing or decreasing at a certain rate, and so on. In some situations, some changes may be acceptable (e.g., if the cluster represents an interface to an online store, and increase in the number of users accessing the online store may be perfectly acceptable given an expansion in the organization's customer base) whereas in other situations changes may be troubling (e.g., if the cluster represents a source code repository and more users are accessing the repository in spite of a contraction in the number of developers that are employed by an organization) and my require alerts, further investigation, or some other remediation workflow.

[00744] In some embodiments, historical information may also be retained and utilized to show the trajectory of a particular cluster relative to other members of a cluster set. For example, historical information may be used to compare the failure rate of particular cluster (as measured by one or more quantifiable characteristics associated with the cluster) over time with the failure rate over time of other clusters in the cluster set. Such a comparison may reveal, for example, that a particular cluster was previously failing at a rate that was in line with the failure rate of other members of the cluster set, but that the particular cluster has more recently been failing more/less than other members of the cluster set. As such, through the use of such techniques a determination may be made as to whether a particular cluster is becoming more/less healthy than other members of the cluster set, more/less secure than other members of the cluster set, more/less efficient than other members of the cluster set, more/less reliable than other members of the cluster set, more/less compliant with relevant regulations than other members of the cluster set, slower/faster than other members of the cluster set, and so on. Such an analysis may be part of identifying best-of-breed deployments, best practices, providing remediation actions, providing recommendations, or utilized for a variety of other purposes.

[00745] Although the examples described above relate to embodiments where components within a cloud deployment are analyzed, monitored, or otherwise observed, in other embodiments the techniques described herein may be applied to other entities. For example, the techniques described above may be applied to analyze, monitor, or otherwise observe different personas with an organization, different users with an organization, different user groups with an organization, and so on. Using such techniques, 'normal' behaviors for a particular persona (e.g., a database administrator, a network administrator) can be identified, 'normal' behaviors for a

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00700

particular user group (e.g., users that are part of an organization's finance department, users that are part of an organization's engineering department) can be identified, and so on.

[00746] In some embodiments, by identifying 'normal' behaviors (i.e., those behaviors that are consistent with standard distributions for particular characteristics associated with an entity) with different personas, different users, different user groups, or other entities, abnormal behavior may also be identified. Consider an example in which a group of users is identified as a cluster by virtue of those users accessing the same set of applications (e.g., all of the users access a set of applications that are finance-related applications). Further assume that similar clusters are identified for other customers, such that a cluster set may be formed. In such an example, a distribution may be identified for the set of applications that members of the cluster typically access. For example, each member of the cluster set may typically access an accounting application, a spreadsheet application, a payroll application, and so on. If one member of the cluster set consists of one or more users that also access an organization's source code repository, such behavior may be determined to be outside of the typical distribution for members of the cluster set. In response to making some determination, alerts may be generated, access to the source code repository may be blocked for the users in the cluster, a remediation workflow may be initiated, or some other action may be taken.

[00747] In some embodiments, the techniques described above may be particularly useful for identifying over-privileged users, user groups, personas, or other entity. Identifying over-privileged users, user groups, personas may be carried out, at least in part, by identifying what set of privileges is 'normal' for a particular entity to have based on evaluating what privileges are given to similar or identical members of a cluster set. That is, an evaluation may be made as to what privileges are given to users, user groups, personas, or other entity by one or more other customers. If the set of privileges given to a particular customer's users, user groups, personas, or other entity are not consistent with (e.g., the set of privileges are much greater than) the privileges given to similar or identical users, user groups, personas, or other entity of another customer (i.e., the other members of the cluster set), a determination may be made that the particular customer's users, user groups, personas, or other entities are over-privileged. In some embodiments, an evaluation as to whether a user, user group, persona, or other entity is over-privileged may also include identifying privileges that the user has and comparing that with the privileges that the user actually utilizes. If some privileges are never used, this may be taken as an indication that the user is over-privileged.

[00748] In some embodiments, the cross customer techniques described above may be used to provide additional context to issues identified in a particular customer's cloud deployment, to provide recommendations to a particular customer, or even to drive remediation actions.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00701

Consider an example in which a particular threat was detected in a first customer's cloud deployment, where the threat turned out to be a ransomware attack, which may in some embodiments include an encryption component and/or a data theft or leakage component. In such an example, if an identical (or sufficiently similar, following a general recognized pattern or 'fingerprint') threat is detected in a second customer's cloud deployment, additional context may be provided by including information in an alert that is delivered to the second customer that indicates that the threat matches the profile of a ransomware attack that was detected in the first customer's cloud deployment. In fact, information describing the remedial actions (e.g., disabling encryption, increasing the frequency of backups, locking down a backup system, blocking transmission of data externally, etc.) that were taken by the first customer may even be included in the alert to the second customer or otherwise recommended to the second customer. Furthermore, if many customers had experienced the same attack and the data platform could determine with sufficient certainty that the second customer's cloud deployment was experiencing the same attack, workflows may be automatically initiated to carry out various remedial actions.

[00749] In some embodiments, the way other customers investigated or responded to a particular alert may also be used when presenting alerts to a particular customer. Consider an example in which a particular threat was detected in ten different customer's cloud deployments. In such an example, assume that 9 of the 10 customers ignored the alert. In such an example, the conditions that are indicative of the particular threat are detected in a particular customer's cloud deployment, when raising the alert to the particular customer, information may be included in the alert indicating that most other customers ignored the alert. In such an embodiment, it is the behavior of users of the data platform – not components in a cloud deployment – that is being monitored and evaluated for the benefit of other customers. In other embodiments, information indicating that customers do or do not ignore a particular alert may be used when scoring or ranking the alert. For example, alerts that are ignored by most customers may be ranked as less critical than alerts that are acted upon by most customers. In addition, the particular security stance of a particular customer may be utilized when determining the extent to which their usage of the data platform should be utilized when guiding other customers. If a first customer has relatively poor security practices, for example, the first customer's usage of the data platform may be ignored (or given less weight) for the purposes of guiding other customers. If a second customer has relatively good security practices, however, the second customer's usage of the data platform may be taken into consideration (or given more weight) for the purposes of guiding other customers. Likewise, if users that ignored the alert experienced a security breach while users that did not ignore the alert did not experience a security breach, such an outcome could be

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00702

taken into consideration when ranking the alert, determining what information to include in the alert, and so on.

[00750] In addition to using other customers interactions with the data platform to score, rank, suppress, or provide context for the alerts provided to other customers, the remediation actions taken by other customers may also be used for guiding a particular customer. Consider an example in which a particular threat was detected in ten different customer's cloud deployments. In such an example, assume that 9 of the 10 customers investigated the threat by accessing a particular customer support page. In this example, if the conditions that are indicative of the threat are detected in a particular customer's cloud deployment, information may be included in an alert indicating that most other customers that received the alert investigated the alert by accessing the particular customer support page. Likewise, if other customers successfully resolved the issue, information may be included in an alert indicating the solution that was implemented by other customers.

[00751] In some embodiments, the way customers investigated or responded to a particular alert may be used with other information to determine whether the customer's responses were actually correct. Consider an example in which an alert is sent to 10 customers identifying a vulnerability that would allow crypto miners to use the customer's resources for solving complex computational problems for the purposes of acquiring cryptocurrency. In such an example, assume that 9 of 10 customers ignored the alert. Further assume that in this example, the 9 customers that ignored the alert subsequently had their cloud resources utilized by hackers for crypto mining, whereas the 1 customer that did not ignore the alert did not suffer such an attack. In such an example, the data platform described above should not cease issuing alerts when detecting this vulnerability by virtue of 9 of 10 customers ignoring the alert. Instead, by taking the ultimate outcome for each customer into consideration, rather than suppressing the alert, the data platforms should respond by taking actions that would make it more likely that customers would not ignore these alerts. For example, the severity level of the alert may be raised, a user of the data platform may be required to confirm receipt of the alert, contextual information could be included in the alert indicating that recipients that do not respond to the alert end up having their systems hijacked by crypto miners, or some other action may be taken.

[00752] In some embodiments, a customer's interactions may be analyzed to improve how alerts are delivered to a customer. For example, if an evaluation of customer interactions with the data platform indicates that most customer ignore alerts issued between 11:00 PM – 7:00 AM whereas most customers take action in response to alerts issued between 7:01 AM – 10:59 PM, then alerts that are issued between 11:00 PM – 7:00 AM may be reissued between 7:01 AM – 10:59 PM. Likewise, if a particular class of alerts (e.g., those related to vulnerability threats) are largely

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00703

acted upon whereas another class of alerts (e.g., those related to a lack of compliance with regulatory requirements) are largely ignored, the manner in which alerts are generated may be altered. For example, alerts that are related to a lack of compliance with regulatory requirements may be issued to additional users, such as some user that is designated as having a compliance persona within a customer's organization. In other embodiments, other aspects of the customer's interactions with the data platform may be used to improve the manner in which the data platform interacts with customers (e.g., what type of devices received alerts, were alerts delivered in a primary or secondary window, do the parties that the alerts were sent to have other obligations at the time of alert as determined from an inspection of their calendar, and so on). While the example described above related to embodiments where the interactions of multiple customers with the data platform is analyzed, in other embodiments a single customer's interaction with the date platform may serve as the basis for altering the way that the data platform interacts with the customer.

[00753] In some embodiments, the same or additional cross customer learning techniques may be applied to earlier stages of a software development pipeline and even before an actual cloud deployment is in place for a particular customer. Stated differently, the same or additional cross customer learning techniques may be applied to things other than a deployed system. For example, cross customer learning may be applied to development processes, testing processes, deployment processes, and so on

[00754] Readers will appreciate that developing, testing, and deploying software in cloud environment comes with a few requirements that were not always present in software development. In particular, software is always expected to be running but the software is also expected to continue to be updated with fixes, new features, or other improvements. As such, it is not desirable to cease running a software application, install a new version, and begin running the new version. As a result of these changes to the software development paradigm, the software development processes, software testing and validation processes, and deployment processes tend to be fairly independent of each other. In some embodiments, each of these processes may be analyzed, monitored, or otherwise observed by the data platforms described herein. Because each of these processes are analyzed, monitored, or otherwise observed by the data platforms described herein, an opportunity exists to leverage cross customer learning to these processes.

[00755] In some embodiments, the development processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. For example, interactions with a code repository may be monitored, permissions granted to each user of the code repository may be monitored, the number of people that are checking in and checking out

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00704

code may be monitored, the extent to which code revisions are documented may be monitored, and so on. In such an example, the development processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

[00756] In some embodiments, the testing processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. For example, the number of tests that are run may monitored, the type of tests that are run may be monitored, the number of people that are running tests and evaluating test results may be monitored, the processes through which the results of testing are communicated with developers may be monitored, and so on. In such an example, the testing processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

[00757] In some embodiments, the results of testing may be carried forward and utilized when an application is deployed. Consider an example in which testing reveals that the code performs some action that appears to create a possible vulnerability, but further testing and validation reveals that a vulnerability is not created. When this piece of code is subsequently deployed, it presumably will perform the same action (i.e., the action that was flagged during testing) that appears to create a possible vulnerability. Rather than raising an alert, initiating some investigative action, initiating a remedial action, or responding in a similar way, by carrying forward the knowledge gained during testing (i.e., that a vulnerability was not, in fact, created) the possible vulnerability may be ignored based on the conclusions reached during testing.

[00758] In some embodiments, the deployment processes for a particular customer may be analyzed, monitored, or otherwise observed by the data platforms described herein. In such an example, the deployment processes for a particular customer may be learned and compared to normal behavior for other customers. Through such comparisons, inefficiencies may be identified, vulnerabilities may be identified, and other shortcomings may be identified. In response to the identification of inefficiencies, vulnerabilities, or other shortcoming, alerts may be issued, remediation workflows may be initiated, or some other action may be taken.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00705

[00759] In some embodiments, some forms of static analysis may be used (in conjunction with other features of the data platform) to detect anomalies, vulnerabilities, threats, misconfigurations, violations of regulatory requirements, and many other things. Consider an example in which a cloud deployment is deployed using IaC. In such an example, one or more configuration files may be examined, and the state of the cloud deployment may be monitored to identify situations in which the state of the cloud deployment drifts from the configuration of the cloud deployment that was described in the configuration file. As such, the intended configuration of the cloud deployment (at least as expressed in one or more configuration files) may be used as a baseline to measure the current cloud deployment, such that alerts may be issued or other remediation workflows may be initiated when a customer's cloud deployment deviates from its codified state.

[00760] In some embodiments, other forms of code other than an IaC configuration may be examined to detect anomalies, vulnerabilities, threats, misconfigurations, violations of regulatory requirements, and many other things. For example, the source code that has been deployed in a customer's environment may be examined to determine all the things that the code could do. This information may be compared to a polygraph for the customer's cloud deployment, which identifies all things that a customer's cloud deployment does do, as learned by monitoring and observing the customer's cloud deployment. Consider an example in which a polygraph for a customer's cloud deployment indicates that a first microservice in their cloud deployment only communicates with other internal microservices. In such an example, however, assume that the source code for the first microservice includes a messaging library that it uses to communicate with the other internal microservices. Further assume in this example, however, that examining the source code for the first microservice reveals that the messaging library also includes functions that enable a user of the library to send messages to recipients on an external network using standard internet protocols (e.g., TCP/IP, HTTPS, and so on). In such an example, although the polygraph for the customer's cloud deployment reveals that the first microservice only communicates with other internal microservices, the presence of functions that could be used for communications with external services, machines, and other entities may be undesirable. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if executed, would rise to the level of anomalous or otherwise unusual activity. For example, the data platform may prompt the customer to delete or disable the library functions described above that enable undesirable data communications.

[00761] In another example, assume that some source code has all the necessary capabilities (e.g. hashing functions, mathematical calculations, etc.) to mine cryptocurrency, as the mining process

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00706

may only require performing standard computations. During a crypto miner attack, however, the source code may be subverted such that the code runs in a loop to perform the standard computations required for mining cryptocurrency. An evaluation as to what functions the source code can perform may therefore not reveal anything concerning but monitoring the actual operation of the source code may reveal that the source code is operating in an undesirable way. Likewise, static analysis may reveal things like a non-incrementing counter or other mechanism that would result in an infinite loop or similar operation. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if or when executed, rise to the level of anomalous or otherwise unusual activity.

[00762] In another example, assume that some source code has all the necessary capabilities to carry out the steps required for a ransomware attack, as the ransomware attack may only require performing standard operations like reading data, encrypting data, sending data, communicating externally, and so on. During a ransomware attack, however, the source code may be subverted such that the code does far more encryption than would be expected in a typical code module. An evaluation as to what functions the source code can perform may therefore not reveal anything concerning but monitoring the actual operation of the source code may reveal that the source code is operating in an undesirable way. Likewise, static analysis may reveal that all paths through the source result in data being encrypted, or that all paths through the source code results in some atypical pattern (e.g., equal amounts of reading data, encrypting data, and writing data, or unusual external data flows), which may rise to the level of being unusual or concerning behavior. As such and in accordance with some embodiments of the present disclosure, the data platform may be configured to alert a customer or initiate some other workflow upon detecting that source code includes features that, if or when executed, rise to the level of anomalous or otherwise unusual activity.

[00763] In some embodiments, the data platforms described above may be used to identify discriminates between two entities that may otherwise appear to be similar or identical. In the examples described herein, 'discriminates' may be embodied as characteristics of entities that, when not similar or identical, prevent the entities from being similar or identical. For example, many customers may have many Java processes. If the data platform were to identify the set of things that each Java process does for each customer and identify that set of things as the 'normal' or acceptable set of things that any Java process was allowed to do, this set would be far too large. As such, the data platform may be configured to look at things like command line arguments and know that one or more Java processes with one set of jar files and command line arguments is actually a separate program from one or more Java processes with another set of jar

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00707

files and command line arguments. In other words, the data platform may identify discriminates (in the example, a first discriminate being distinct command line arguments and a second discriminate may be that the jar files for each process are distinct) between two sets of Java processes to determine that the Java processes are not actually similar or identical entities. As such, any attempt to engage in cross customer learning based on a first set of Java processes in a first customer's environment and a second set of Java processes in a second customer's environment may result in undesirable outcomes. Through the use of discriminates, however, a decision can be reached that these two sets of Java processes are not related and attempts to learn through an examination of these distinct entities would be undesirable.

[00764] In some embodiments, the usage of discriminates described above may be extended to other entities. For example, if a first user in a first customer's environment is a database administrator and a second user in a second customer's environment is also a database administrator, these users may initially be determined to be similar enough such that cross customer learning can take place. In such an example, however, if an examination of their activities, privileges, or something else reveals that they are not actually occupying the same roles, cross customer learning with respect to these two users may be disabled. In such embodiments, the data platform described above, and models leveraged by such a data platform may be used to identify discriminates that may be used to decouple multiple users, applications, microservices, devices, or other entities that would otherwise be candidates for cross customer learning.

[00765] In some embodiments, the data platforms may be configured to analyze, monitor, or otherwise observe environments other than cloud deployments. In fact, the data platforms described here could apply the principles and techniques described herein to any environment such as, for example, an on-premises environment, a hybrid cloud environment, or some special purpose environment. As one example of a special purpose environment, consider an example in which the data platforms described herein are used to analyze, monitor, or otherwise observe a container orchestration environment such as a Kubernetes cluster (which may be deployed on-premises, in a public cloud, or in some other way). In such an example, the data platform may be configured to ingest Kubernetes audit logs via one or more agents or in some other way. Through the ingestion and subsequent analysis of such audit logs, the data platform may model normal behaviors of a Kubernetes cluster, normal behavior of a cluster administrator, and so on. As described above, any deviations from such normal behaviors may result in an alert being generated or some other remediation workflow being initiated. For example, if the ingestion and subsequent analysis of audit logs revealed that workloads are deployed and deleted according to some pattern, a customer deploying or deleting a workload in a manner that is inconsistent with

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00708

the identified pattern may result in an alert being generated or some other remediation workflow being initiated. In fact, by evaluating audit logs from multiple customer's Kubernetes clusters, cross customer learning can be carried out to help define normal behavior for a Kubernetes cluster, a Kubernetes administrator, or some other entity associated with a Kubernetes cluster. For example, evaluating audit logs from the Kubernetes deployment of multiple customers may reveal that one entity (presumed to be a Kubernetes administrator, or related group of such administrators) is generally responsible for creating and deleting nodes from the cluster whereas another entity (perhaps a developer) is responsible for deploying new versions of the code that is executing in a container that is supported by the cluster. In such an example, if a single entity (or related group of entities) is observed creating nodes and modifying the code that is executing on a node by deploying a new container, an alert may be generated or some other remediation workflow may be initiated as a consequence of observing this atypical pattern.

[00766] One or more embodiments may be described herein with the aid of method steps illustrating the performance of specified functions and relationships thereof. The boundaries and sequence of these functional building blocks and method steps have been arbitrarily defined herein for convenience of description. Alternate boundaries and sequences can be defined so long as the specified functions and relationships are appropriately performed. Any such alternate boundaries or sequences are thus within the scope and spirit of the claims. Further, the boundaries of these functional building blocks have been arbitrarily defined for convenience of description. Alternate boundaries could be defined as long as the certain significant functions are appropriately performed. Similarly, flow diagram blocks may also have been arbitrarily defined herein to illustrate certain significant functionality.

[00767] To the extent used, the flow diagram block boundaries and sequence could have been defined otherwise and still perform the certain significant functionality. Such alternate definitions of both functional building blocks and flow diagram blocks and sequences are thus within the scope and spirit of the claims. One of average skill in the art will also recognize that the functional building blocks, and other illustrative blocks, modules and components herein, can be implemented as illustrated or by discrete components, application specific integrated circuits, processors executing appropriate software and the like or any combination thereof.

[00768] While particular combinations of various functions and features of the one or more embodiments are expressly described herein, other combinations of these features and functions are likewise possible. The present disclosure is not limited by the particular examples disclosed herein and expressly incorporates these other combinations.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00709

CLAIMS

What is claimed is:

1. A method of learning from similar cloud deployments, the method comprising:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning;

receiving information describing configurations associated with the additional cloud deployments; and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment.

2. The method of claim 1 further comprising:

receiving information describing one or more actions associated with the additional cloud deployments; and

wherein identifying the one or more configurations to adopt for the first cloud deployment is further based on the one or more actions.

3. The method of claim 2 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00710

4. The method of claim 2 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a detected vulnerability associated with one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to address the vulnerability.

5. The method of claim 1 wherein receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing permissions for one or more users of the additional cloud deployments.

6. The method of claim 4 further comprising:

determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and

wherein identifying one or more configurations to adopt for the first cloud deployment further comprises identifying a reduced privilege level to give to the one or more users.

7. The method of claim 1 further comprising identifying abnormally configured components in the first cloud deployment.

8. The method of claim 1 further comprising receiving information describing one or more deployment processes associated with the additional cloud deployments.

9. A system for learning from similar cloud deployments, the system including computer program instructions that, when executed, cause the system to carry out the steps of:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning;

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00711

receiving information describing configurations associated with the additional cloud deployments; and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment.

10. The system of claim 9 further comprising computer program instructions that, when executed, cause the system to carry out the step of:

receiving information describing one or more actions associated with the additional cloud deployments; and

wherein identifying the one or more configurations to adopt for the first cloud deployment is further based on the one or more actions..

11. The system of claim 10 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat.

12. The system of claim 10 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a detected vulnerability associated with one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to address the vulnerability.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00712

13. The system of claim 9 wherein receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing permissions for one or more users of the additional cloud deployments.

14. The system of claim 9 further comprising computer program instructions that, when executed, cause the system to carry out the step of:

determining, based on the information describing permissions for one or more users of the additional cloud deployments, that one or more users of the first cloud deployment are over-permissioned; and

wherein identifying one or more configurations to adopt for the first cloud deployment further comprises identifying a reduced privilege level to give to the one or more users.

15. The system of claim 9 further comprising computer program instructions that, when executed, cause the system to carry out the step identifying abnormally configured components in the first cloud deployment.

16. The system of claim 9 further comprising computer program instructions that, when executed, cause the system to carry out the step receiving information describing one or more deployment processes associated with the additional cloud deployments.

17. A computer program product for learning from similar cloud deployments, the computer program product disposed on a computer readable medium, the computer program product including computer program instructions that, when executed, carry out the steps of:

identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning;

receiving information describing configurations associated with the additional cloud deployments; and

identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00713

18. The computer program product of claim 17 further comprising computer program instructions that, when executed, carry out the step of:

receiving information describing one or more actions associated with the additional cloud deployments; and

wherein identifying the one or more configurations to adopt for the first cloud deployment is further based on the one or more actions.

19. The computer program product of claim 18 wherein:

receiving information describing one or more actions associated with the additional cloud deployments further comprises receiving information describing a security threat to one or more of the additional cloud deployments; and

receiving information describing configurations associated with the additional cloud deployments further comprises receiving information describing configuration settings used to combat the security threat.

20. The computer program product of claim 18 further comprising computer program instructions that, when executed, carry out the step of creating a polygraph associated with the first cloud deployment.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00714

ABSTRACT

Learning from similar cloud deployments, including: identifying, for at least a portion of a first cloud deployment, one or more additional cloud deployments to utilize for cross-customer learning; receiving information describing configurations associated with the additional cloud deployments; and identifying, based on the configurations, one or more configurations to adopt for the first cloud deployment.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00715

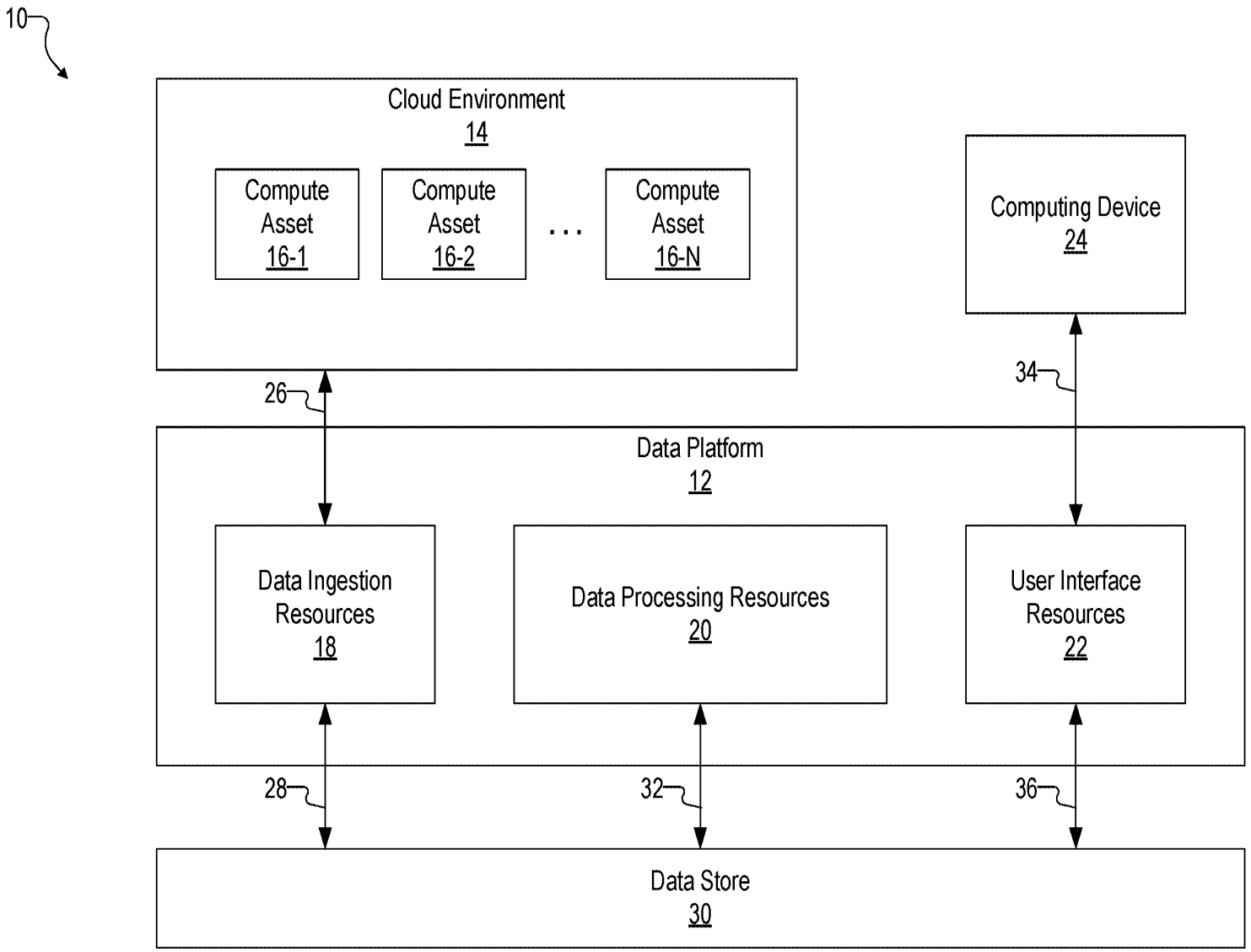


Fig. 1A

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00716

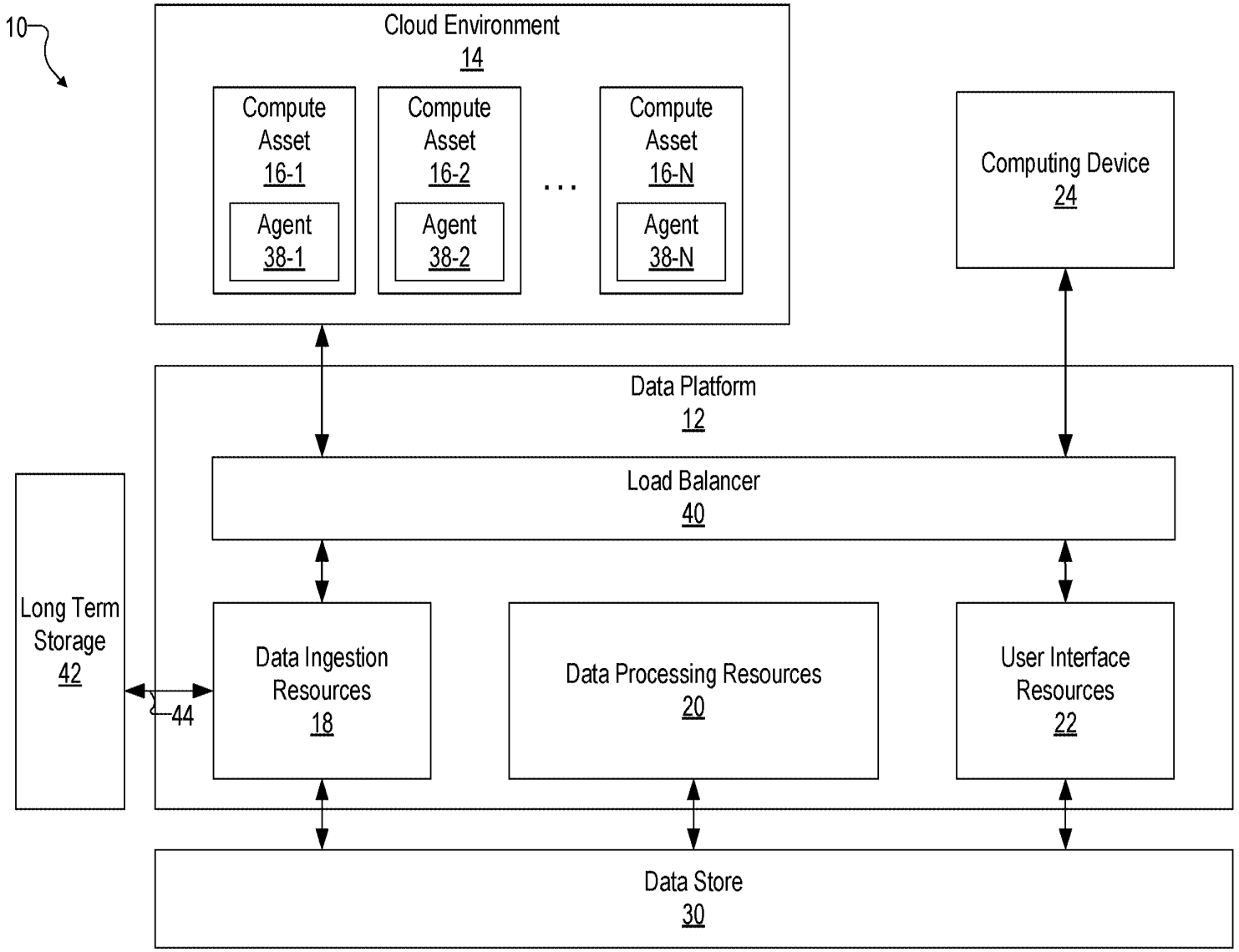

10
Cloud Environment
14
Compute Asset 16-1
Agent 38-1
Compute Asset 16-2
Agent 38-2
Compute Asset 16-N
Agent 38-N
Computing Device
24
Data Platform
12
Load Balancer
40
Long Term Storage
42
44
Data Ingestion Resources
18
Data Processing Resources
20
User Interface Resources
22
Data Store
30

Fig. 1B

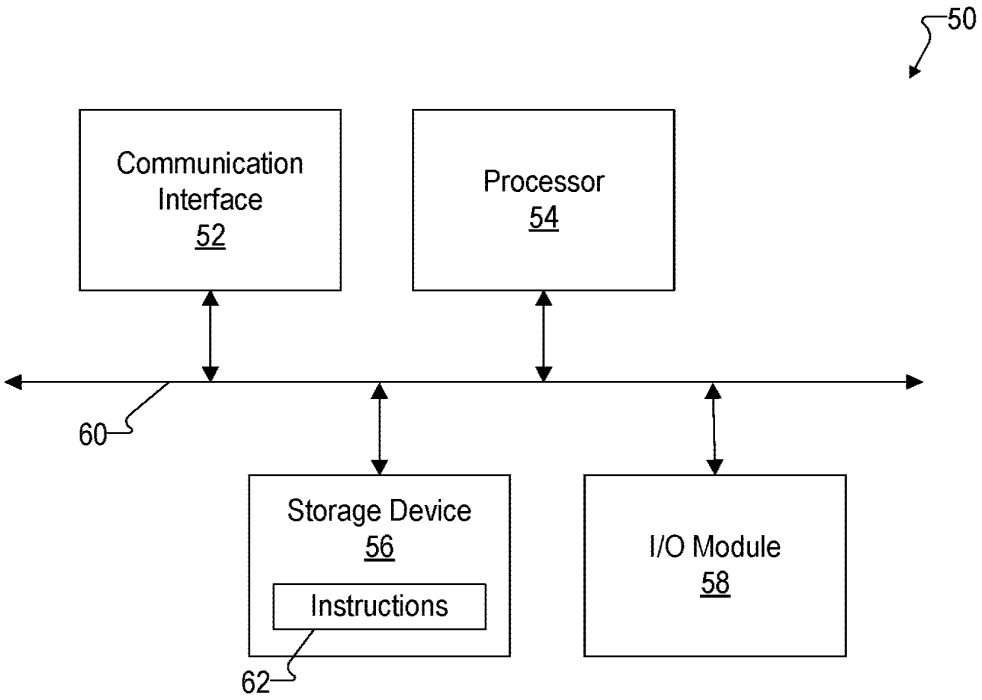

50
Communication Interface 52
Processor 54
60
Storage Device 56
Instructions
62
I/O Module 58

Fig. 1C

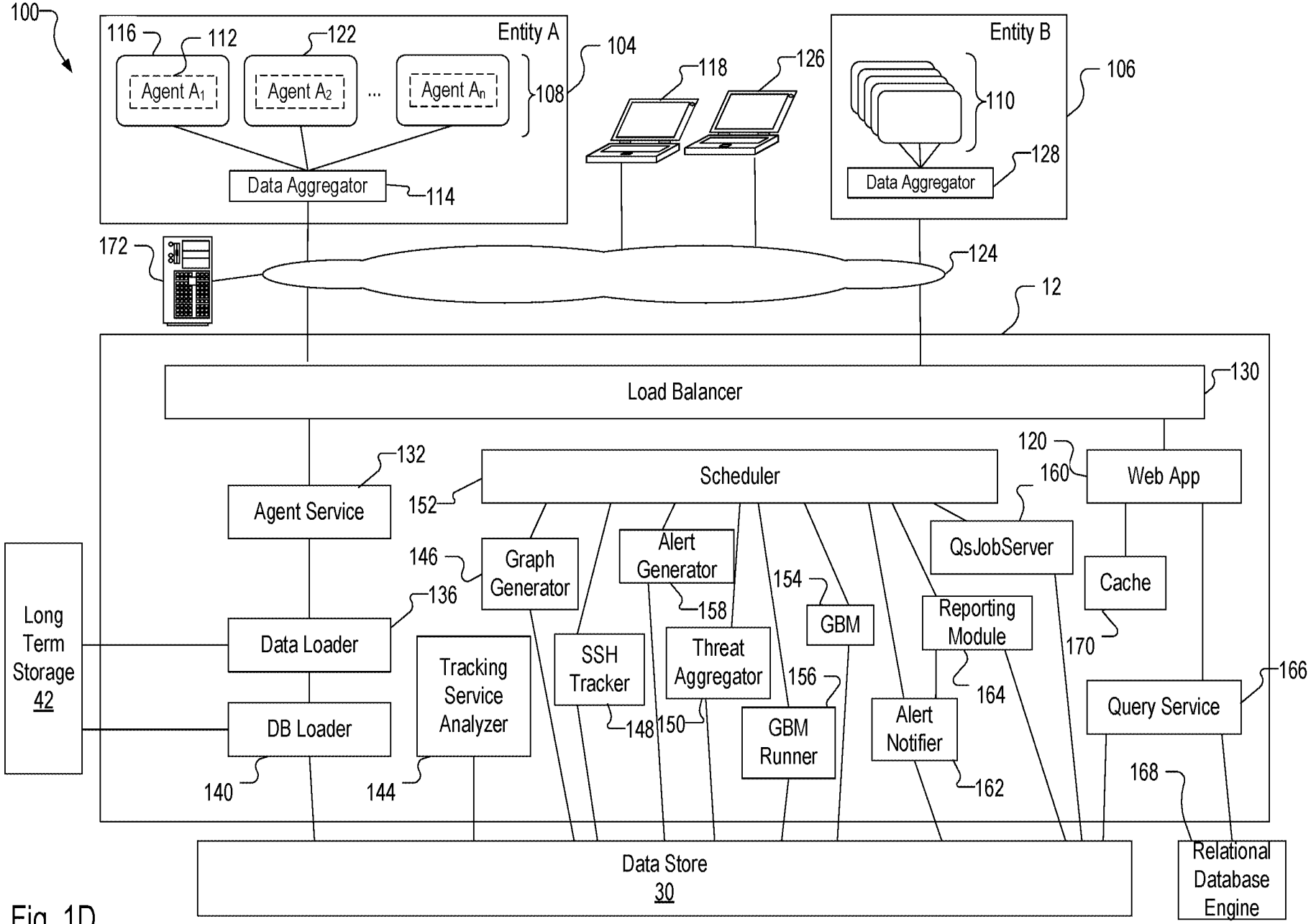

100
116
112
122
Entity A
Agent A1
Agent A2
...
Agent An
108
104
Data Aggregator
114
118
126
Entity B
110
106
128
Data Aggregator
172
124
12
Load Balancer
130
120
160
Web App
132
Agent Service
152
Scheduler
146
Graph Generator
Alert Generator
158
154
GBM
QsJobServer
Cache
170
136
Data Loader
Tracking Service Analyzer
SSH Tracker
148
Threat Aggregator
150
156
GBM Runner
Reporting Module
164
Alert Notifier
162
166
Query Service
Long Term Storage 42
DB Loader
140
144
168
Data Store 30
Relational Database Engine

Fig. 1D

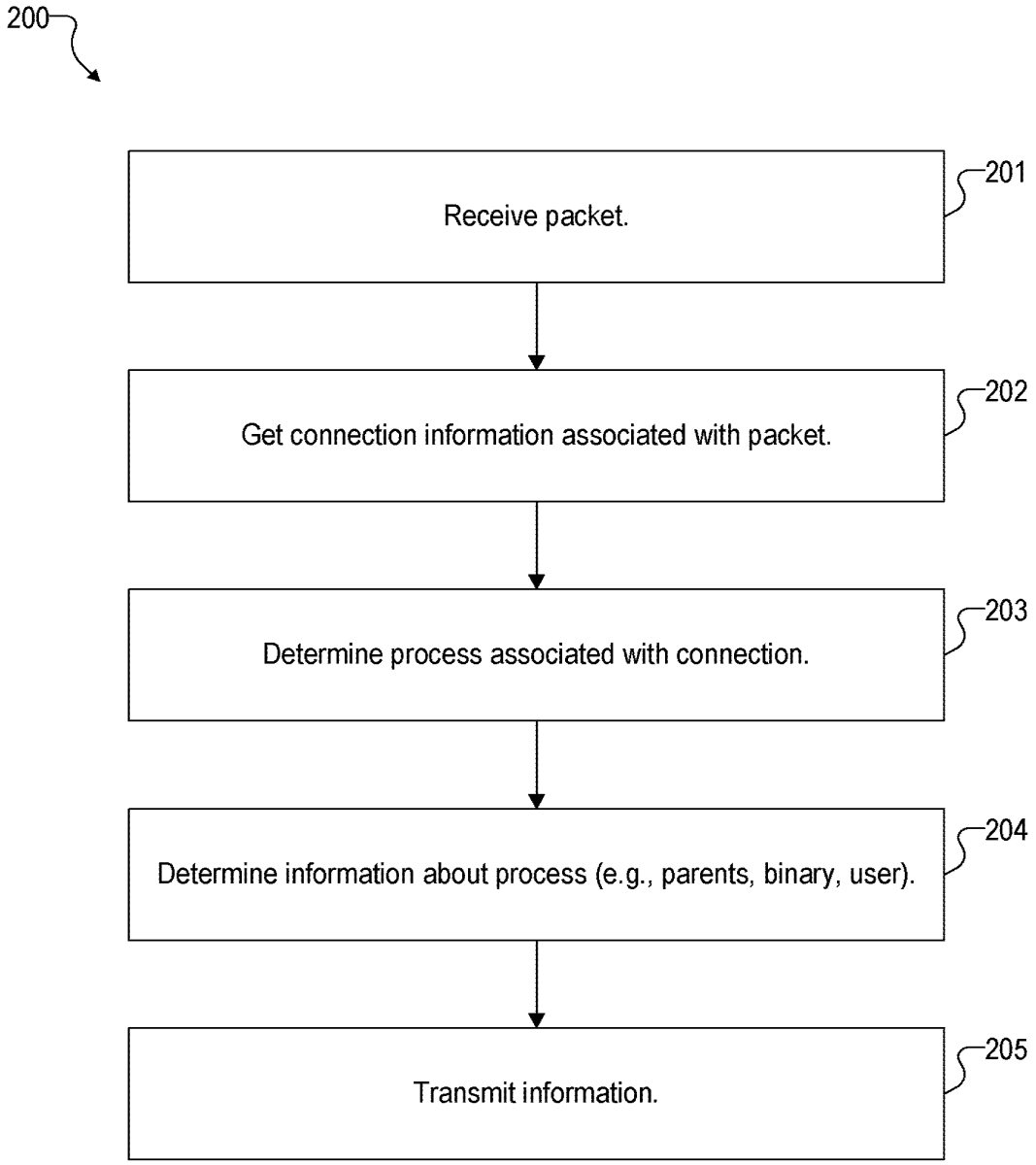

200
Receive packet.
201
Get connection information associated with packet.
202
Determine process associated with connection.
203
Determine information about process (e.g., parents, binary, user).
204
Transmit information.
205

Fig. 2A

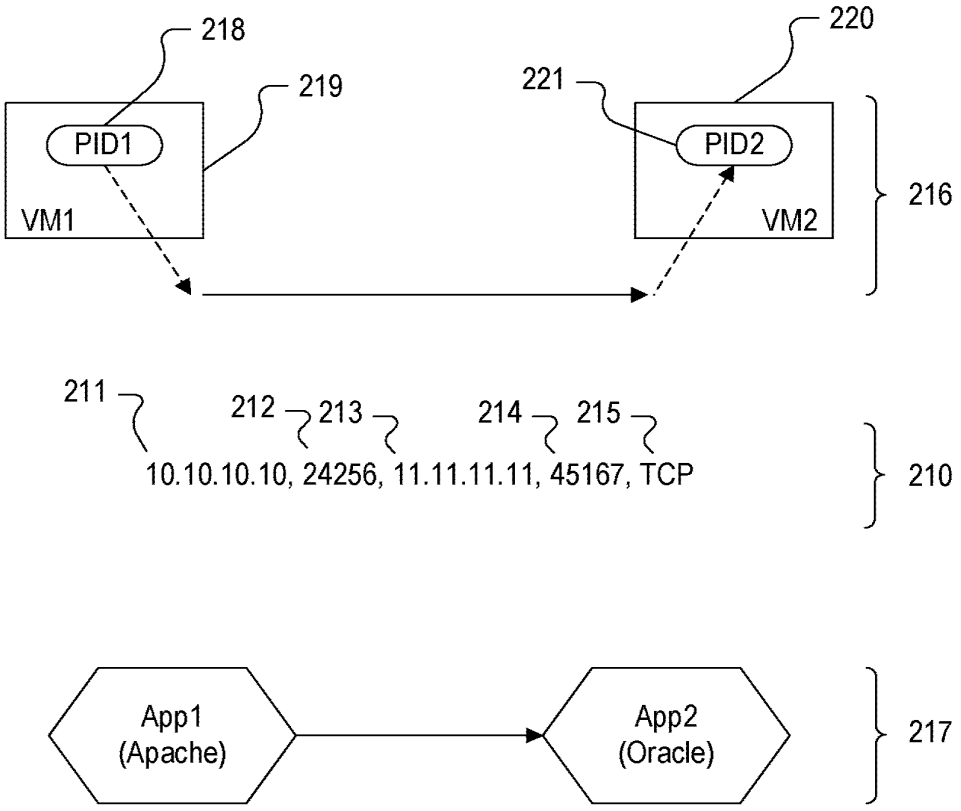

218
219
220
221
PID1
PID2
VM1
VM2
216
211
212
213
214
215
10.10.10.10, 24256, 11.11.11.11, 45167, TCP
210
App1
(Apache)
App2
(Oracle)
217

Fig. 2B

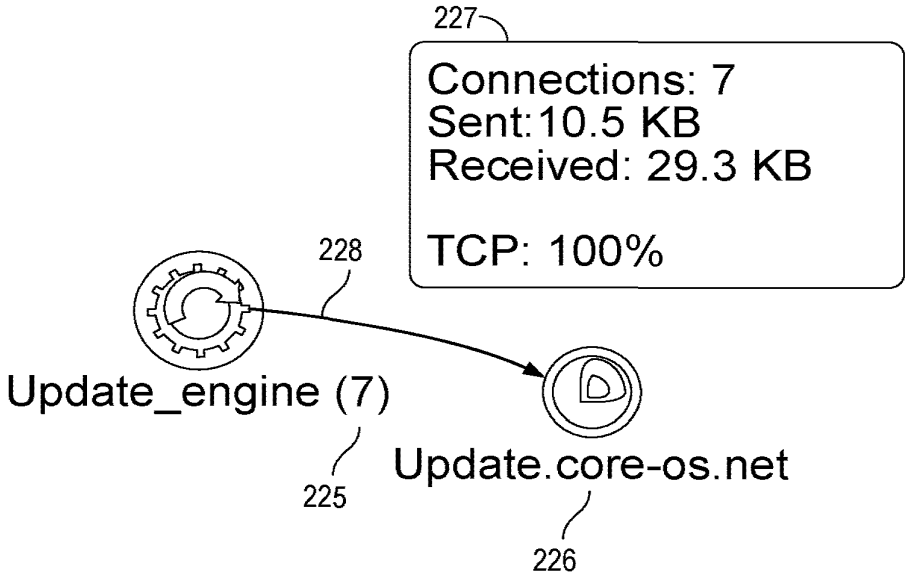

227
Connections: 7
Sent:10.5 KB
Received: 29.3 KB
TCP: 100%
228
Update_engine (7)
225
Update.core-os.net
226

Fig. 2C

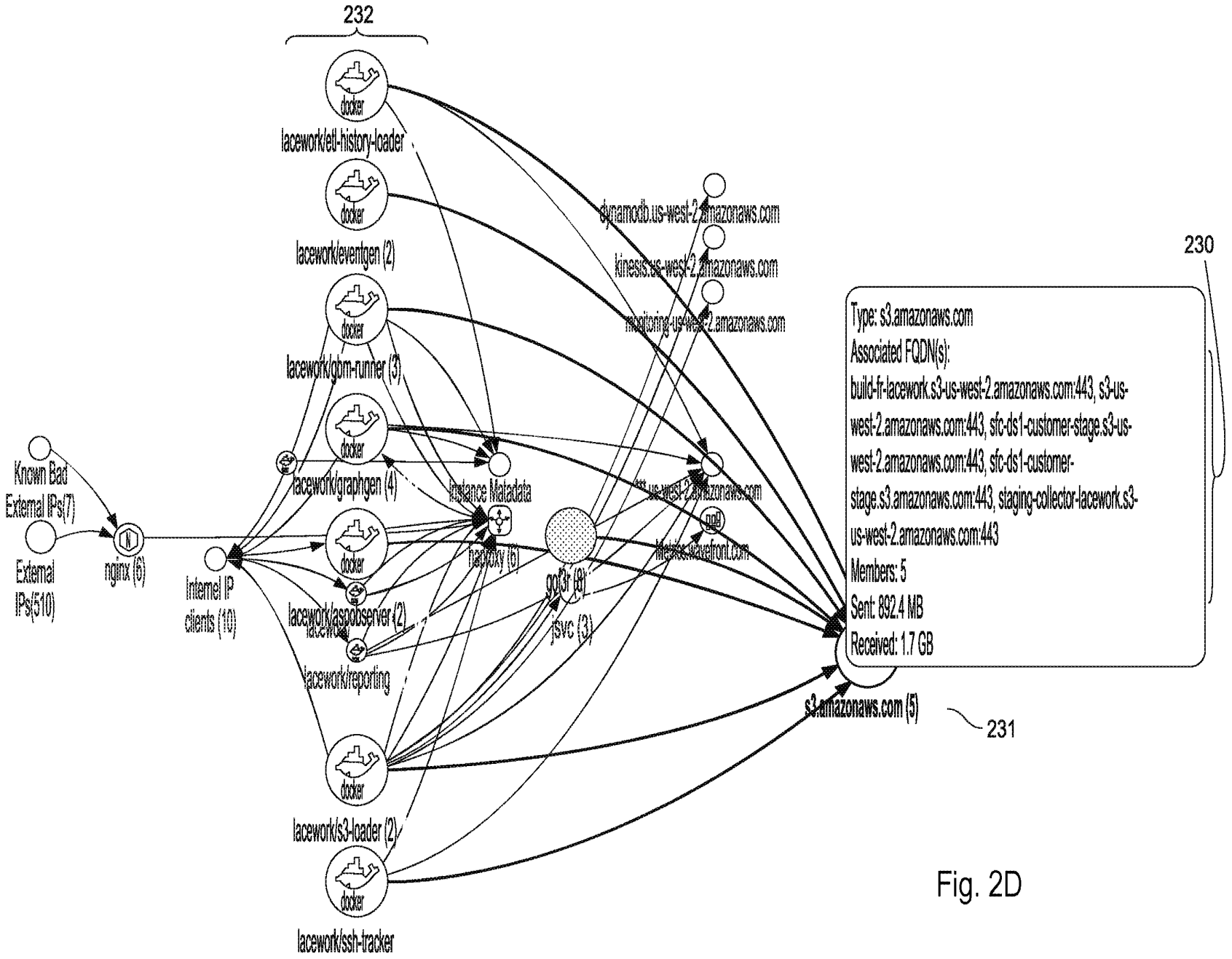


Fig. 2D

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00723

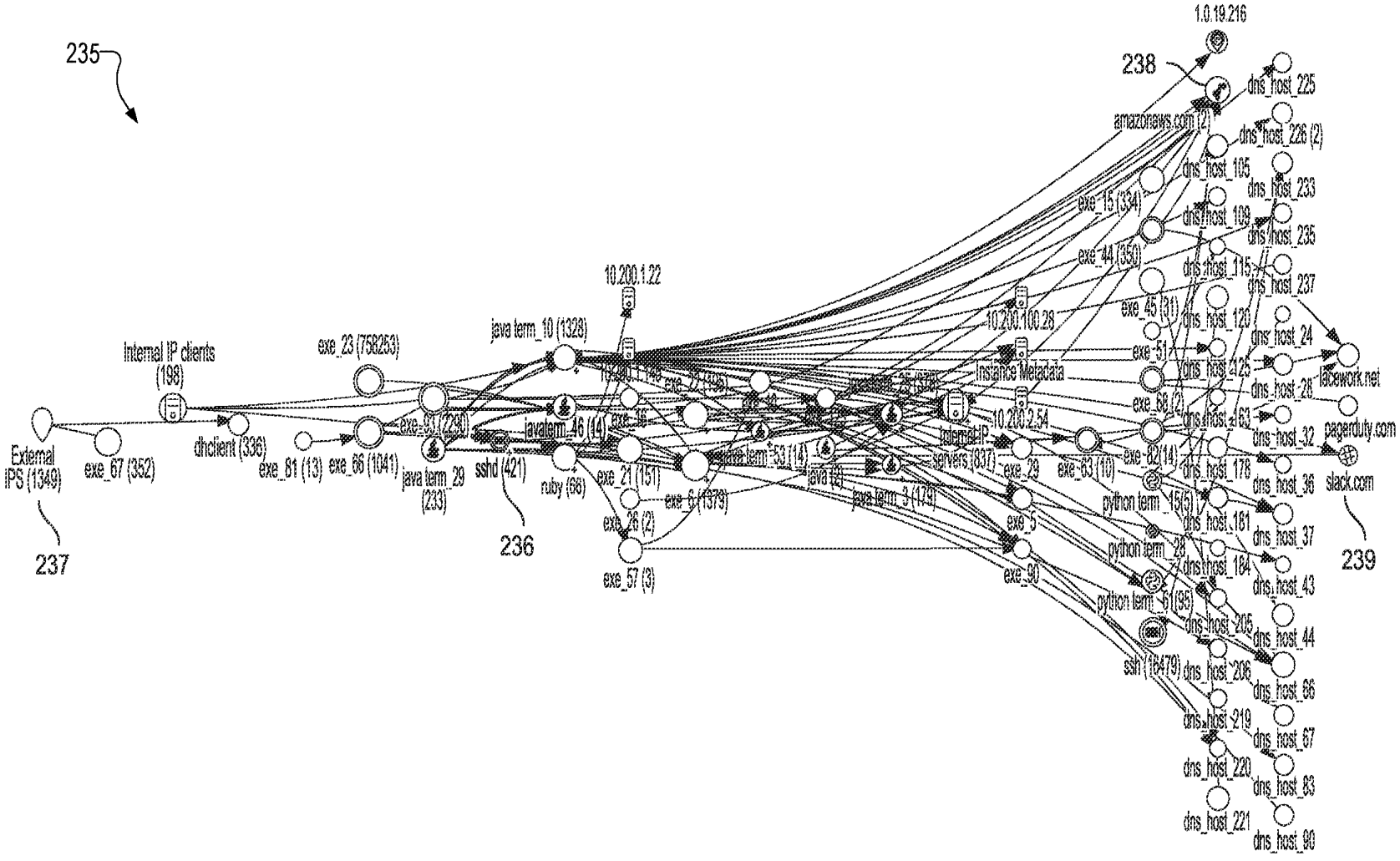


Fig. 2E

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00724

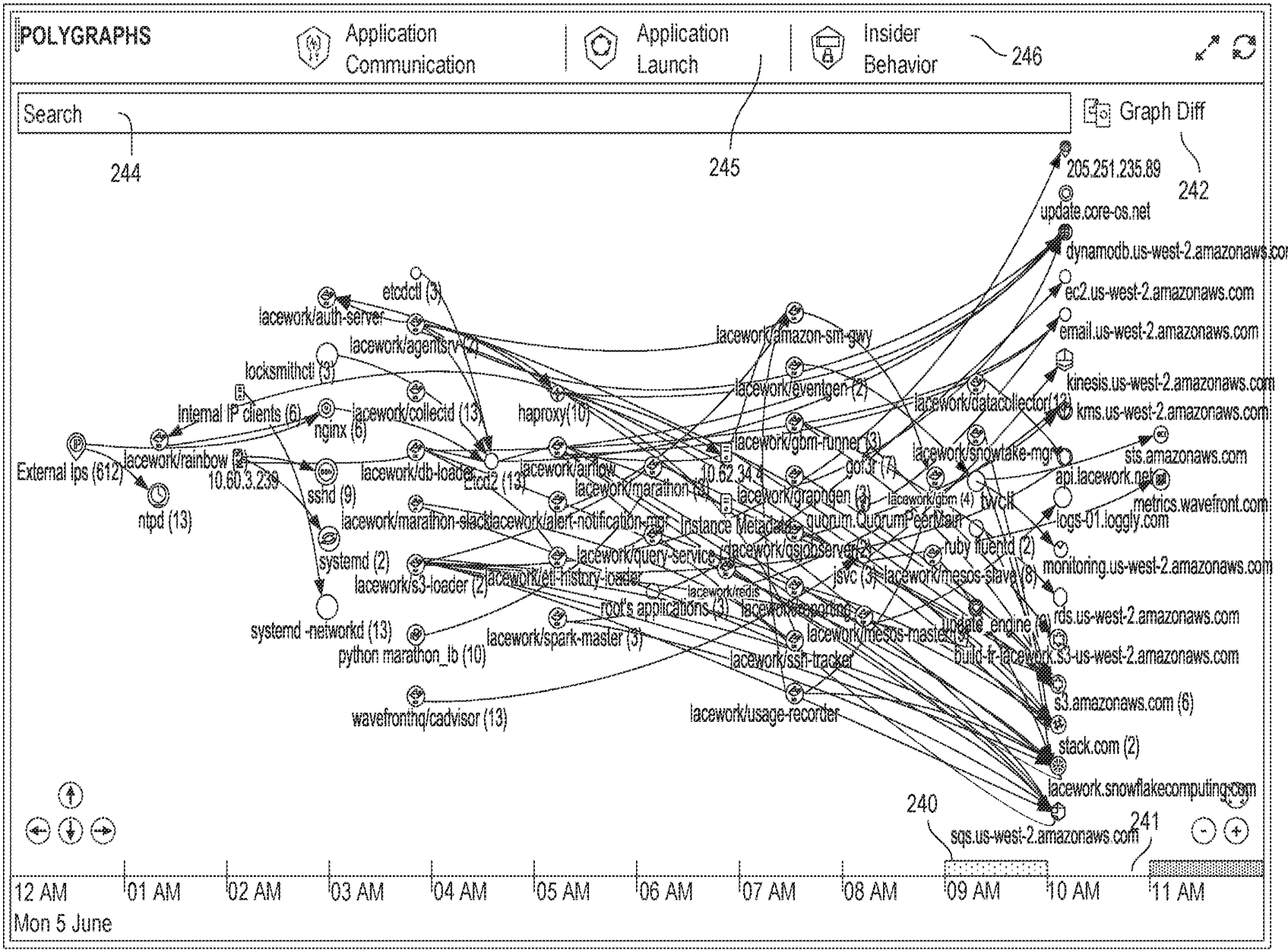


Fig. 2F

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00725

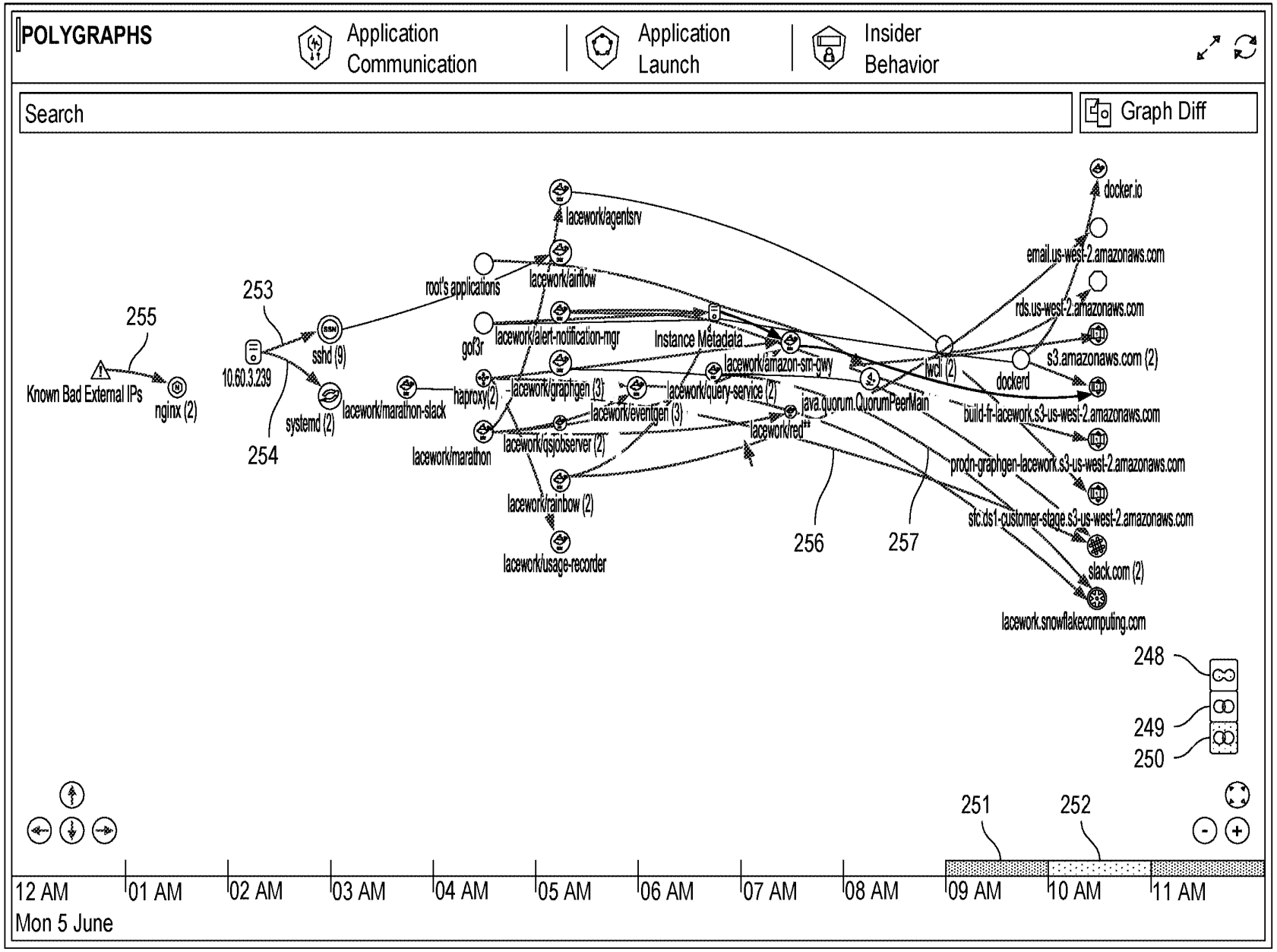


Fig. 2G

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00726

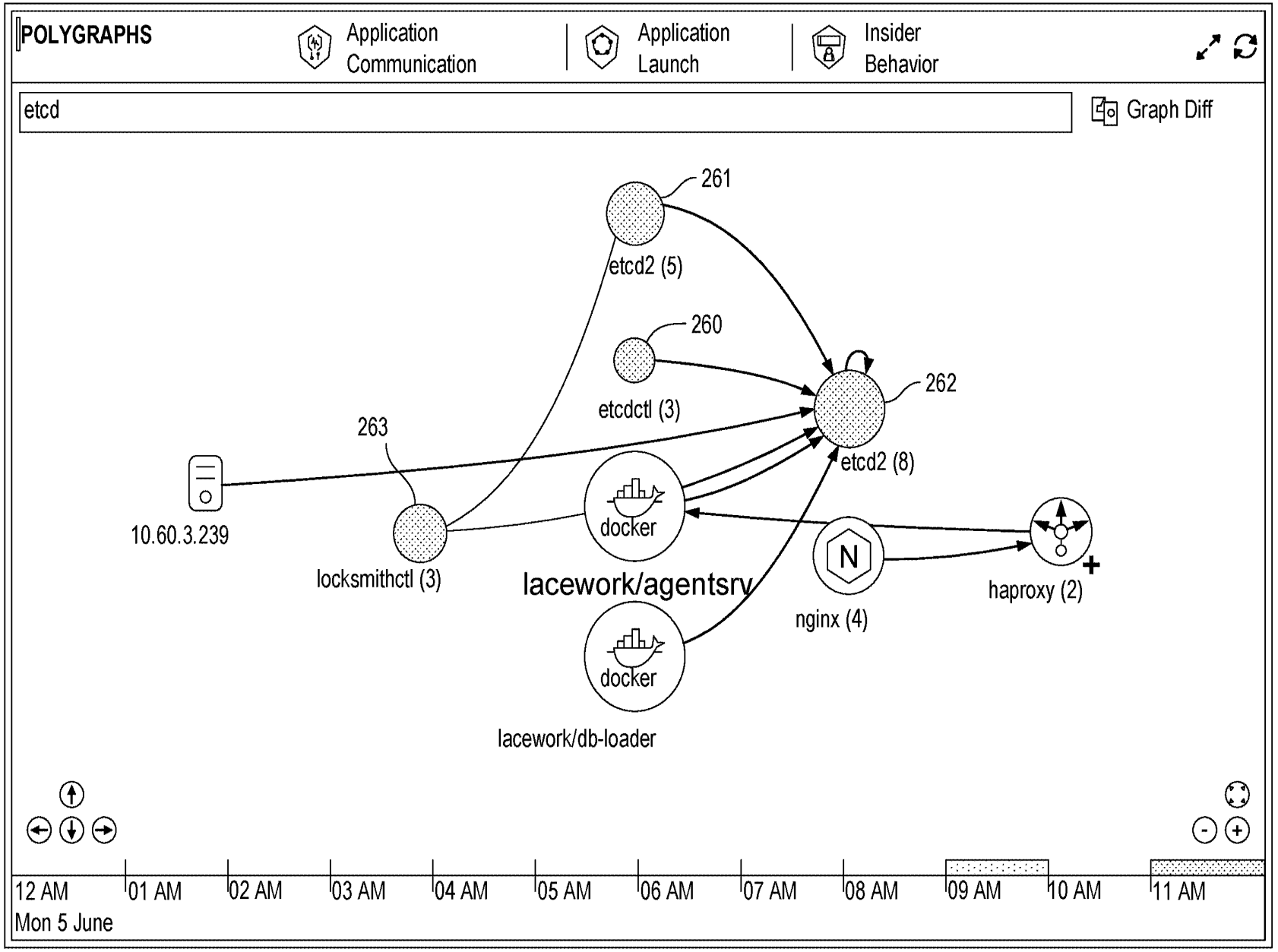

POLYGRAPHS
Application Communication
Application Launch
Insider Behavior
etcd
Graph Diff
261
etcd2 (5)
260
etcdctl (3)
262
etcd2 (8)
263
10.60.3.239
locksmithctl (3)
docker
lacework/agentsrv
docker
lacework/db-loader
N
nginx (4)
haproxy (2)
12 AM
01 AM
02 AM
03 AM
04 AM
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM
Mon 5 June

Fig. 2H

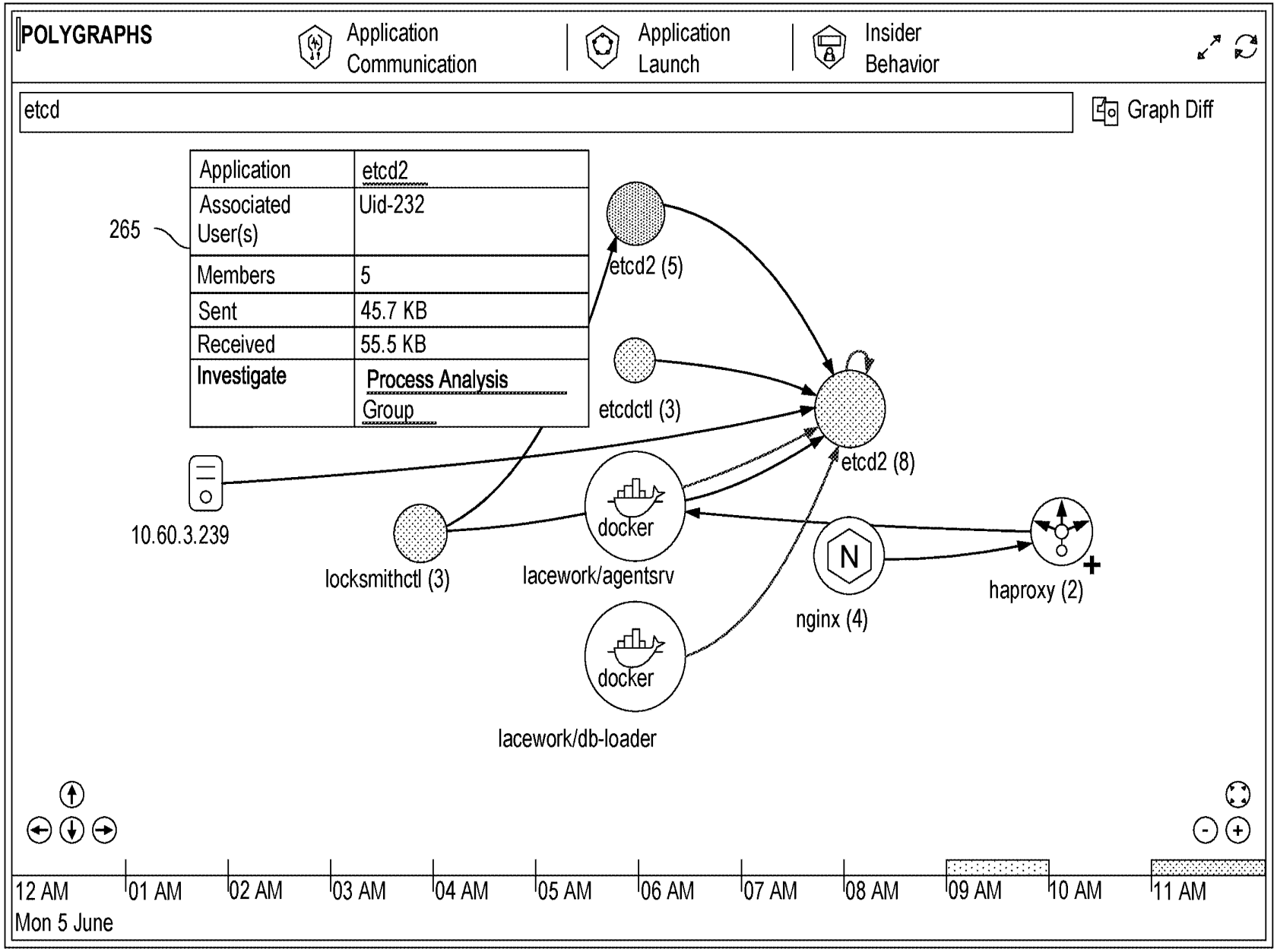

POLYGRAPHS
Application Communication
Application Launch
Insider Behavior
etcd
Graph Diff
Application
etcd2
Associated User(s)
Uid-232
265
Members
5
Sent
45.7 KB
Received
55.5 KB
Investigate
Process Analysis Group
etcd2 (5)
etcdctl (3)
etcd2 (8)
10.60.3.239
locksmithctl (3)
docker
lacework/agentsrv
N
nginx (4)
haproxy (2)
docker
lacework/db-loader
12 AM
01 AM
02 AM
03 AM
04 AM
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM
Mon 5 June

Fig. 2I

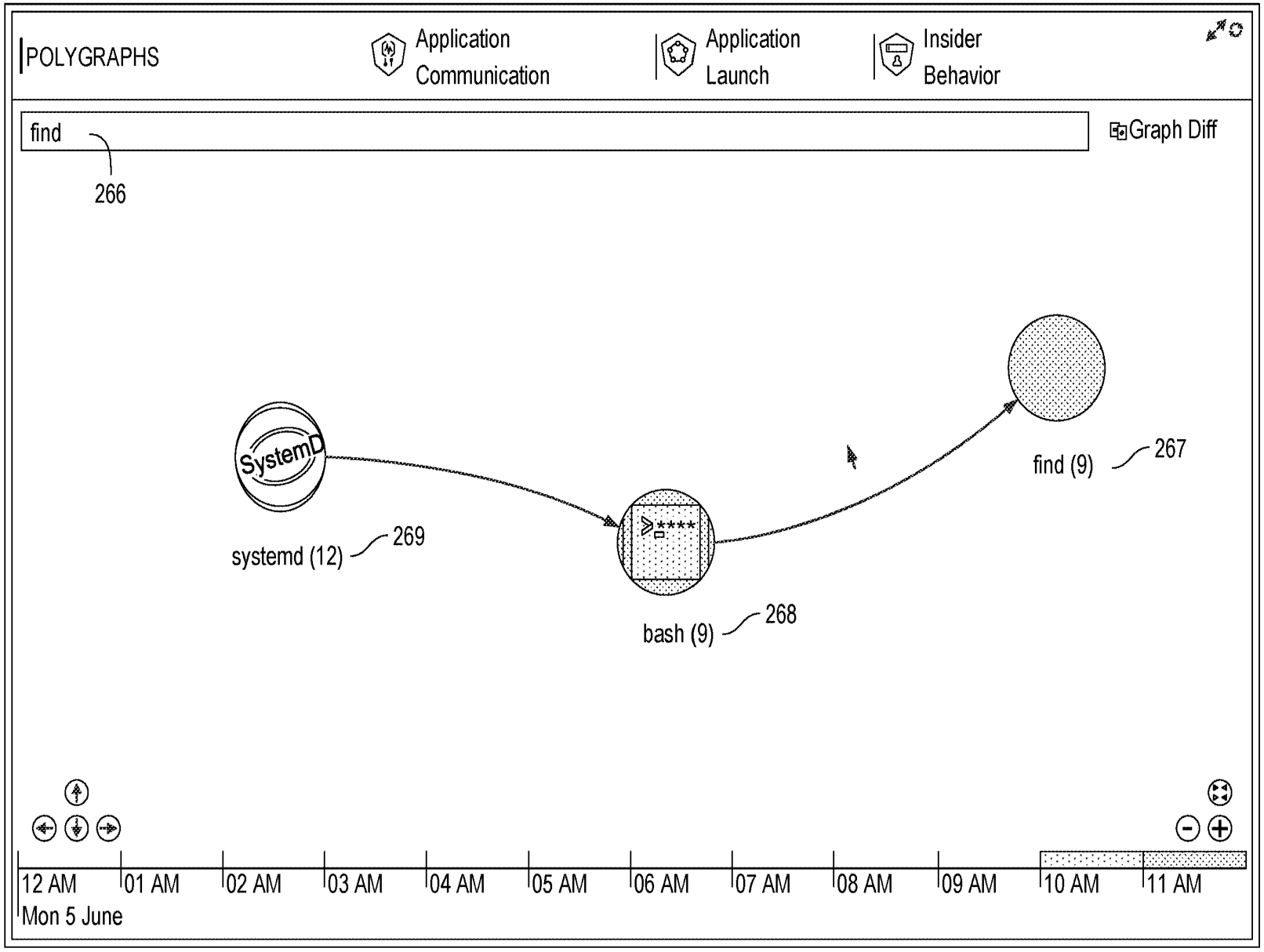

POLYGRAPHS
Application Communication
Application Launch
Insider Behavior
find
Graph Diff
266
SystemD
systemd (12)
269
bash (9)
268
find (9)
267
12 AM
Mon 5 June
01 AM
02 AM
03 AM
04 AM
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM

Fig. 2J

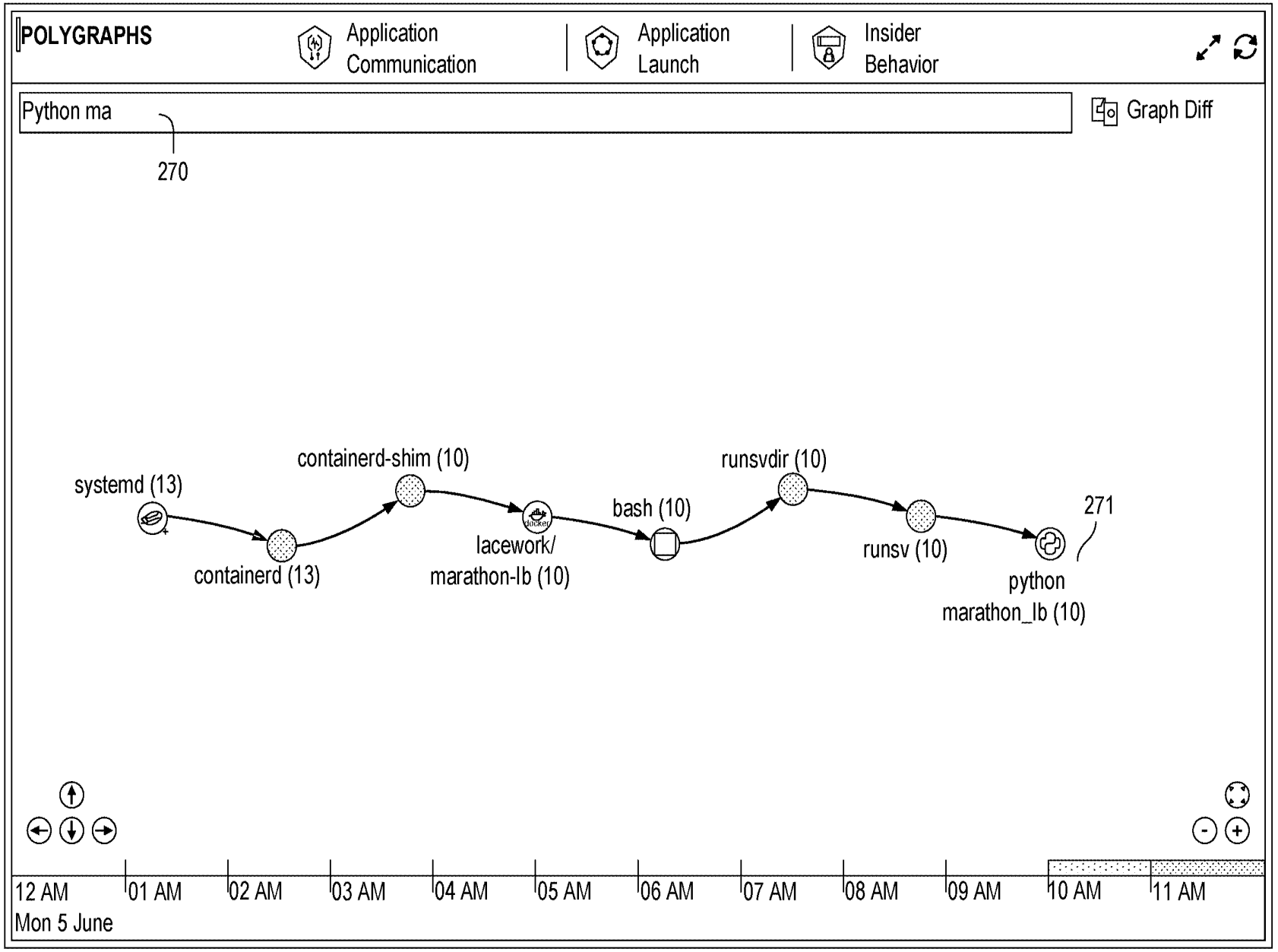

POLYGRAPHS
Application Communication
Application Launch
Insider Behavior
Python ma
270
Graph Diff
systemd (13)
containerd (13)
containerd-shim (10)
lacework/ marathon-lb (10)
bash (10)
runsvdir (10)
runsv (10)
271
python marathon_lb (10)
12 AM
Mon 5 June
01 AM
02 AM
03 AM
04 AM
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM

Fig. 2K

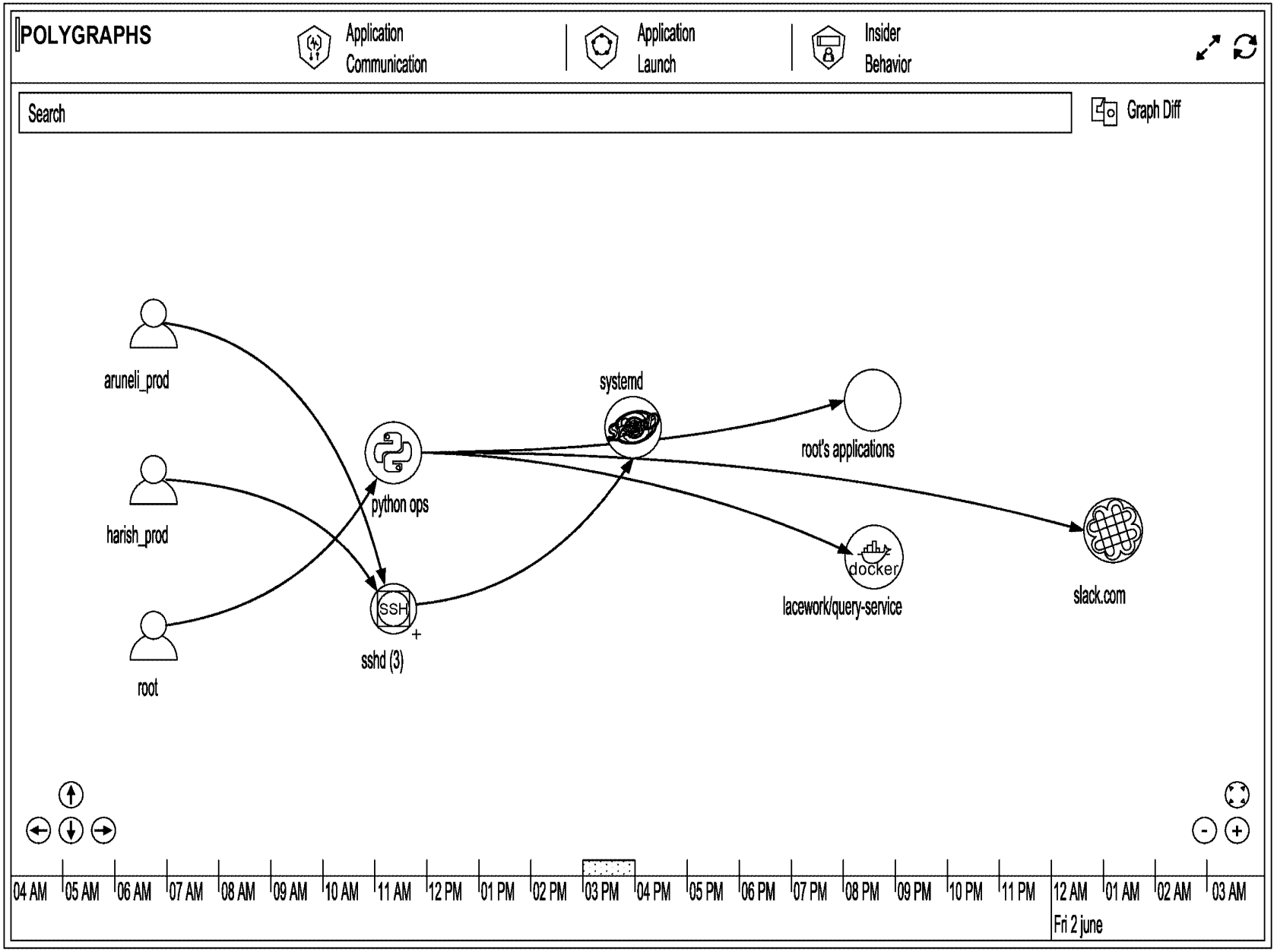


Fig. 2L

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00731

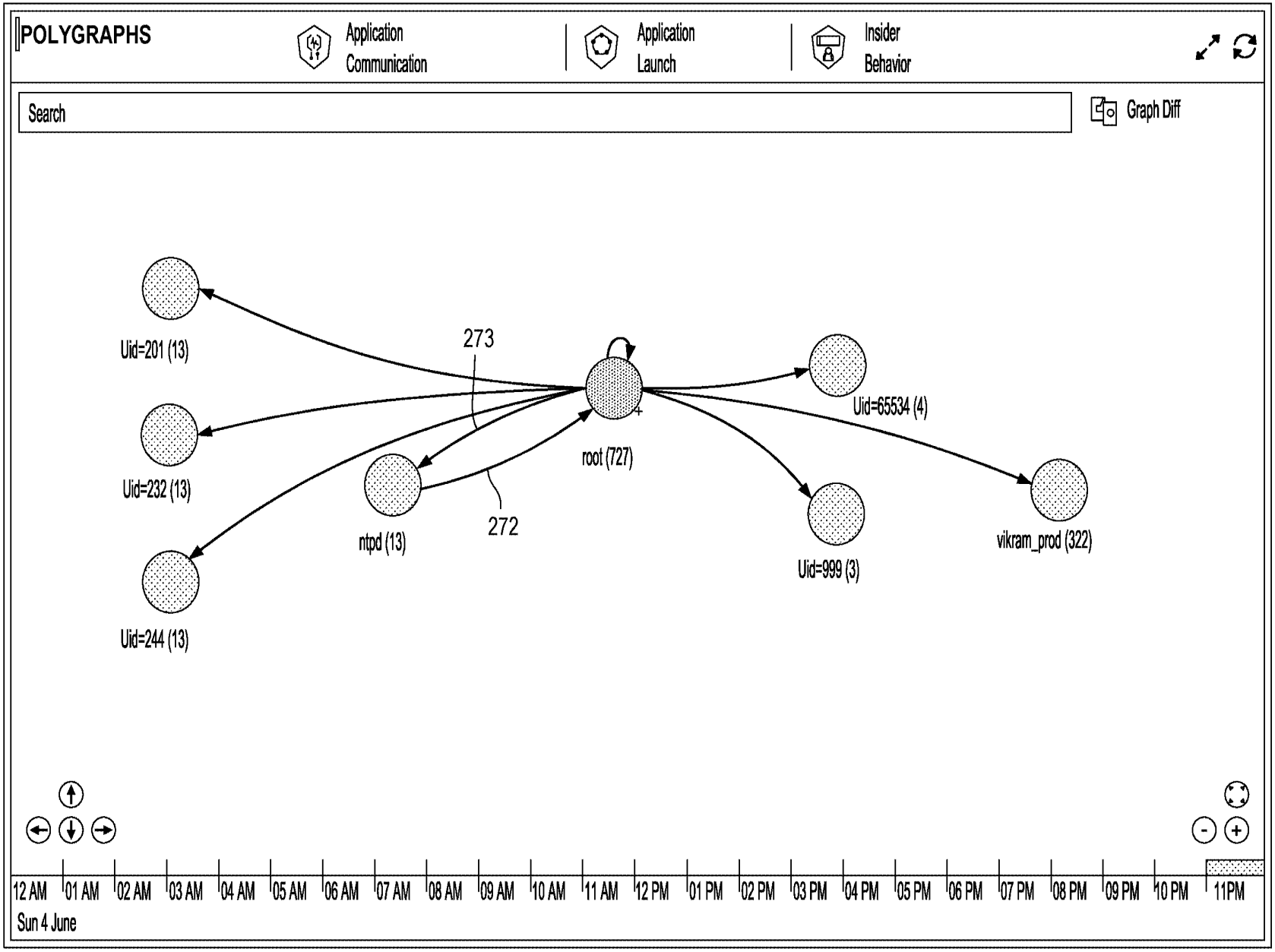

POLYGRAPHS
Application Communication
Application Launch
Insider Behavior
Search
Graph Diff
Uid=201 (13)
Uid=232 (13)
Uid=244 (13)
273
272
ntpd (13)
root (727)
Uid=65534 (4)
Uid=999 (3)
vikram_prod (322)
12 AM
01 AM
02 AM
03 AM
04 AM
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM
12 PM
01 PM
02 PM
03 PM
04 PM
05 PM
06 PM
07 PM
08 PM
09 PM
10 PM
11PM
Sun 4 June

Fig. 2M

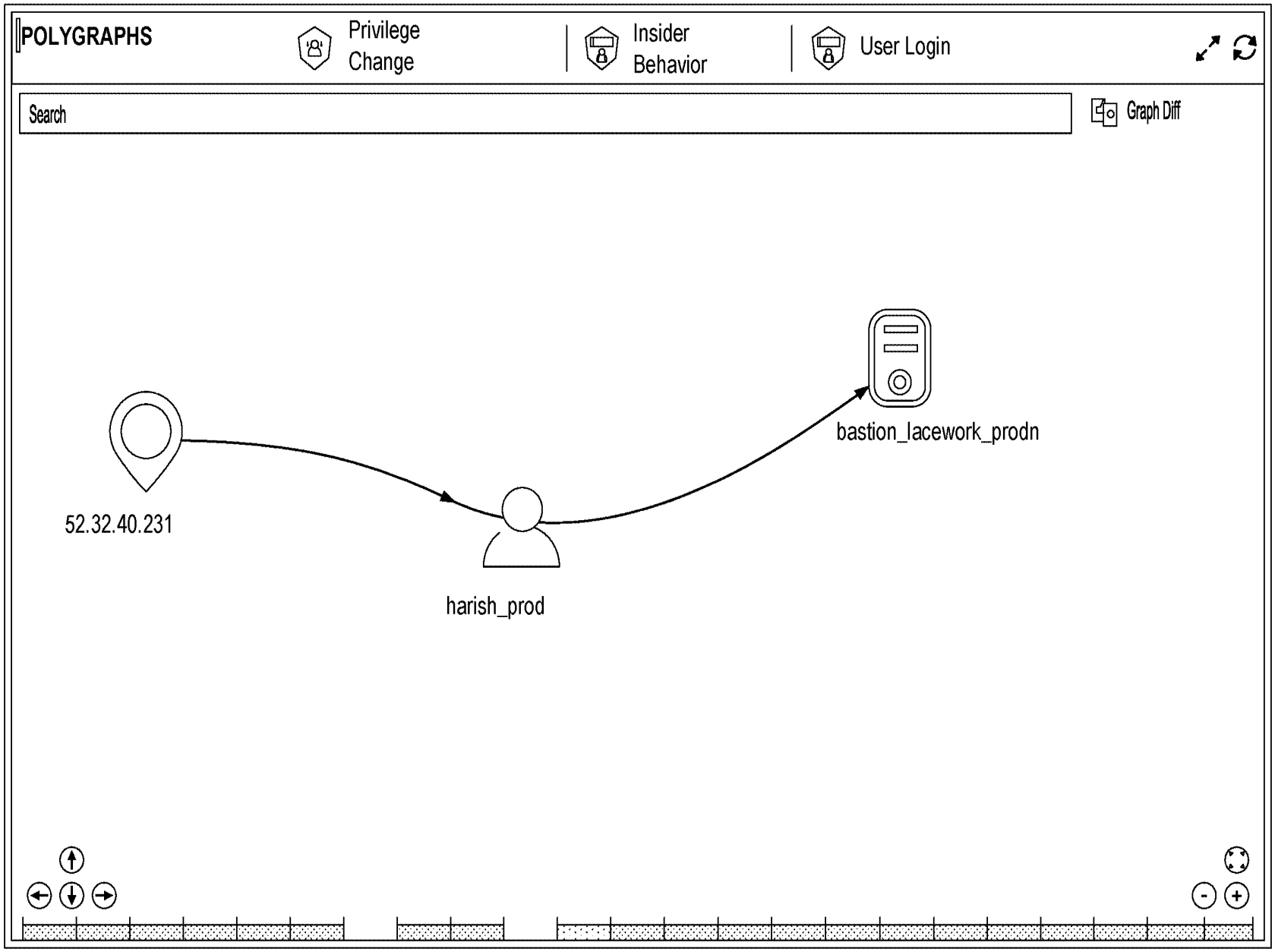

POLYGRAPHS
Privilege Change
Insider Behavior
User Login
Search
Graph Diff
bastion_lacework_prodn
52.32.40.231
harish_prod

Fig. 2N

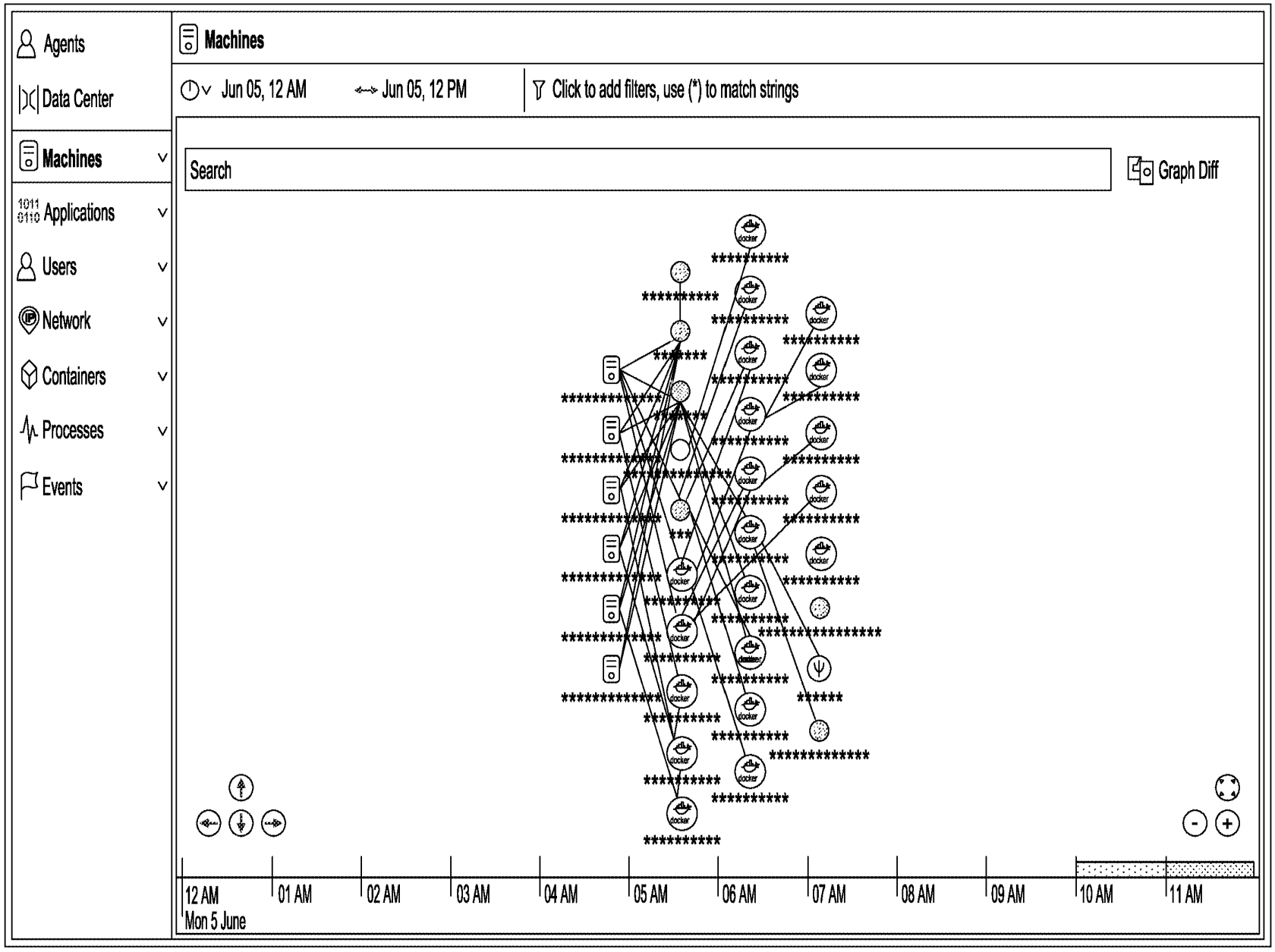

Agents
Data Center
Machines
Applications
Users
Network
Containers
Processes
Events
Machines
Jun 05, 12 AM
Jun 05, 12 PM
Click to add filters, use (*) to match strings
Search
Graph Diff
12 AM
Mon 5 June
01 AM
02 AM
03 AM
04 AM
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM

Fig. 20

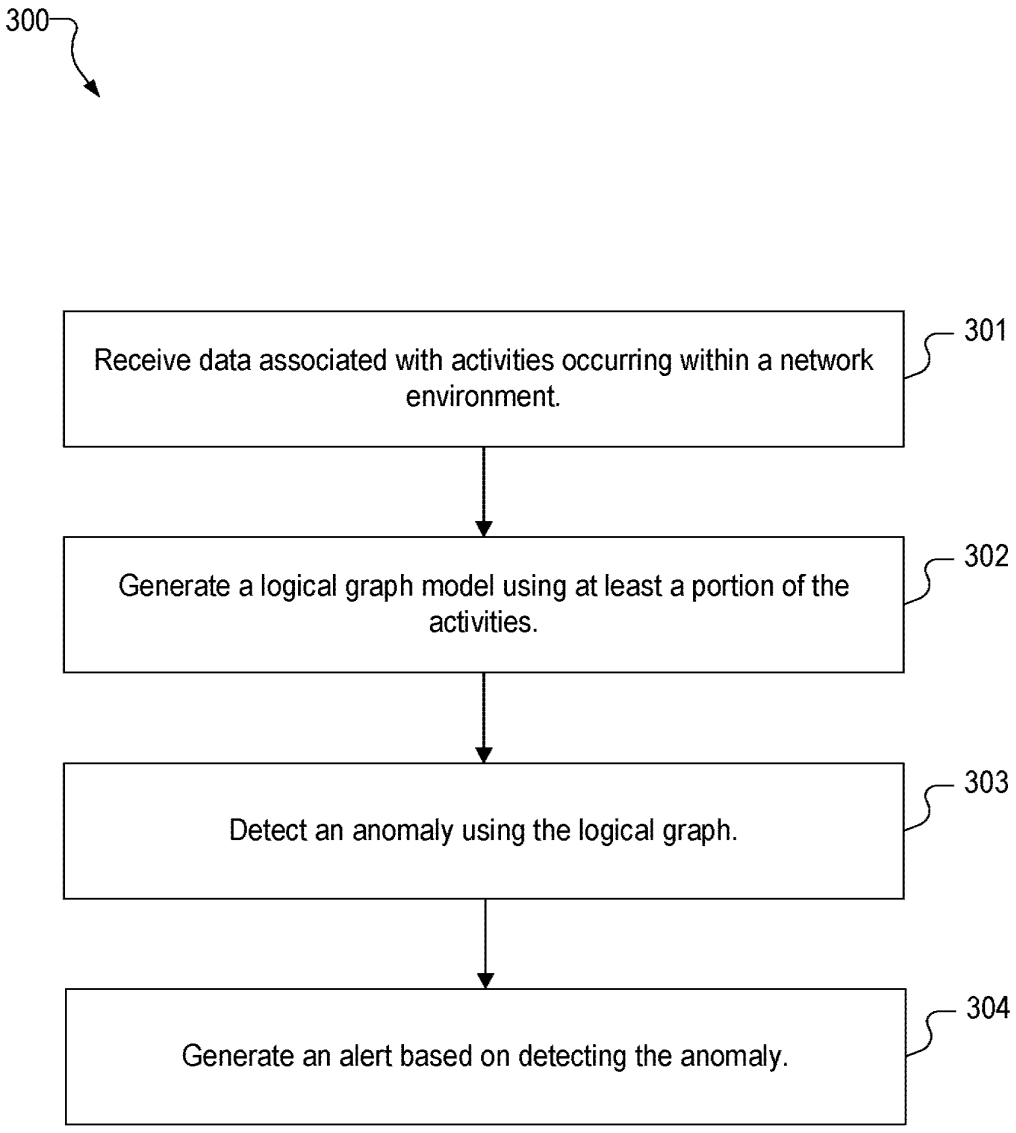

300
Receive data associated with activities occurring within a network environment.
301
Generate a logical graph model using at least a portion of the activities.
302
Detect an anomaly using the logical graph.
303
Generate an alert based on detecting the anomaly.
304

Fig. 3A

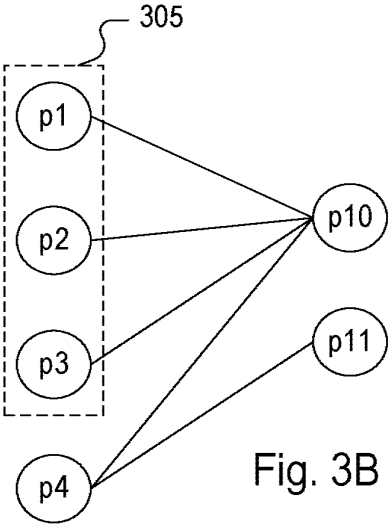


Fig. 3B

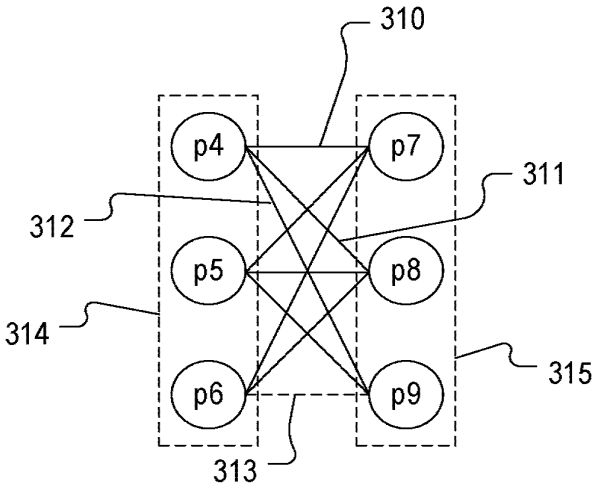


Fig. 3C

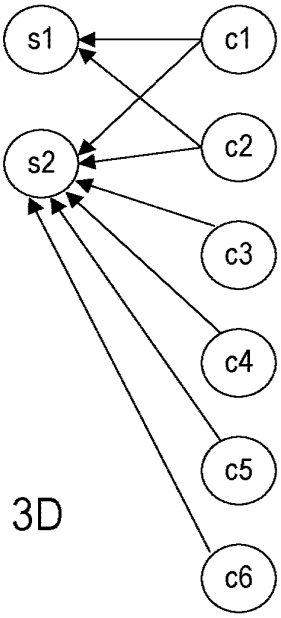


Fig. 3D

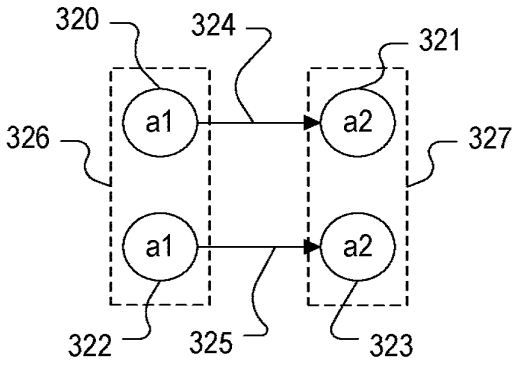


Fig. 3E

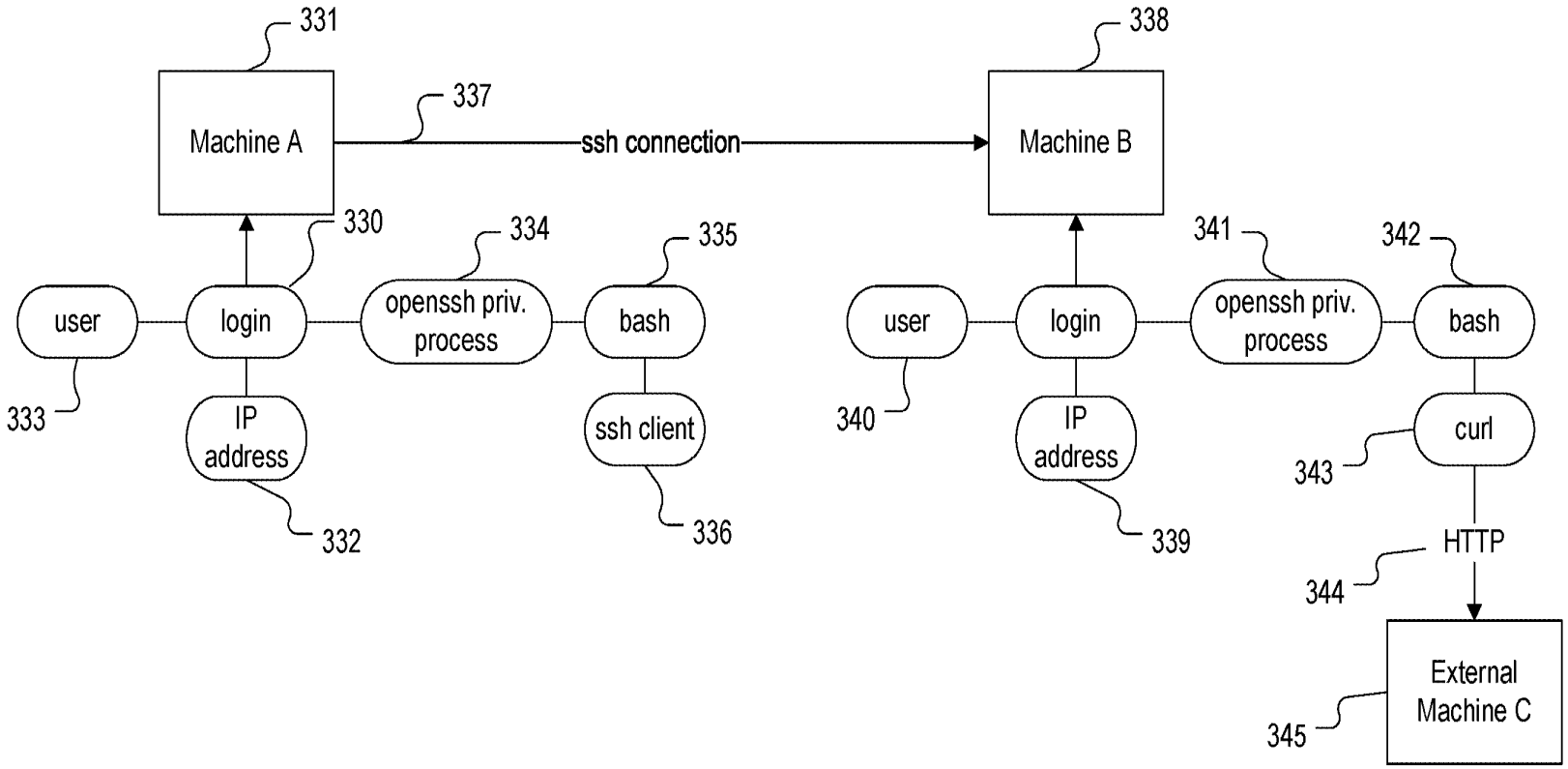


Fig. 3F

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00737

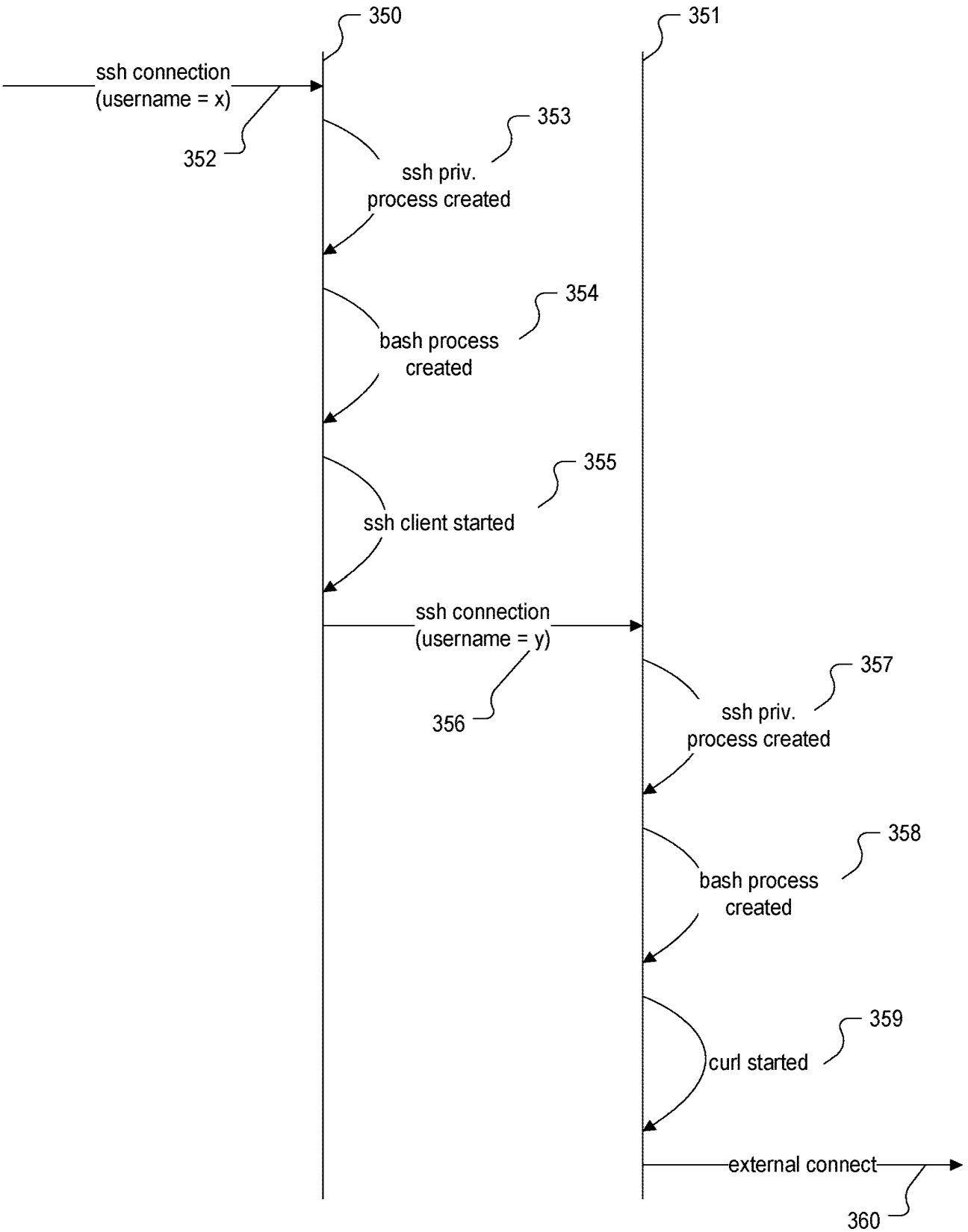

350
351
ssh connection
(username = x)
352
353
ssh priv.
process created
354
bash process
created
355
ssh client started
ssh connection
(username = y)
356
357
ssh priv.
process created
358
bash process
created
359
curl started
external connect
360

Fig. 3G

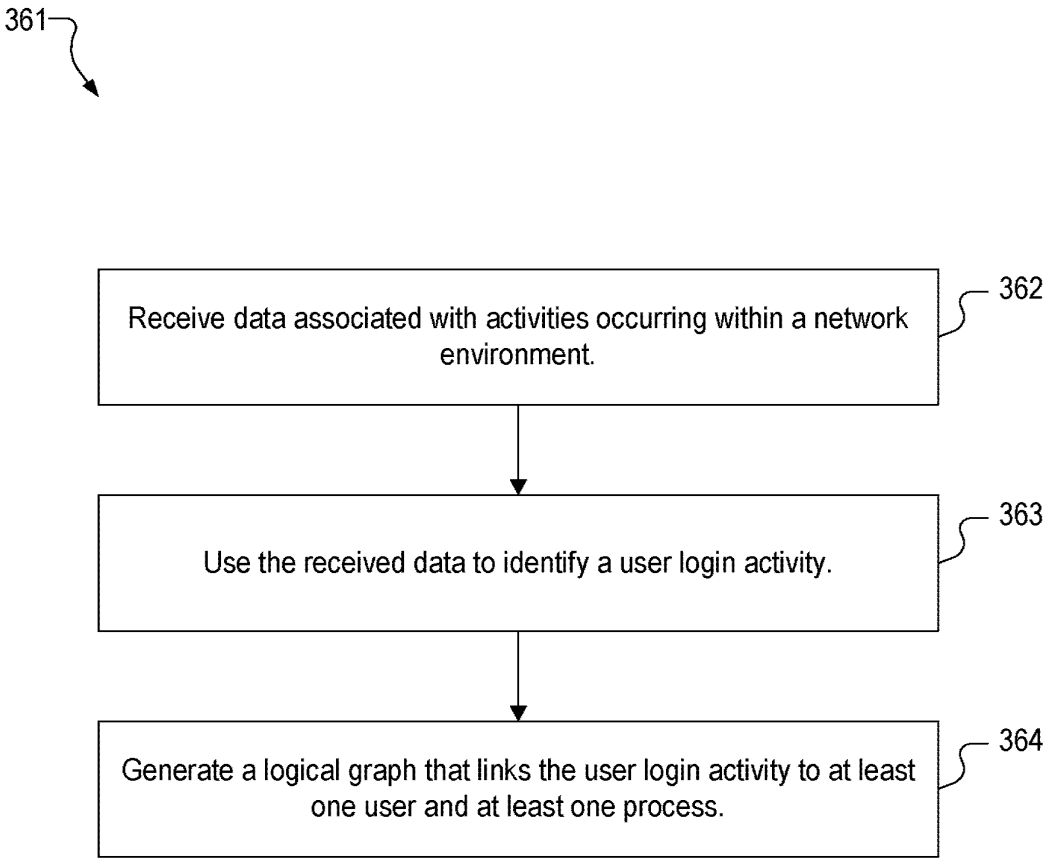

361
Receive data associated with activities occurring within a network environment.
362
Use the received data to identify a user login activity.
363
Generate a logical graph that links the user login activity to at least one user and at least one process.
364

Fig. 3H

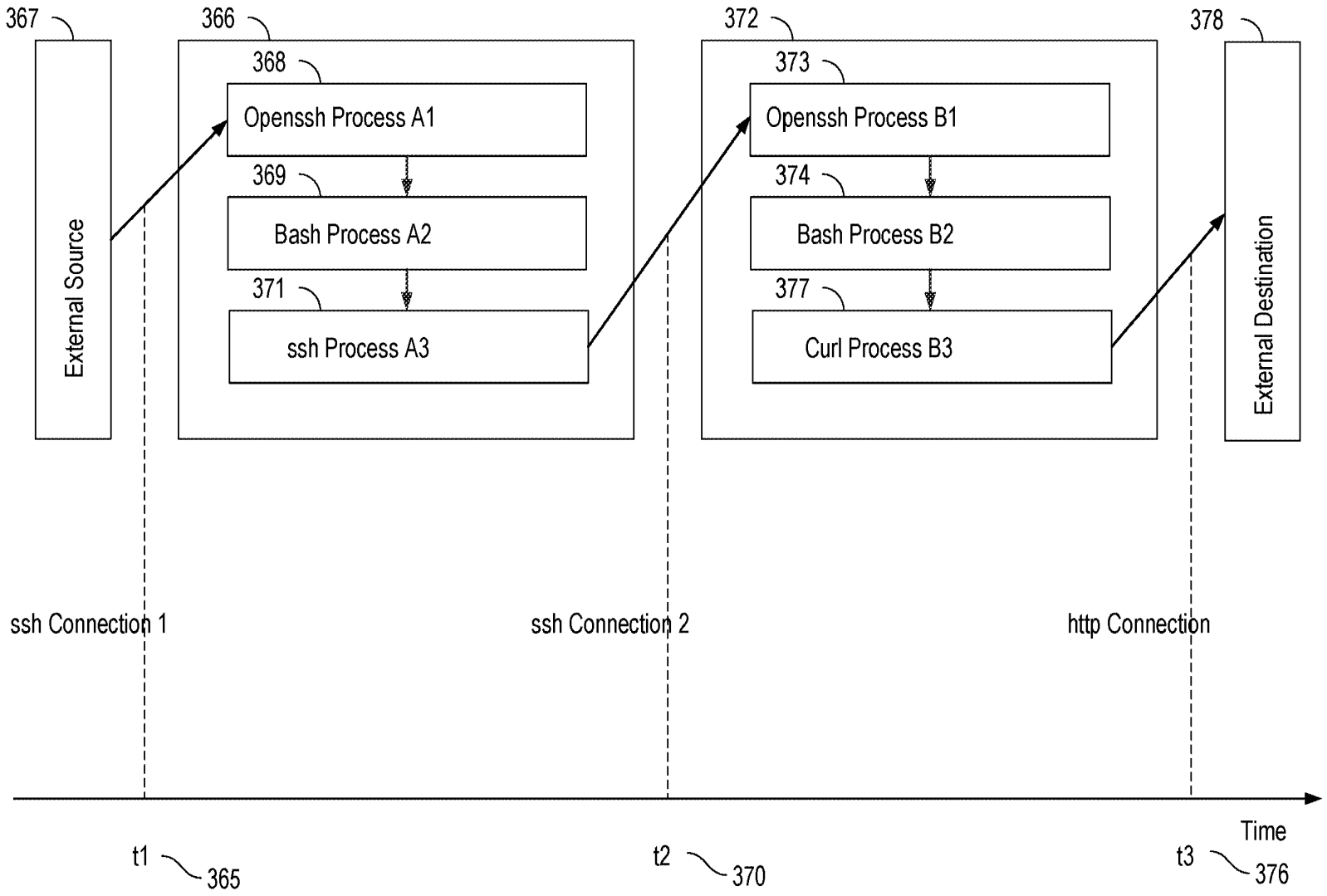


Fig. 3I

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00740

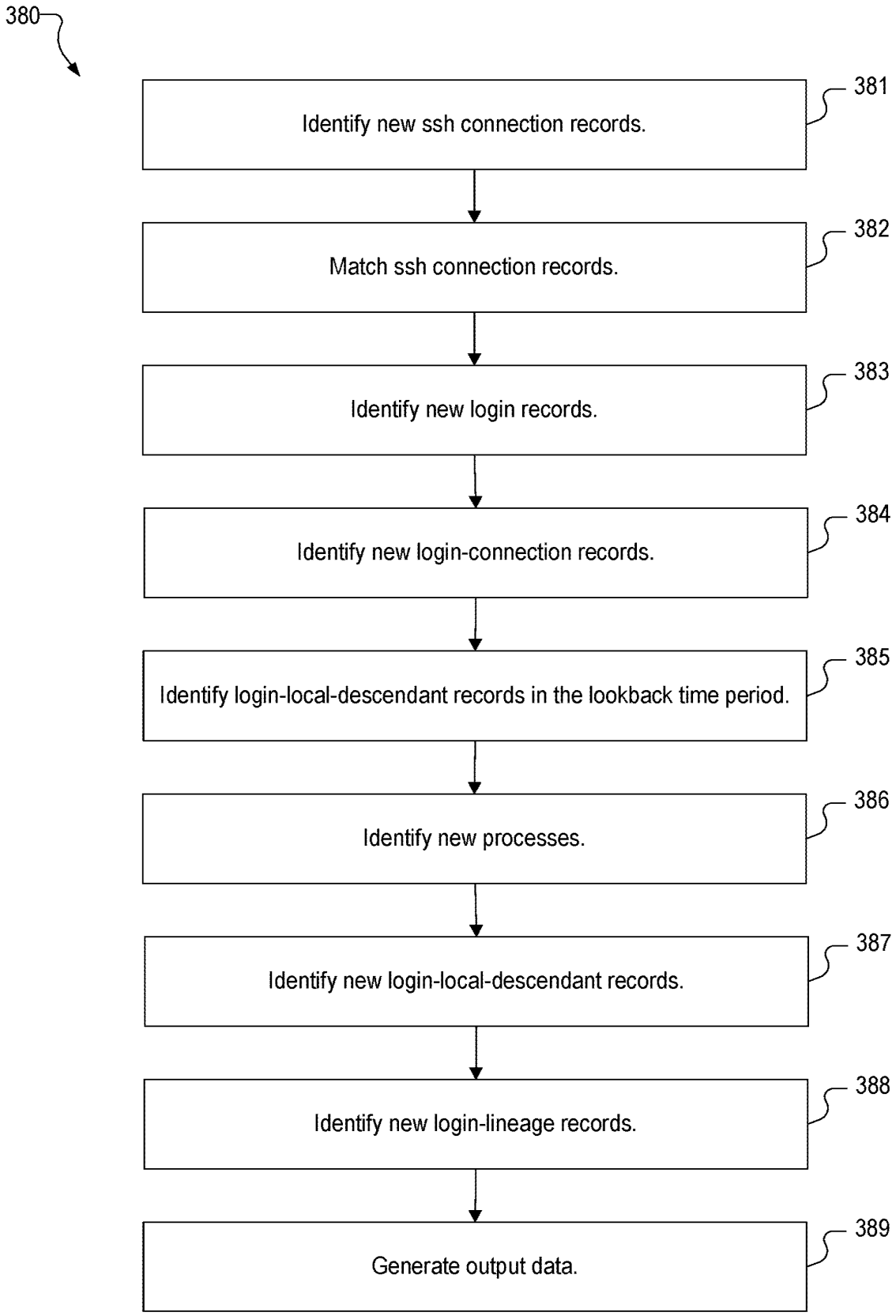

380
Identify new ssh connection records.
381
Match ssh connection records.
382
Identify new login records.
383
Identify new login-connection records.
384
Identify login-local-descendant records in the lookback time period.
385
Identify new processes.
386
Identify new login-local-descendant records.
387
Identify new login-lineage records.
388
Generate output data.
389

Fig. 3J

MID	start_time	PID_hash	src_IP_addr	src_port	dst_IP_addr	dst_port	prot	dir
A	t1	A1	1.1.1.10	10000	2.2.2.20	22	TCP	Incoming
A	t2	A3	2.2.2.20	10001	2.2.2.21	22	TCP	Outgoing
B	t2	B1	2.2.2.20	10001	2.2.2.21	22	TCP	Incoming

390

Fig. 3K

391

src_MID	src_PID_hash	dst_MID	dst_PID_hash	dst_start_time	src_IP_addr	src_port	dst_IP_addr	dst_port
null	null	A	A1	t1	1.1.1.10	10000	2.2.2.20	22
A	A3	B	B1	t2	2.2.2.20	10001	2.2.2.21	22

Fig. 3L

MID	login_time	sshd_PID_hash
A	t1	A1
B	t2	B1

Fig. 3M

MID	sshd_PID_hash	login_time	login_username	src_IP_addr	src_port	dst_IP_addr	dst_port
A	A1	t1	X	1.1.1.10	10000	2.2.2.20	22
B	B1	t2	Y	2.2.2.20	10001	2.2.2.21	22

Fig. 3N

MID	start_time	PID_hash	exe_path	parent_PID_hash
A	t1	A1	/usr/sbin/sshd	A0
A	t1	A2	/bin/bash	A1
A	t2	A3	/usr/bin/ssh	A2
B	t2	B1	/usr/sbin/sshd	B0
B	t2	B2	/bin/bash	B1
B	t3	B3	/usr/bin/curl	B2

Fig. 3O

MID	sshd_PID_hash	PID_hash
A	A1	A1
A	A1	A2
A	A1	A3
B	B1	B1
B	B1	B2
B	B1	B3

Fig. 3P

parent_MID	parent_sshd_PID_hash	child_MID	origin_sshd_PID_hash
A	A1	B	B1

Fig. 3Q

MID	sshd_PID_hash	parent_MID	parent_sshd_PID_hash	origin_MID	origin_sshd_PID_hash
A	A1	null	null	A	A1
B	B1	A	A1	A	A1

Fig. 3R

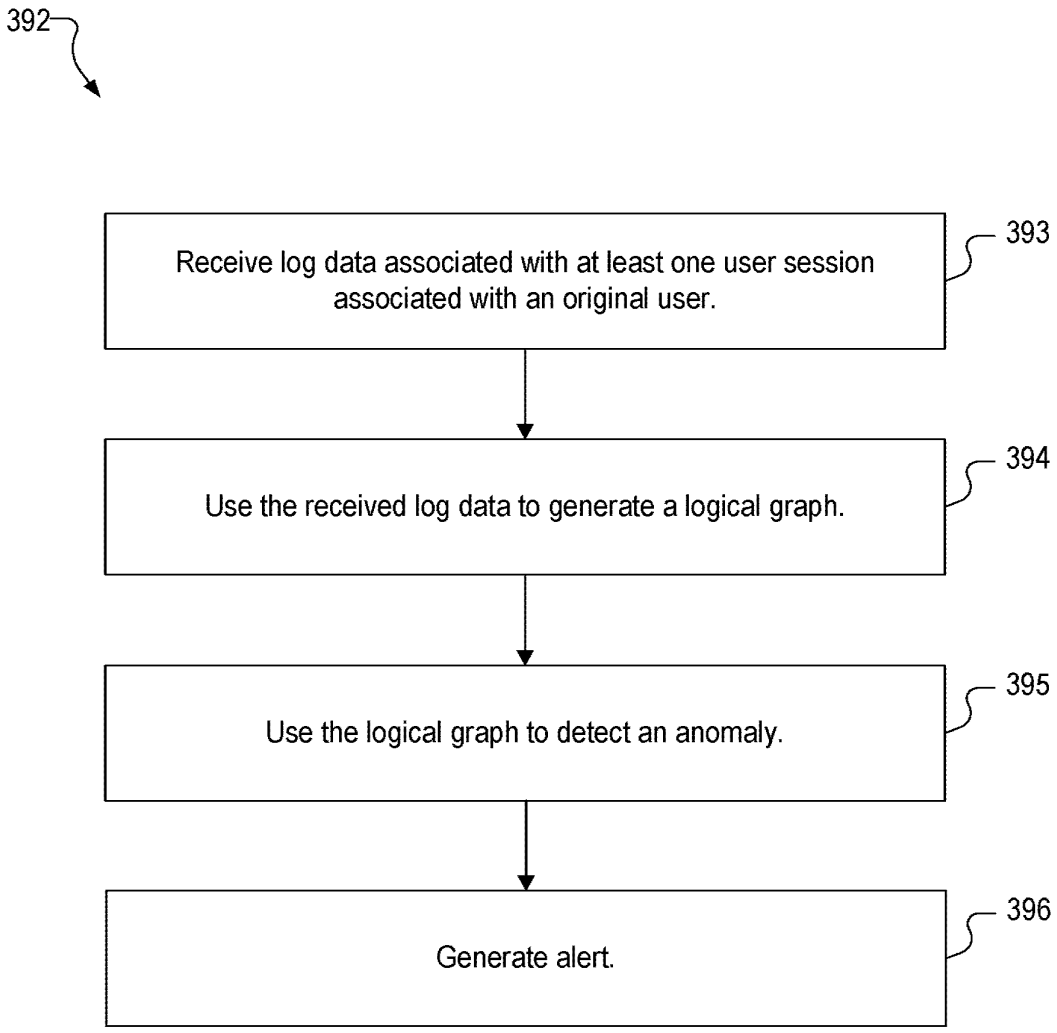

392
Receive log data associated with at least one user session associated with an original user.
393
Use the received log data to generate a logical graph.
394
Use the logical graph to detect an anomaly.
395
Generate alert.
396

Fig. 3S

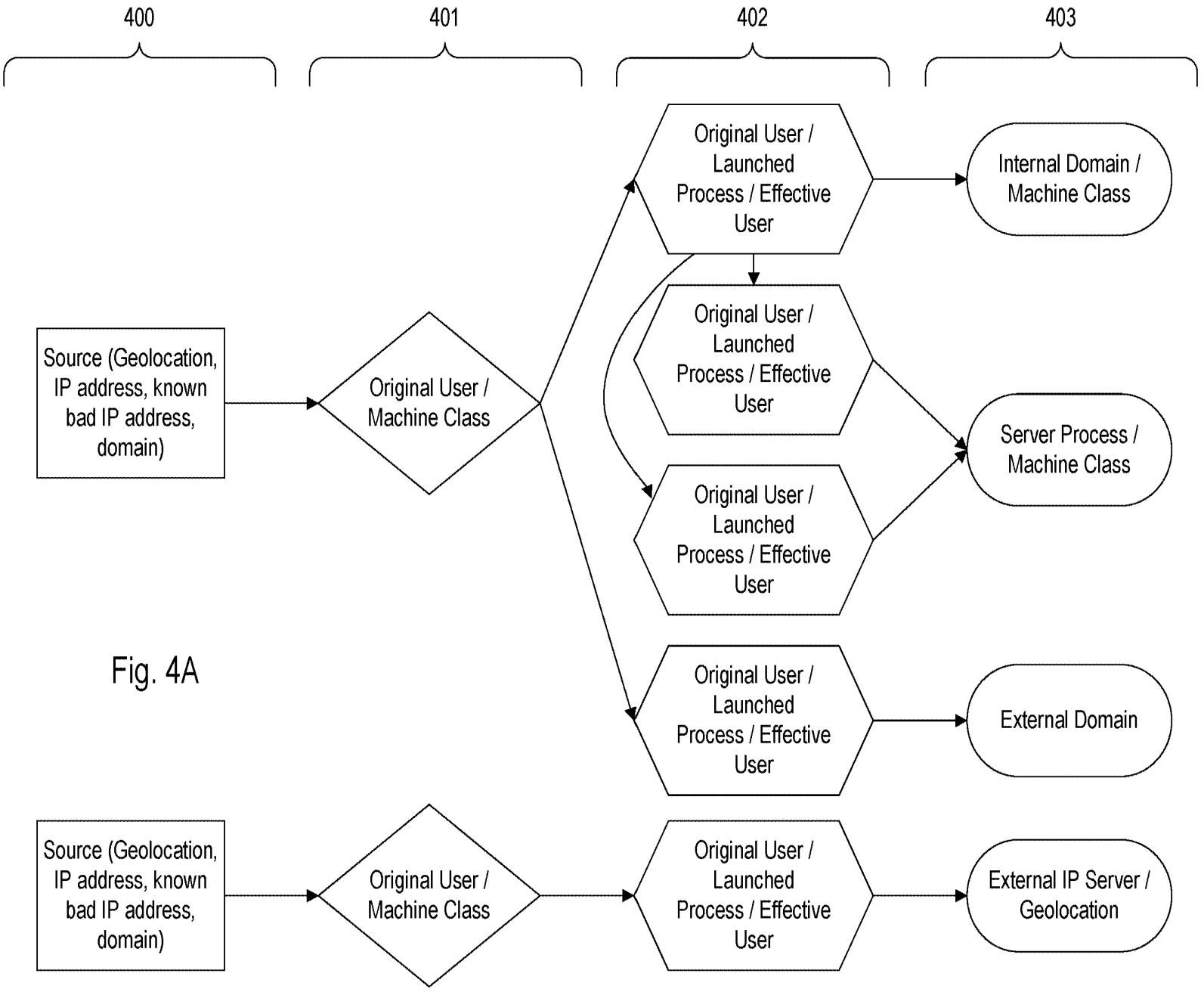


Fig. 4A

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00745

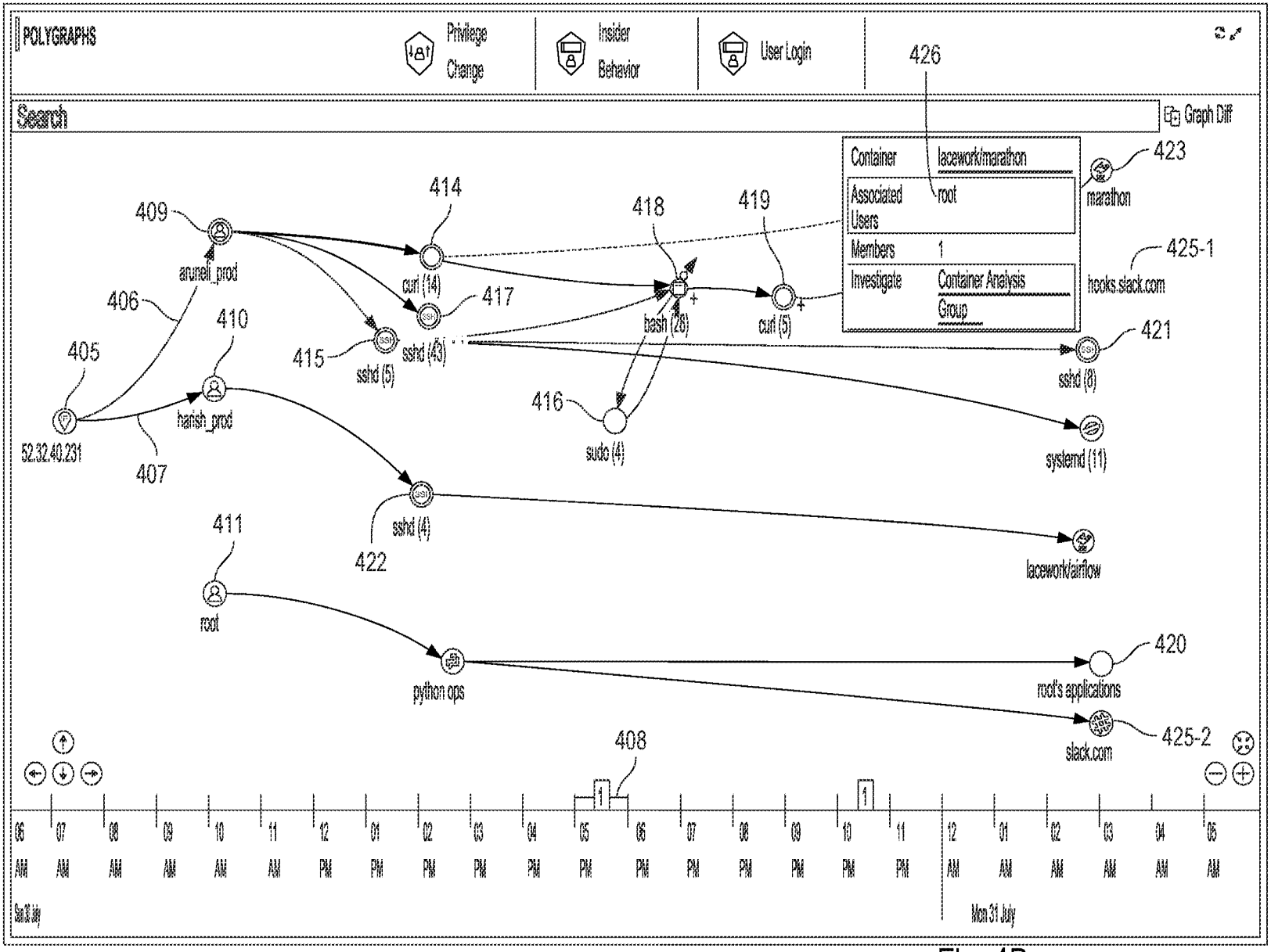

404
POLYGRAPHS
Privilege Change
Insider Behavior
User Login
426
Search
Graph Diff
Container
lacework/marathon
Associated Users
root
Members
1
Investigate
Container Analysis
Group
423
marathon
425-1
hooks.slack.com
409
414
418
419
aruneli_prod
curl (14)
bash (26)
curl (5)
406
417
425
410
421
sshd (5)
sshd (43)
sshd (8)
405
415
416
harish_prod
sudo (4)
52.32.40.231
407
systemd (11)
411
sshd (4)
422
lacework/airflow
root
python ops
420
roof's applications
425-2
slack.com
408
06 AM
07 AM
08 AM
09 AM
10 AM
11 AM
12 PM
01 PM
02 PM
03 PM
04 PM
05 PM
06 PM
07 PM
08 PM
09 PM
10 PM
11 PM
12 AM
01 AM
02 AM
03 AM
04 AM
05 AM
Sun 30 July
Mon 31 July

Fig. 4B

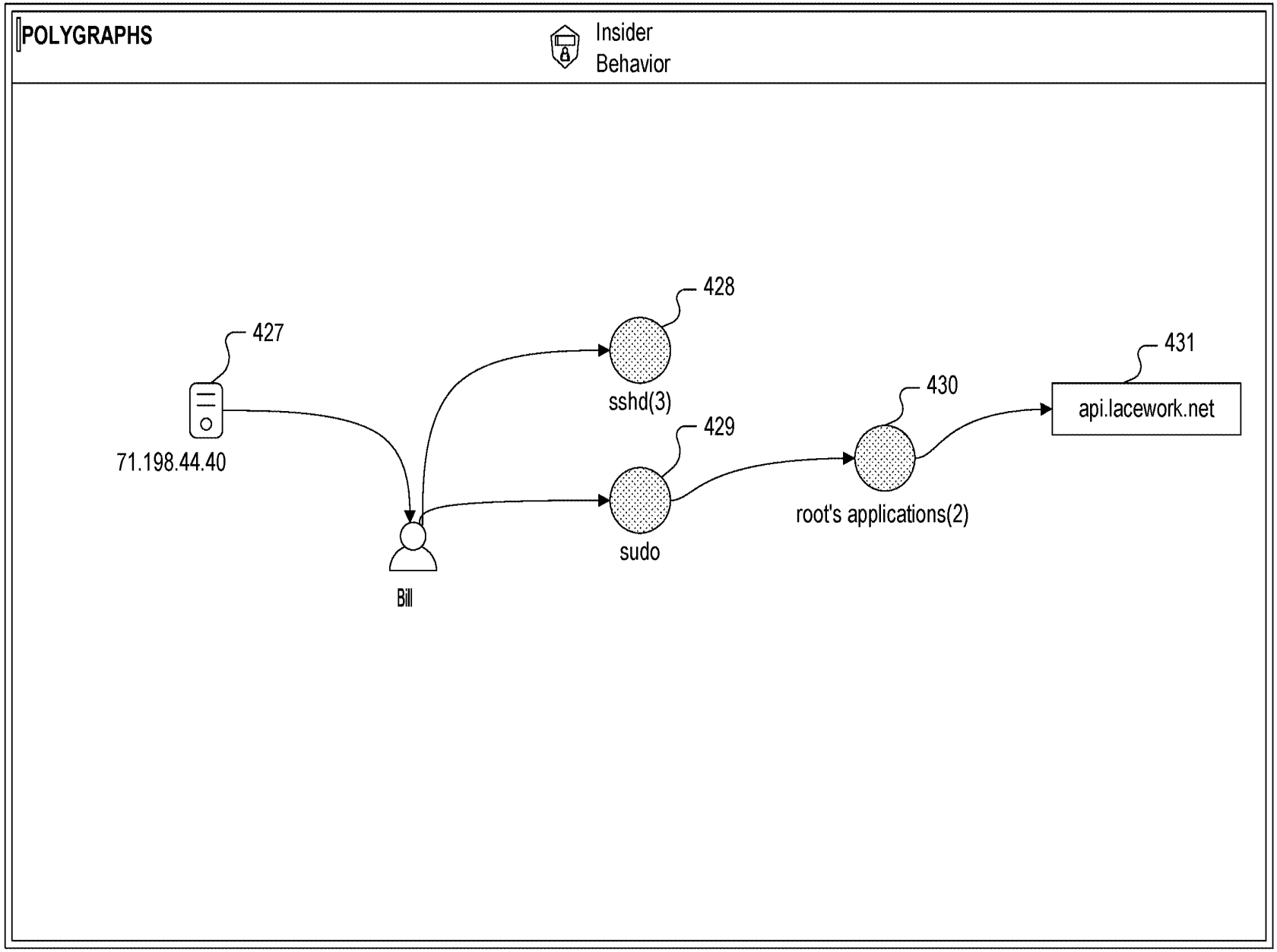

POLYGRAPHS
Insider
Behavior
427
71.198.44.40
428
sshd(3)
429
sudo
Bill
430
root's applications(2)
431
api.lacework.net

Fig. 4C

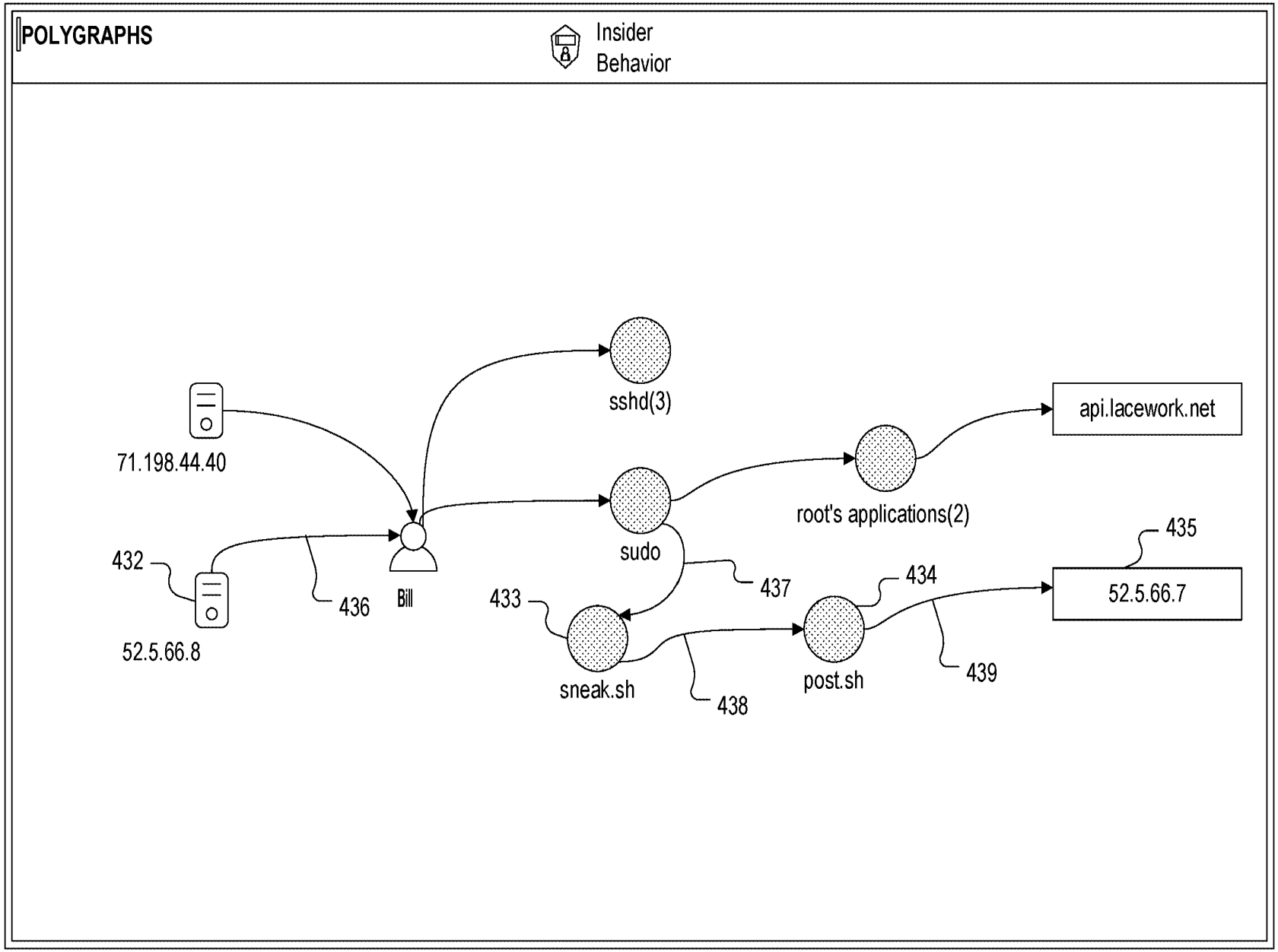

POLYGRAPHS
Insider Behavior
71.198.44.40
sshd(3)
api.lacework.net
root's applications(2)
sudo
435
432
436
Bill
437
434
433
52.5.66.7
52.5.66.8
sneak.sh
post.sh
439
438

Fig. 4D

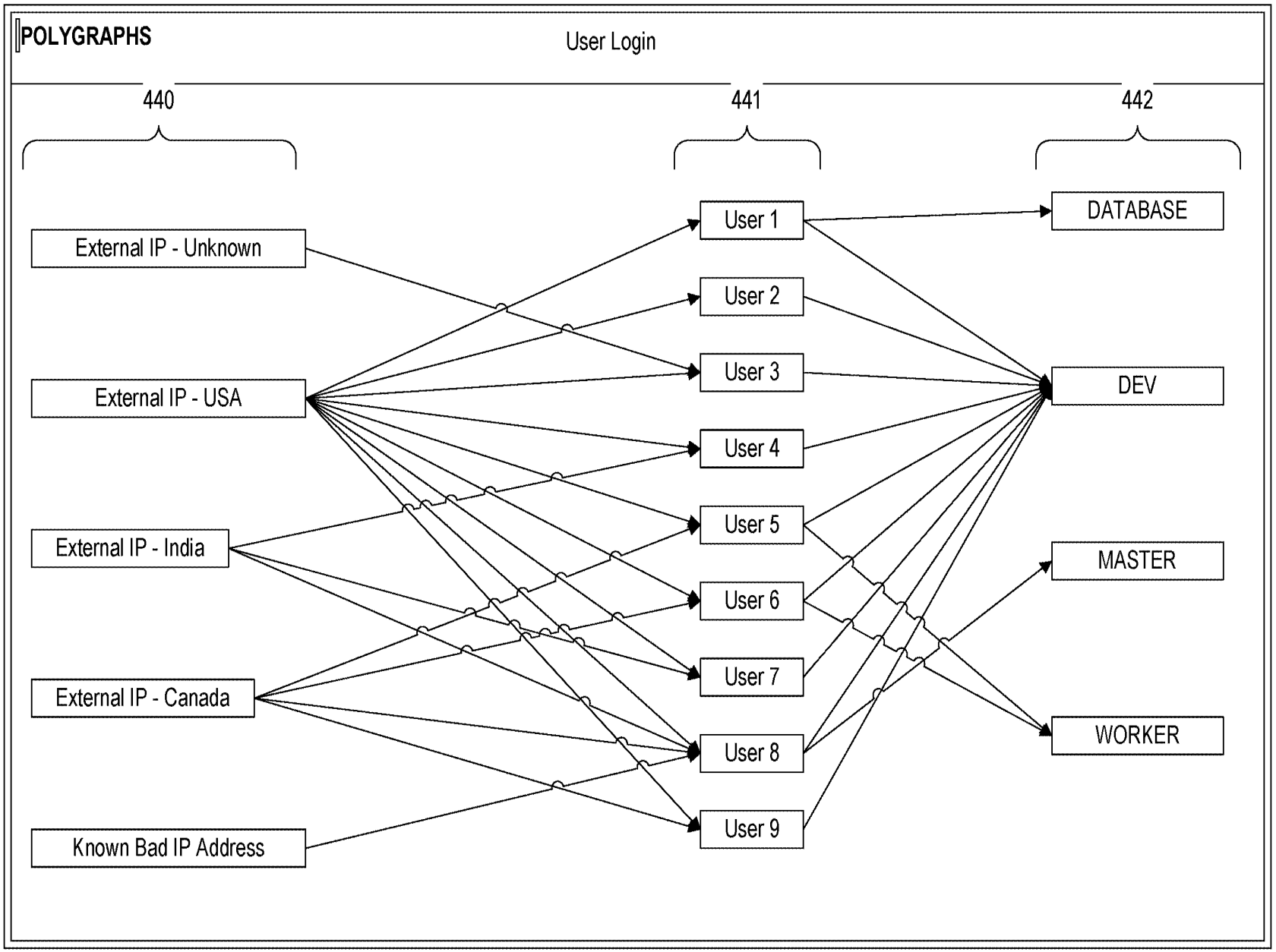


Fig. 4E

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00749

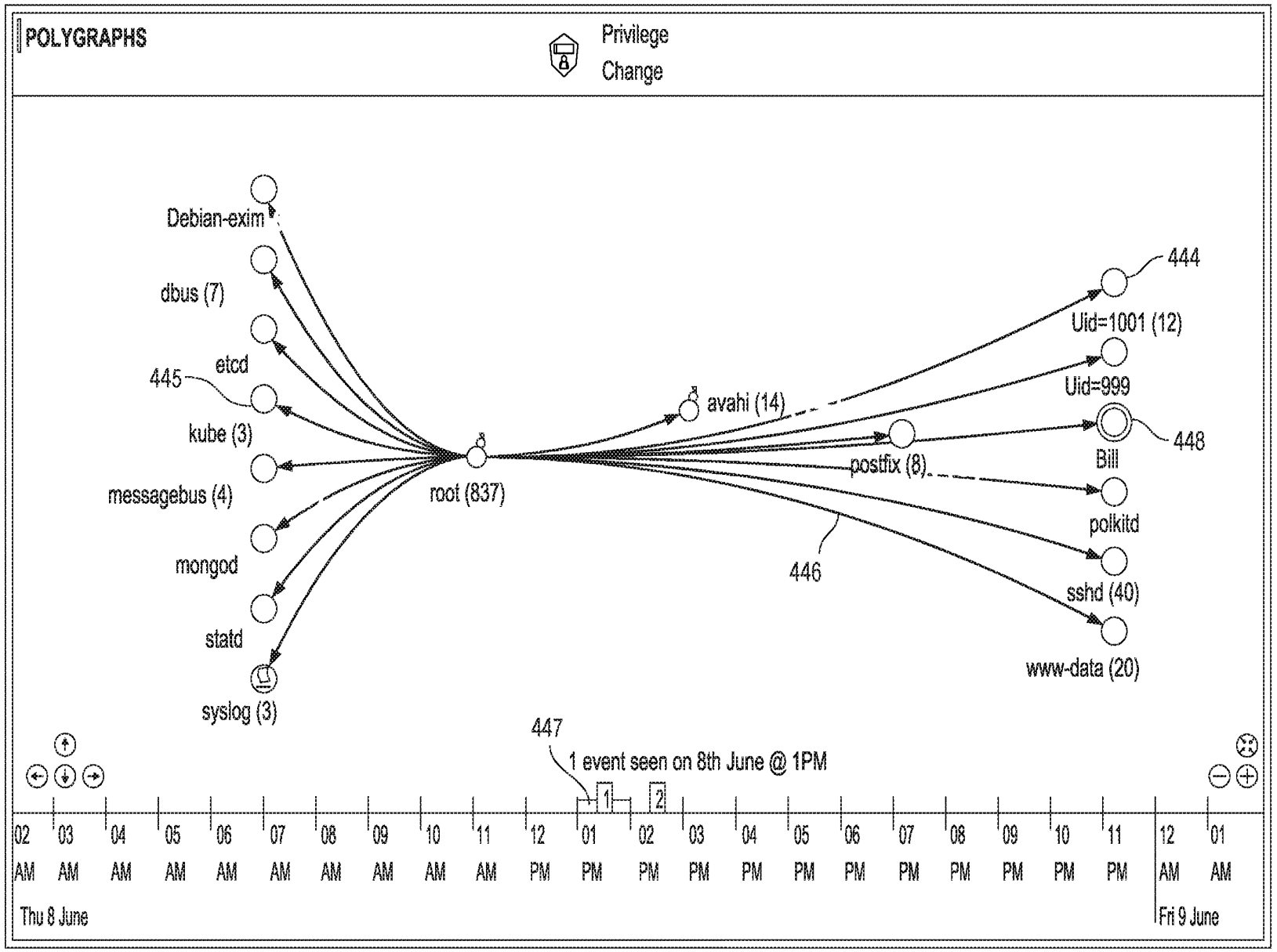

443
POLYGRAPHS
Privilege Change
Debian-exim
dbus (7)
etcd
445
kube (3)
messagebus (4)
mongod
statd
syslog (3)
root (837)
avahi (14)
postfix (8)
446
444
Uid=1001 (12)
Uid=999
448
Bill
polkitd
sshd (40)
www-data (20)
447
1 event seen on 8th June @ 1PM
1
2
02 AM 03 AM 04 AM 05 AM 06 AM 07 AM 08 AM 09 AM 10 AM 11 AM 12 PM 01 PM 02 PM 03 PM 04 PM 05 PM 06 PM 07 PM 08 PM 09 PM 10 PM 11 PM 12 AM 01 AM
Thu 8 June
Fri 9 June

Fig. 4F

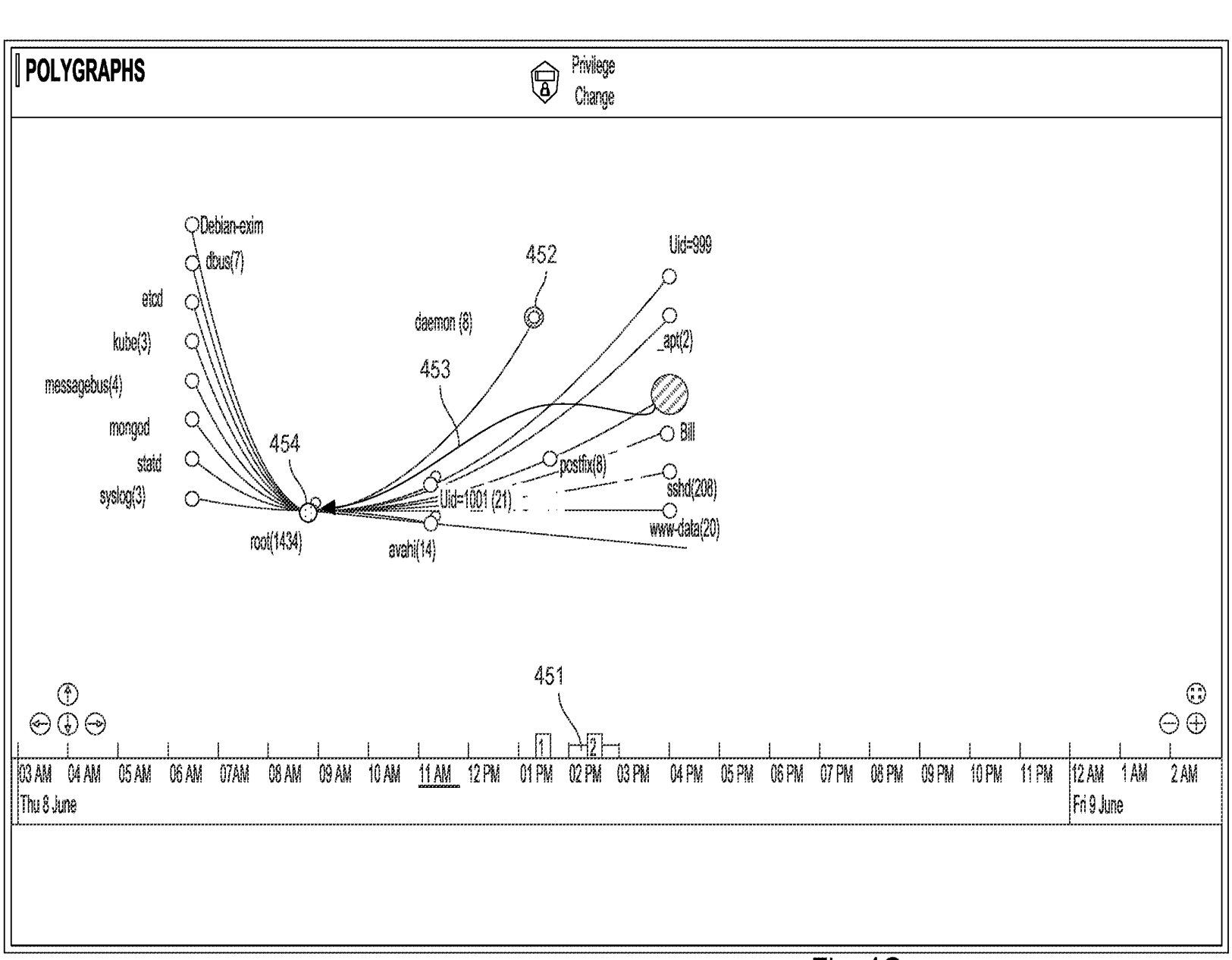

450
POLYGRAPHS
Privilege Change
Debian-exim
dbus(7)
etcd
kube(3)
messagebus(4)
mongod
statd
syslog(3)
root(1434)
452
daemon (8)
453
454
Uid=1001 (21)
avahi(14)
Uid=999
_apt(2)
Bill
postfix(8)
sshd(208)
www-data(20)
451
03 AM 04 AM 05 AM 06 AM 07AM 08 AM 09 AM 10 AM 11 AM 12 PM 01 PM 02 PM 03 PM 04 PM 05 PM 06 PM 07 PM 08 PM 09 PM 10 PM 11 PM 12 AM 1 AM 2 AM
Thu 8 June
Fri 9 June

Fig. 4G

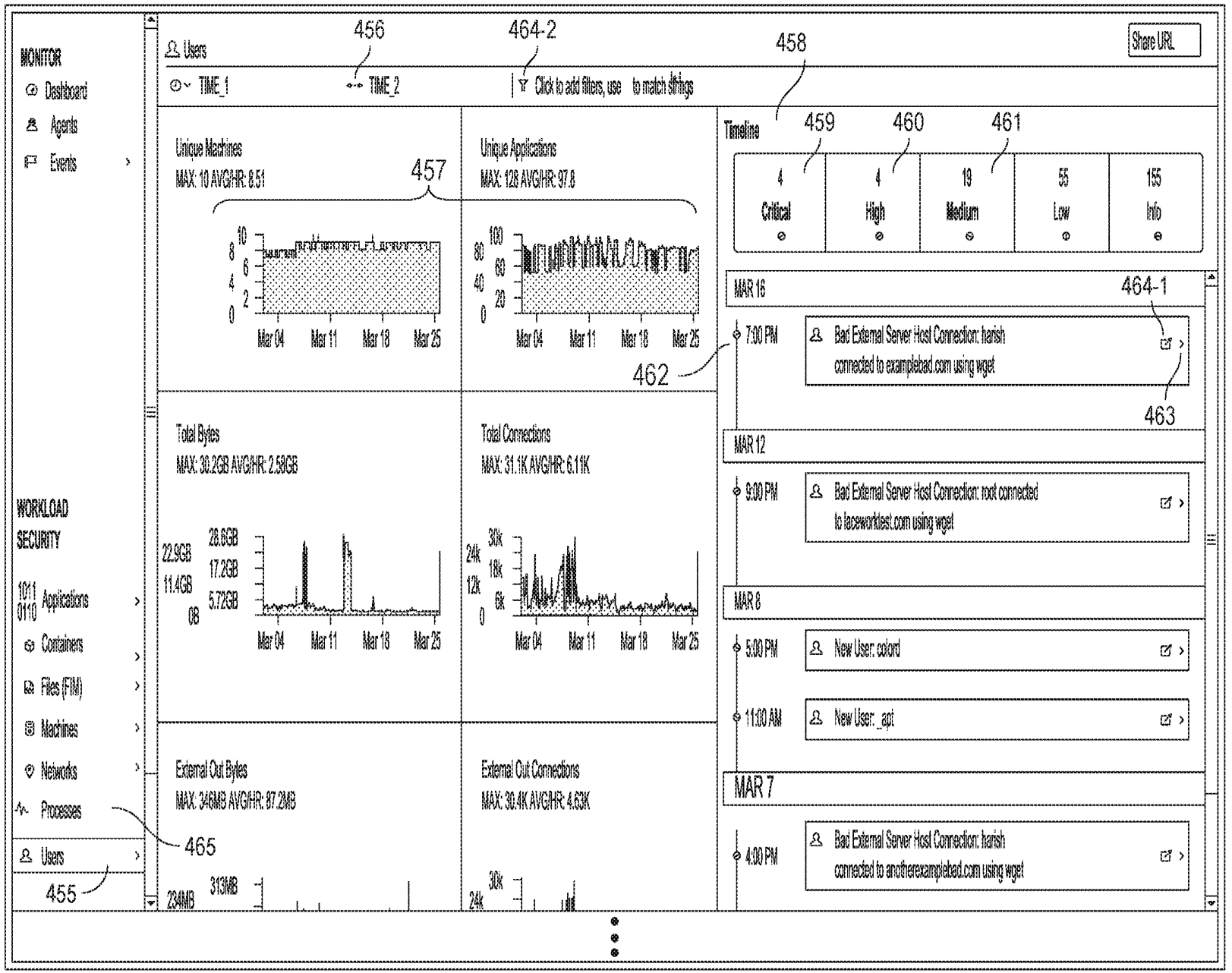


Fig. 4H

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00752

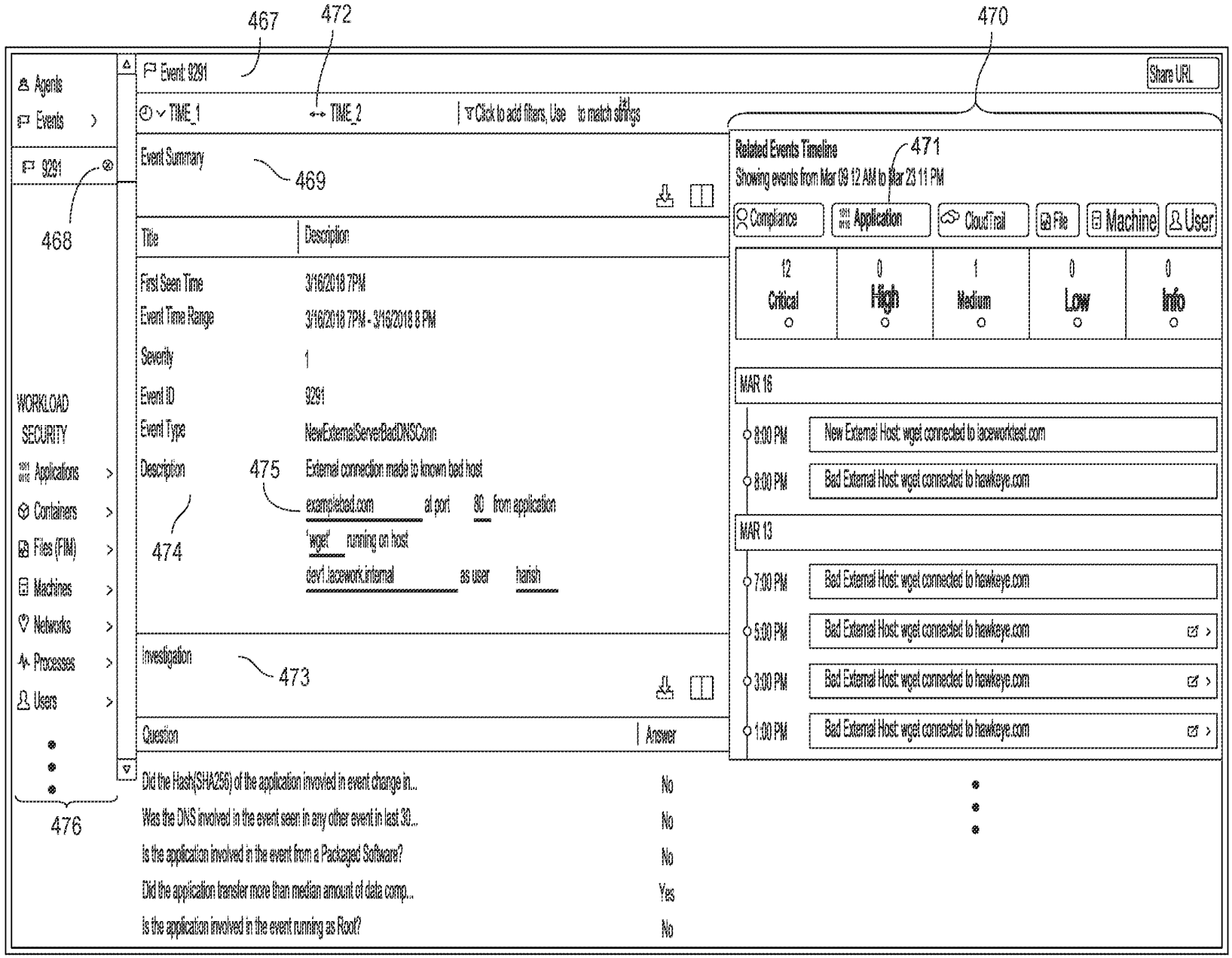


Fig. 4I

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00753

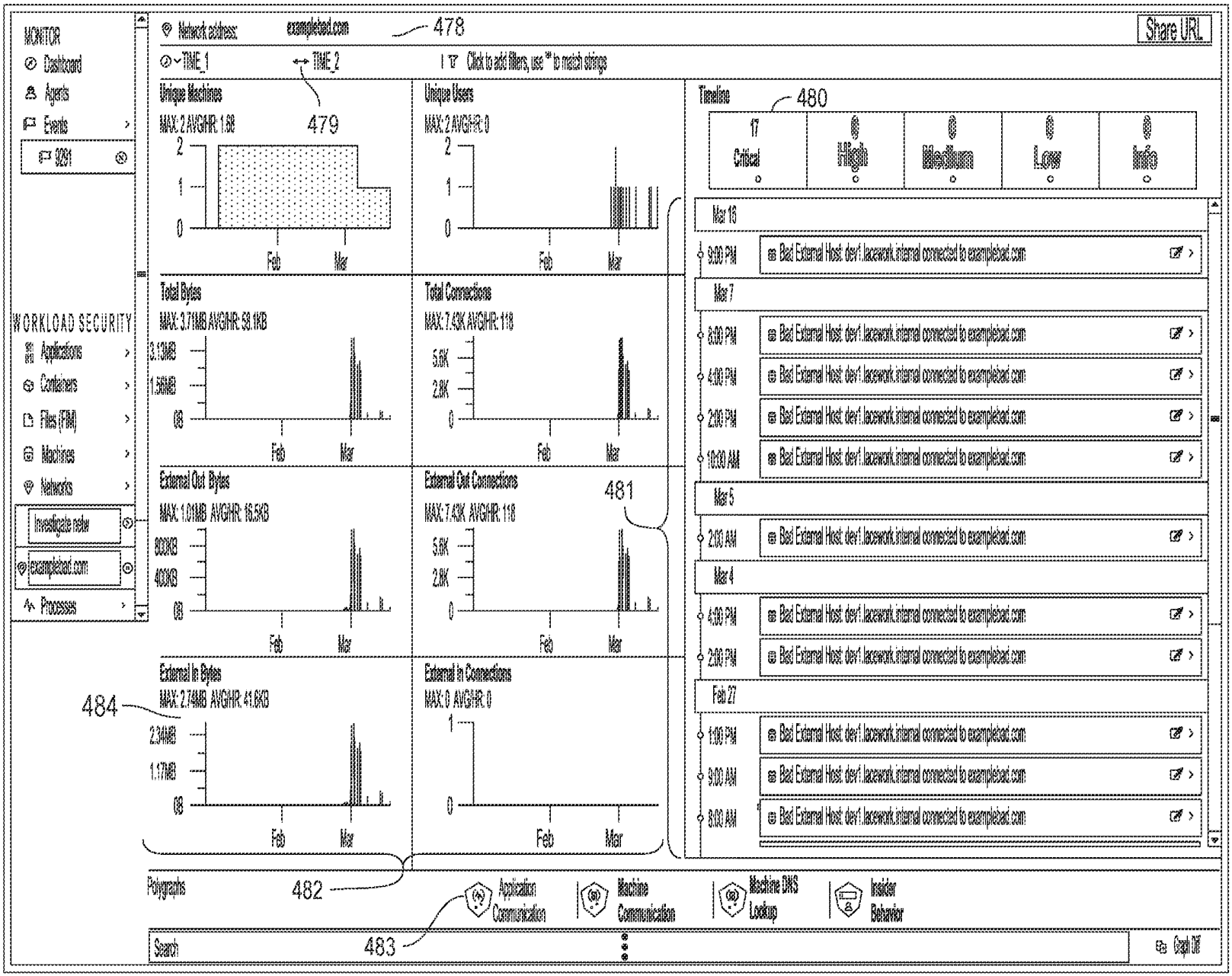


Fig. 4J

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00754

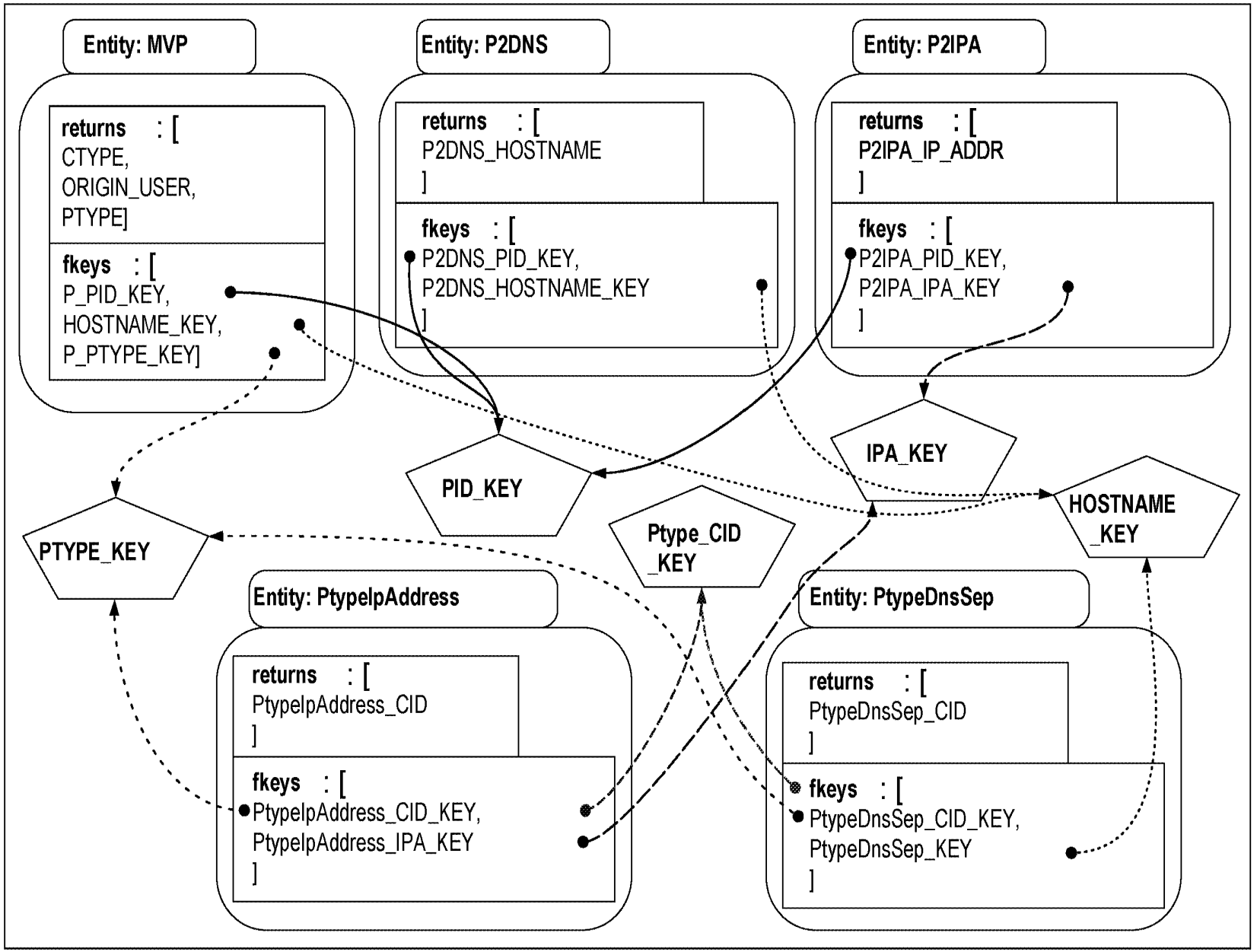


Fig. 4K

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00755

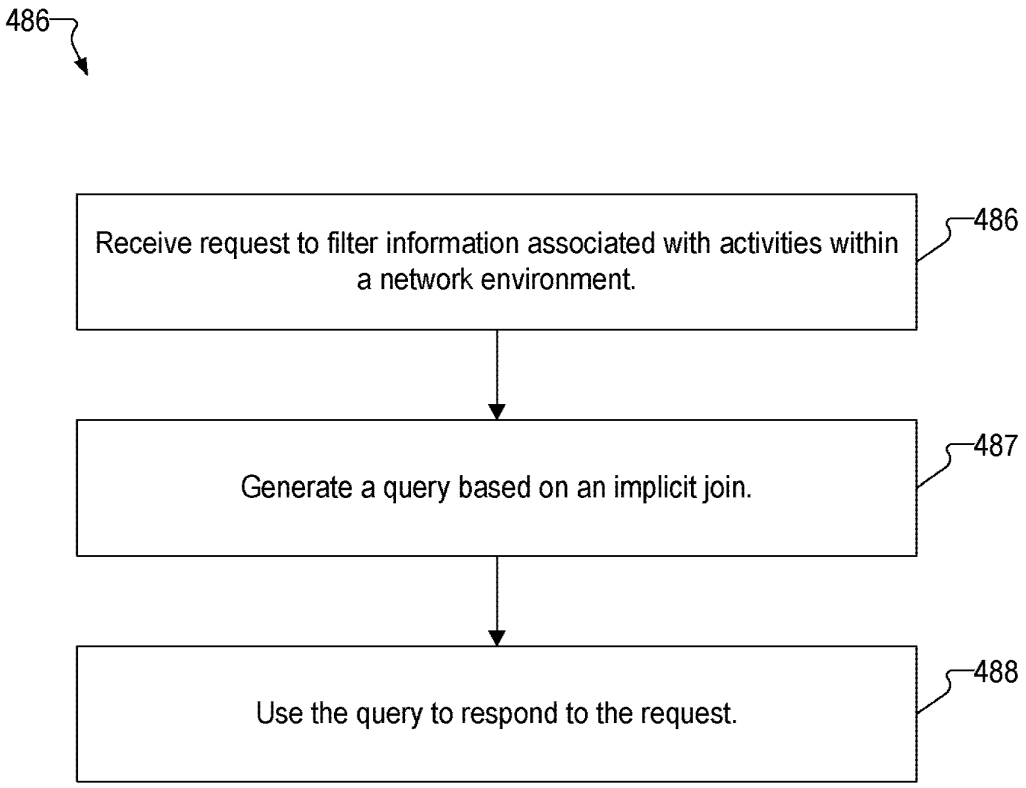
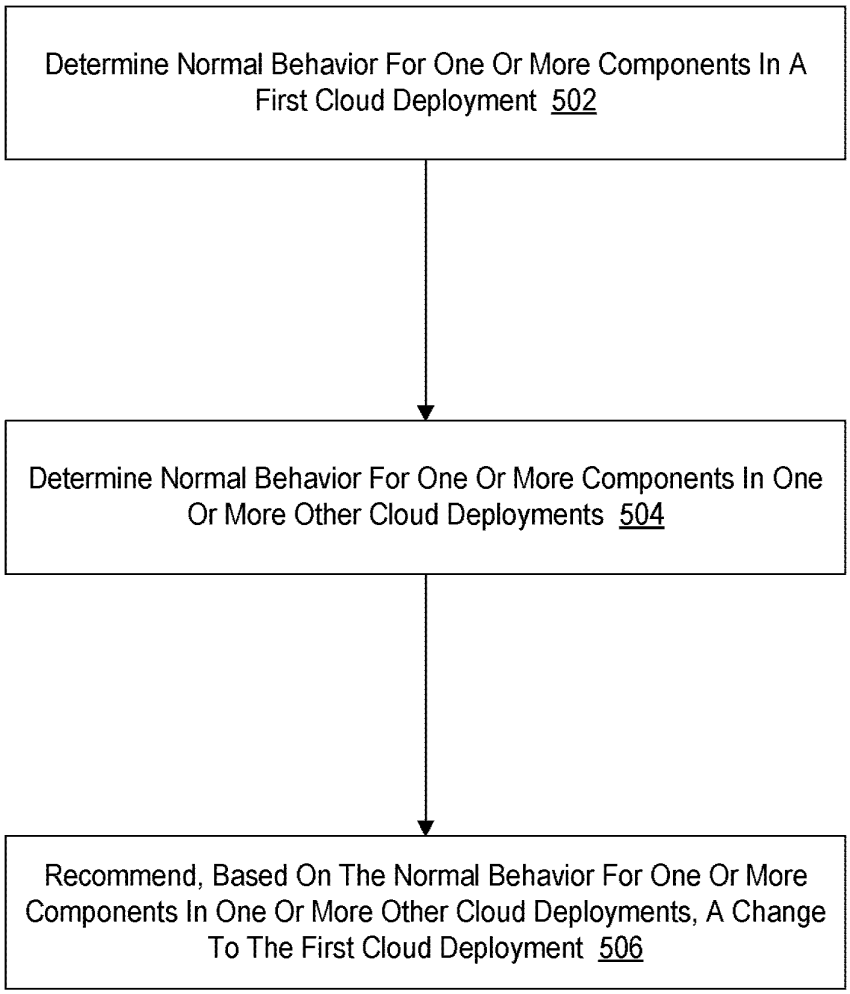

486
Receive request to filter information associated with activities within a network environment.
486
Generate a query based on an implicit join.
487
Use the query to respond to the request.
488

Fig. 4L


Determine Normal Behavior For One Or More Components In A First Cloud Deployment 502
Determine Normal Behavior For One Or More Components In One Or More Other Cloud Deployments 504
Recommend, Based On The Normal Behavior For One Or More Components In One Or More Other Cloud Deployments, A Change To The First Cloud Deployment 506

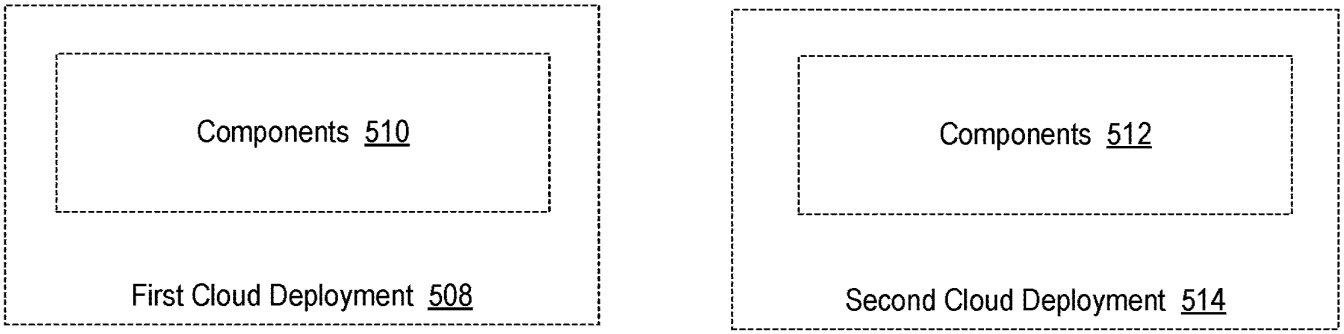

Components 510
First Cloud Deployment 508
Components 512
Second Cloud Deployment 514

Fig. 5

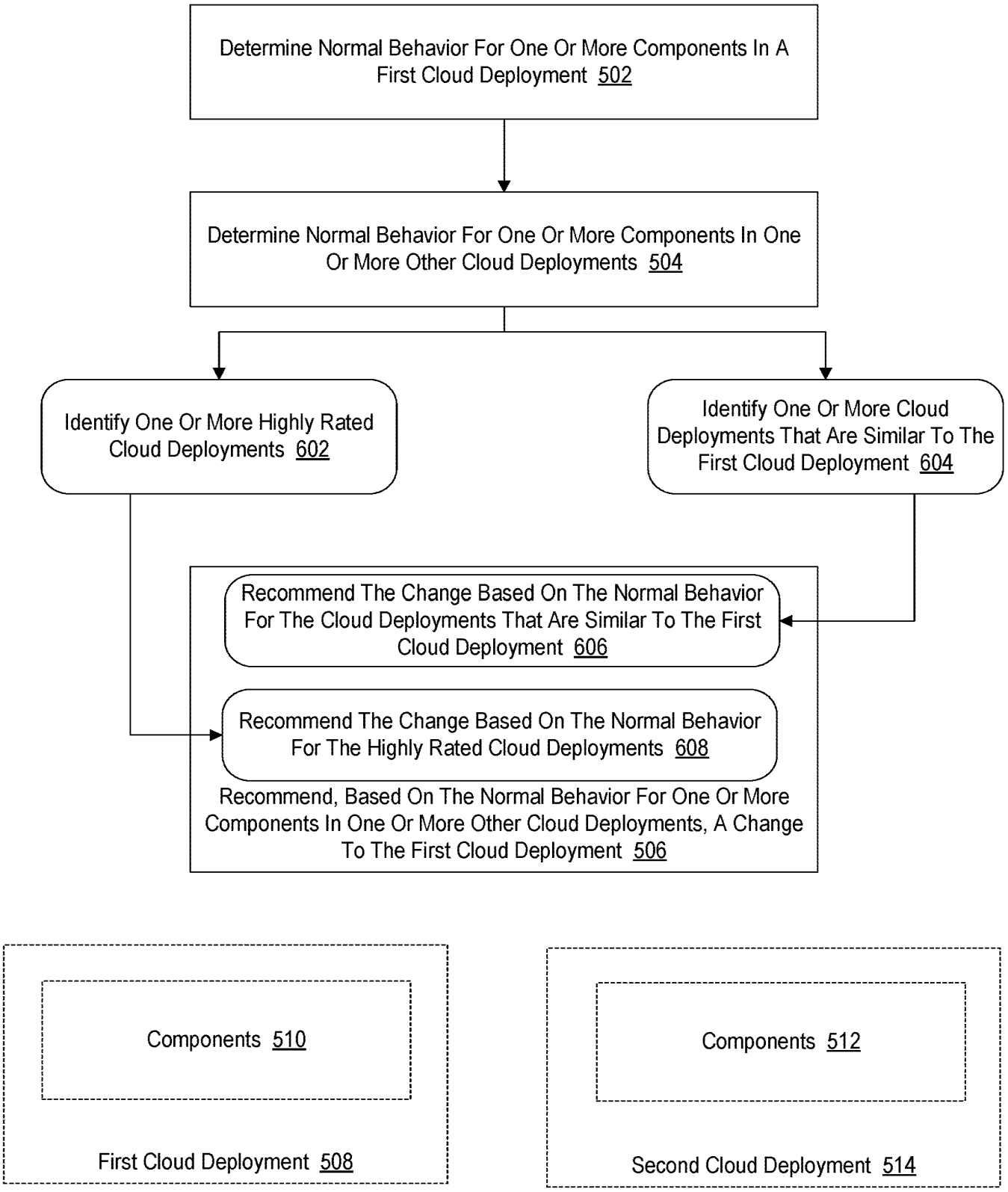

Determine Normal Behavior For One Or More Components In A First Cloud Deployment 502
Determine Normal Behavior For One Or More Components In One Or More Other Cloud Deployments 504
Identify One Or More Highly Rated Cloud Deployments 602
Identify One Or More Cloud Deployments That Are Similar To The First Cloud Deployment 604
Recommend The Change Based On The Normal Behavior For The Cloud Deployments That Are Similar To The First Cloud Deployment 606
Recommend The Change Based On The Normal Behavior For The Highly Rated Cloud Deployments 608
Recommend, Based On The Normal Behavior For One Or More Components In One Or More Other Cloud Deployments, A Change To The First Cloud Deployment 506
Components 510
First Cloud Deployment 508
Components 512
Second Cloud Deployment 514

Fig. 6

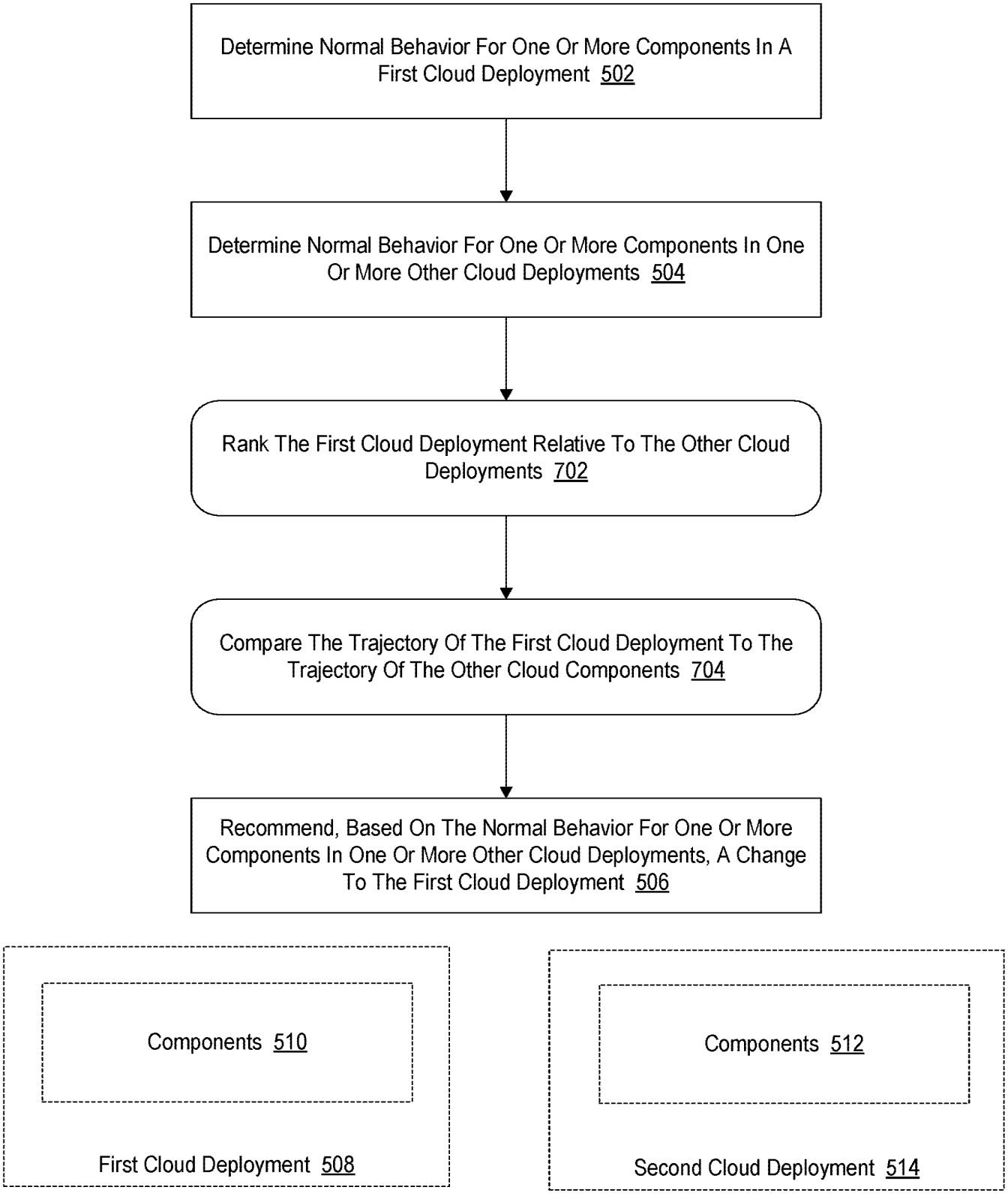

Determine Normal Behavior For One Or More Components In A First Cloud Deployment 502
Determine Normal Behavior For One Or More Components In One Or More Other Cloud Deployments 504
Rank The First Cloud Deployment Relative To The Other Cloud Deployments 702
Compare The Trajectory Of The First Cloud Deployment To The Trajectory Of The Other Cloud Components 704
Recommend, Based On The Normal Behavior For One Or More Components In One Or More Other Cloud Deployments, A Change To The First Cloud Deployment 506
Components 510
First Cloud Deployment 508
Components 512
Second Cloud Deployment 514

Fig. 5

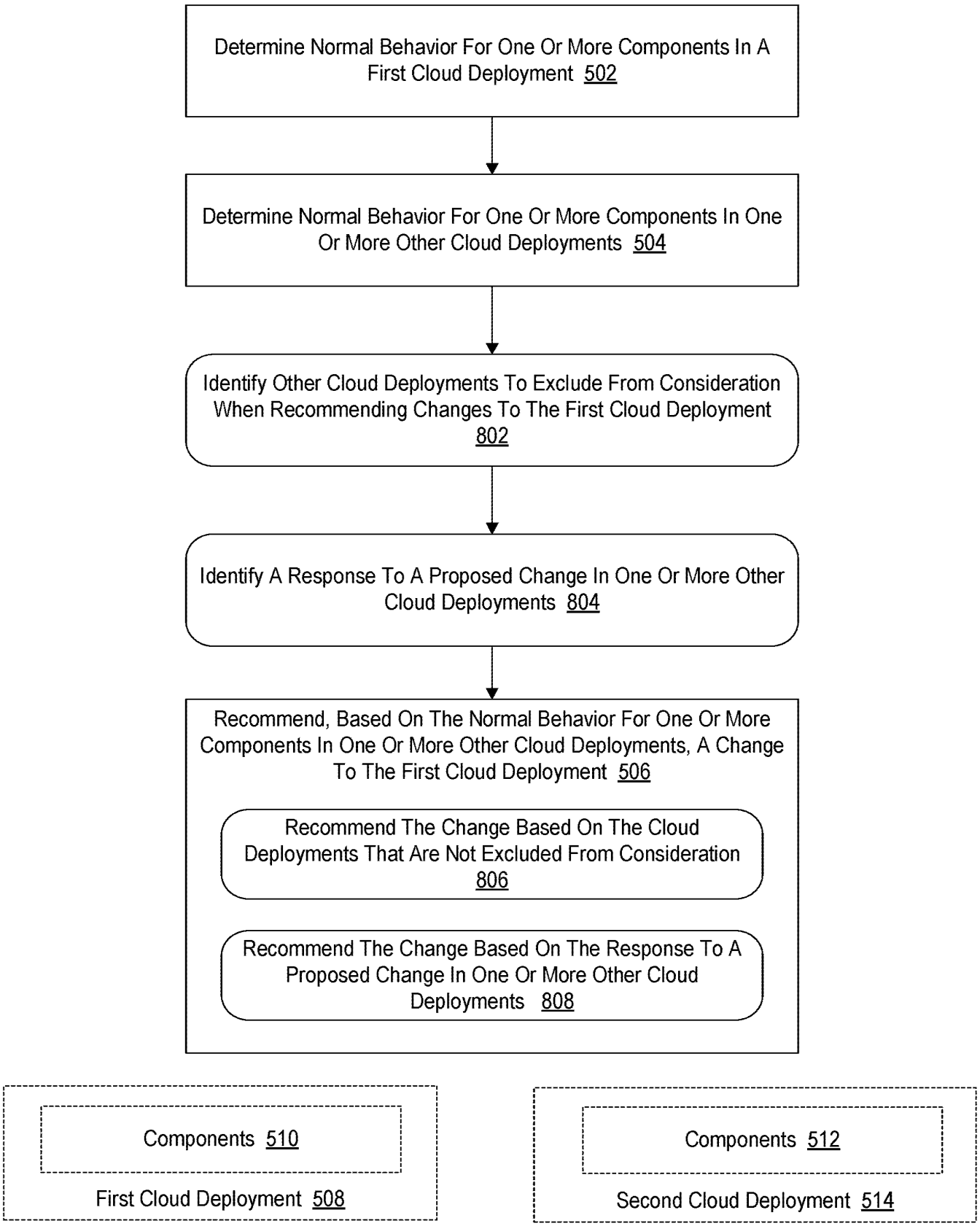

Determine Normal Behavior For One Or More Components In A First Cloud Deployment 502
Determine Normal Behavior For One Or More Components In One Or More Other Cloud Deployments 504
Identify Other Cloud Deployments To Exclude From Consideration When Recommending Changes To The First Cloud Deployment 802
Identify A Response To A Proposed Change In One Or More Other Cloud Deployments 804
Recommend, Based On The Normal Behavior For One Or More Components In One Or More Other Cloud Deployments, A Change To The First Cloud Deployment 506
Recommend The Change Based On The Cloud Deployments That Are Not Excluded From Consideration 806
Recommend The Change Based On The Response To A Proposed Change In One Or More Other Cloud Deployments 808
Components 510
First Cloud Deployment 508
Components 512
Second Cloud Deployment 514

Fig. 8

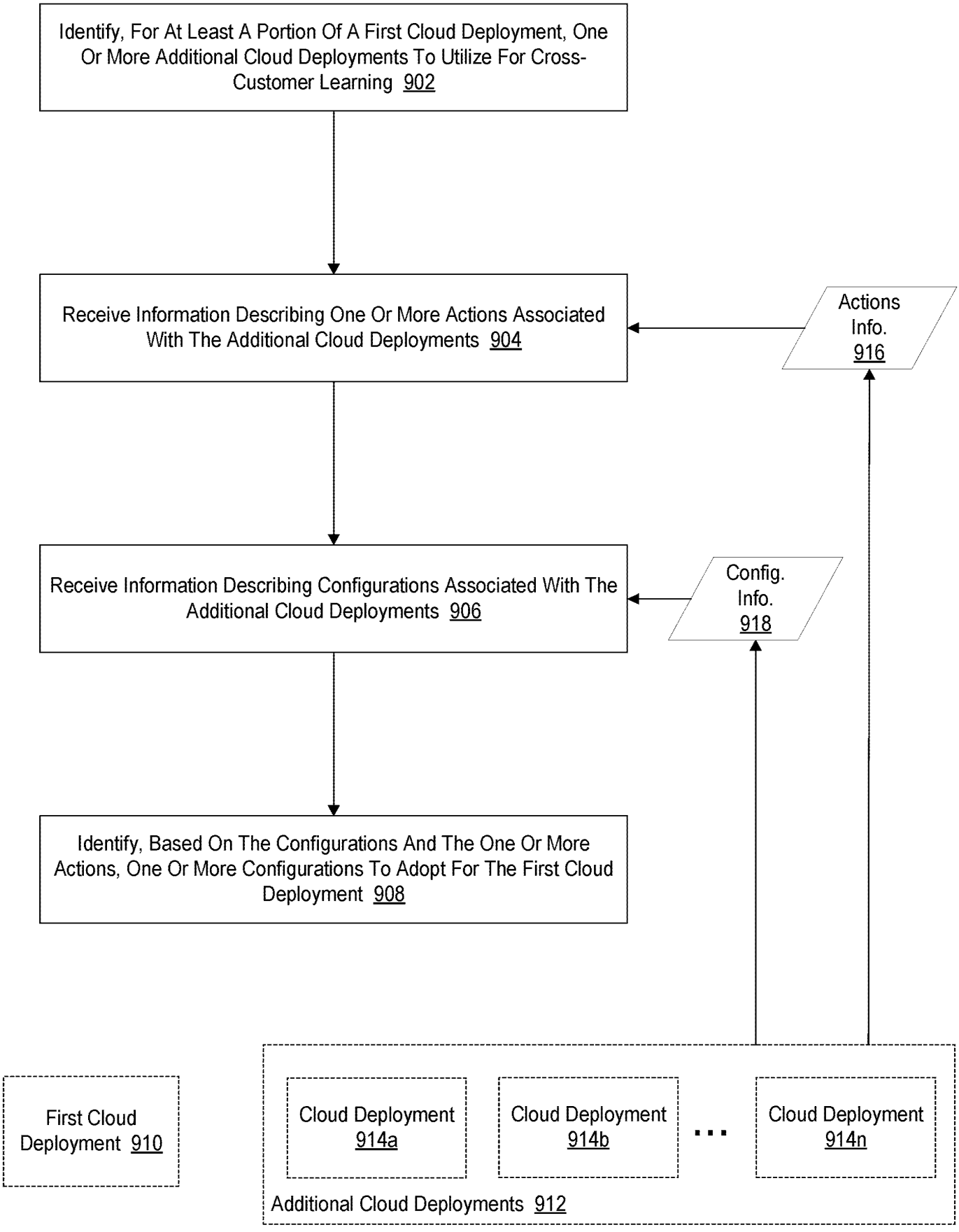

Identify, For At Least A Portion Of A First Cloud Deployment, One Or More Additional Cloud Deployments To Utilize For Cross-Customer Learning 902
Receive Information Describing One Or More Actions Associated With The Additional Cloud Deployments 904
Actions Info. 916
Receive Information Describing Configurations Associated With The Additional Cloud Deployments 906
Config. Info. 918
Identify, Based On The Configurations And The One Or More Actions, One Or More Configurations To Adopt For The First Cloud Deployment 908
First Cloud Deployment 910
Cloud Deployment 914a
Cloud Deployment 914b
...
Cloud Deployment 914n
Additional Cloud Deployments 912

Fig. 9

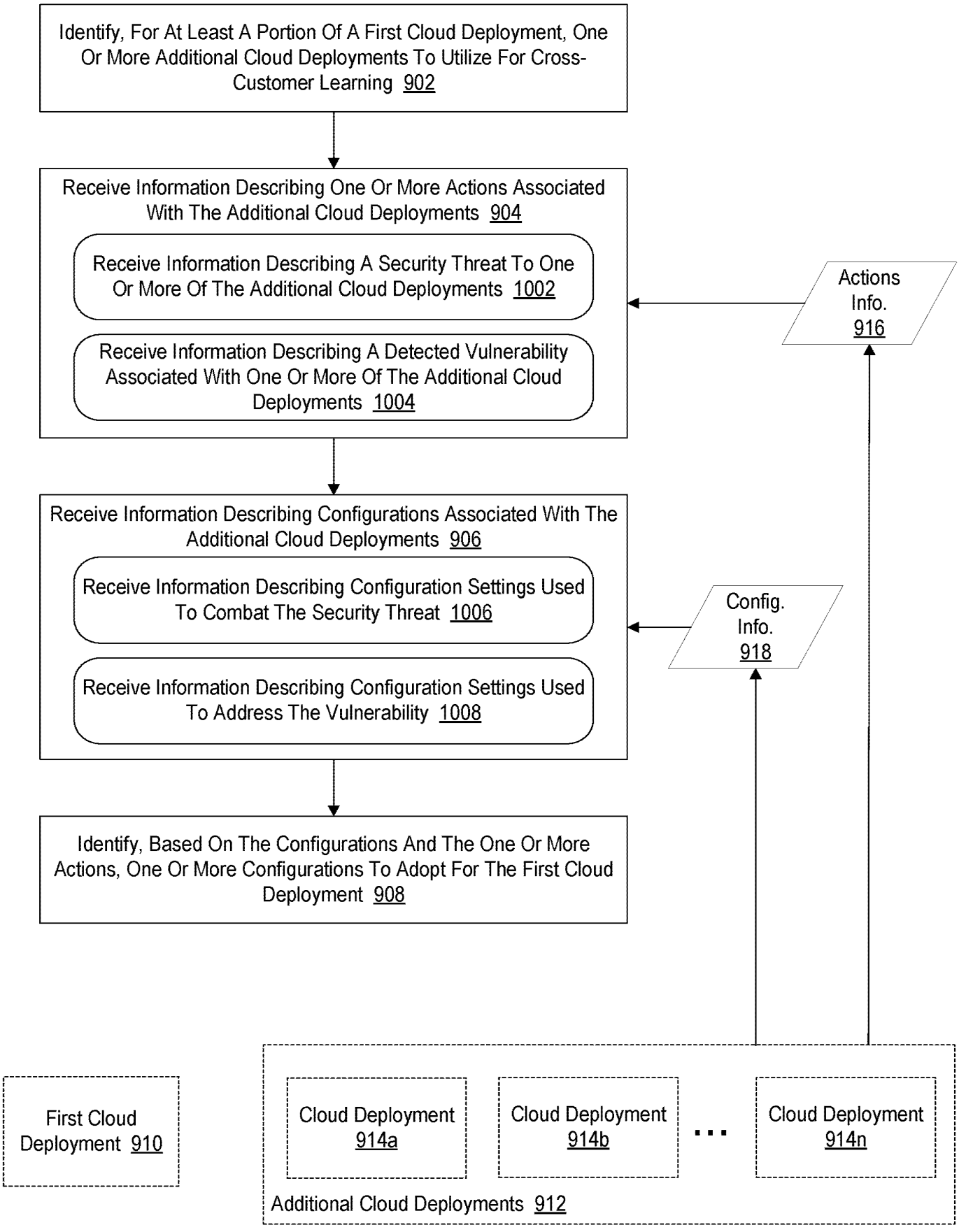

Identify, For At Least A Portion Of A First Cloud Deployment, One Or More Additional Cloud Deployments To Utilize For Cross-Customer Learning 902
Receive Information Describing One Or More Actions Associated With The Additional Cloud Deployments 904
Receive Information Describing A Security Threat To One Or More Of The Additional Cloud Deployments 1002
Receive Information Describing A Detected Vulnerability Associated With One Or More Of The Additional Cloud Deployments 1004
Actions Info. 916
Receive Information Describing Configurations Associated With The Additional Cloud Deployments 906
Receive Information Describing Configuration Settings Used To Combat The Security Threat 1006
Receive Information Describing Configuration Settings Used To Address The Vulnerability 1008
Config. Info. 918
Identify, Based On The Configurations And The One Or More Actions, One Or More Configurations To Adopt For The First Cloud Deployment 908
First Cloud Deployment 910
Cloud Deployment 914a
Cloud Deployment 914b
...
Cloud Deployment 914n
Additional Cloud Deployments 912

Fig. 10

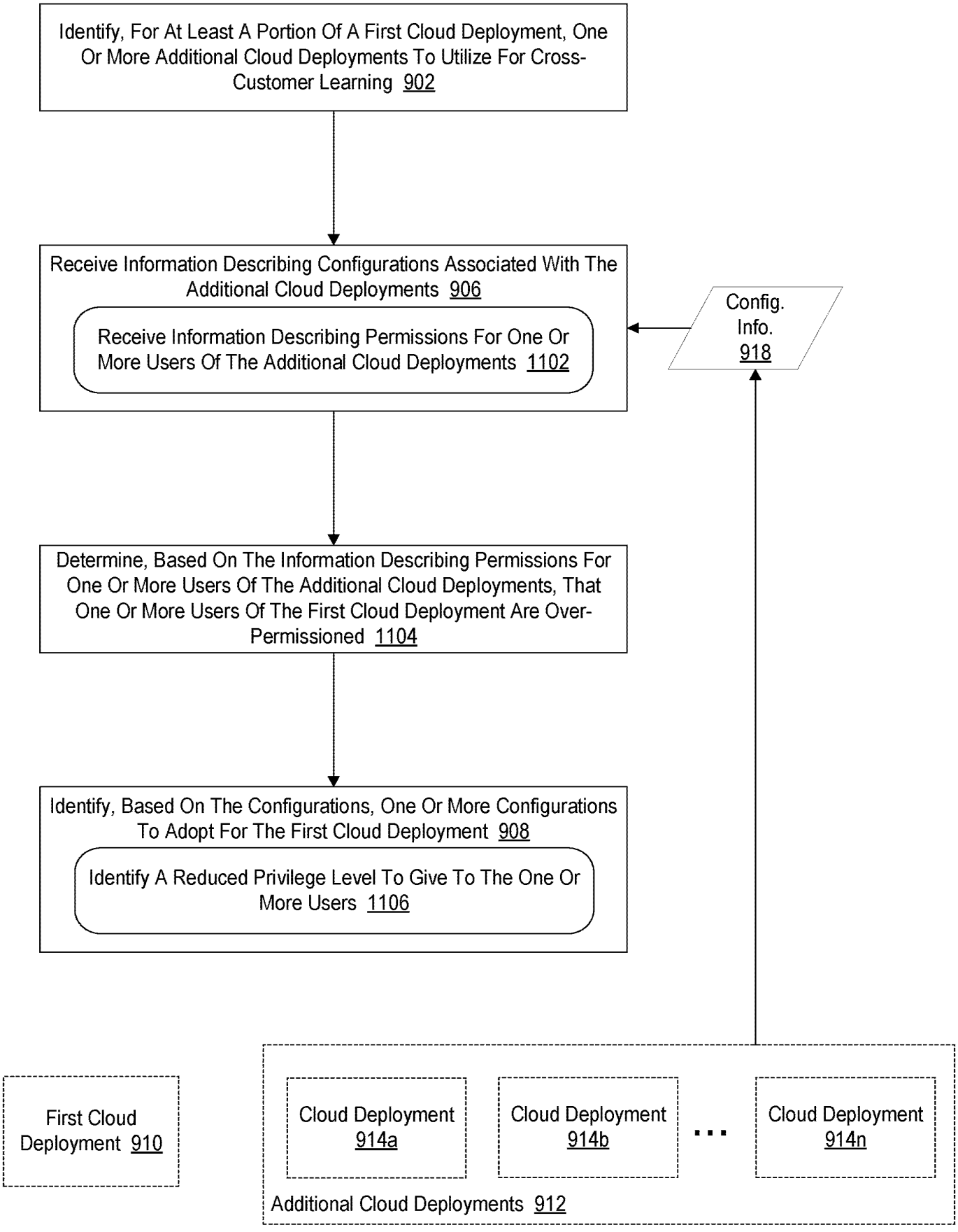

Identify, For At Least A Portion Of A First Cloud Deployment, One Or More Additional Cloud Deployments To Utilize For Cross-Customer Learning 902
Receive Information Describing Configurations Associated With The Additional Cloud Deployments 906
Receive Information Describing Permissions For One Or More Users Of The Additional Cloud Deployments 1102
Config. Info. 918
Determine, Based On The Information Describing Permissions For One Or More Users Of The Additional Cloud Deployments, That One Or More Users Of The First Cloud Deployment Are Over-Permissioned 1104
Identify, Based On The Configurations, One Or More Configurations To Adopt For The First Cloud Deployment 908
Identify A Reduced Privilege Level To Give To The One Or More Users 1106
First Cloud Deployment 910
Cloud Deployment 914a
Cloud Deployment 914b
...
Cloud Deployment 914n
Additional Cloud Deployments 912

Fig. 11

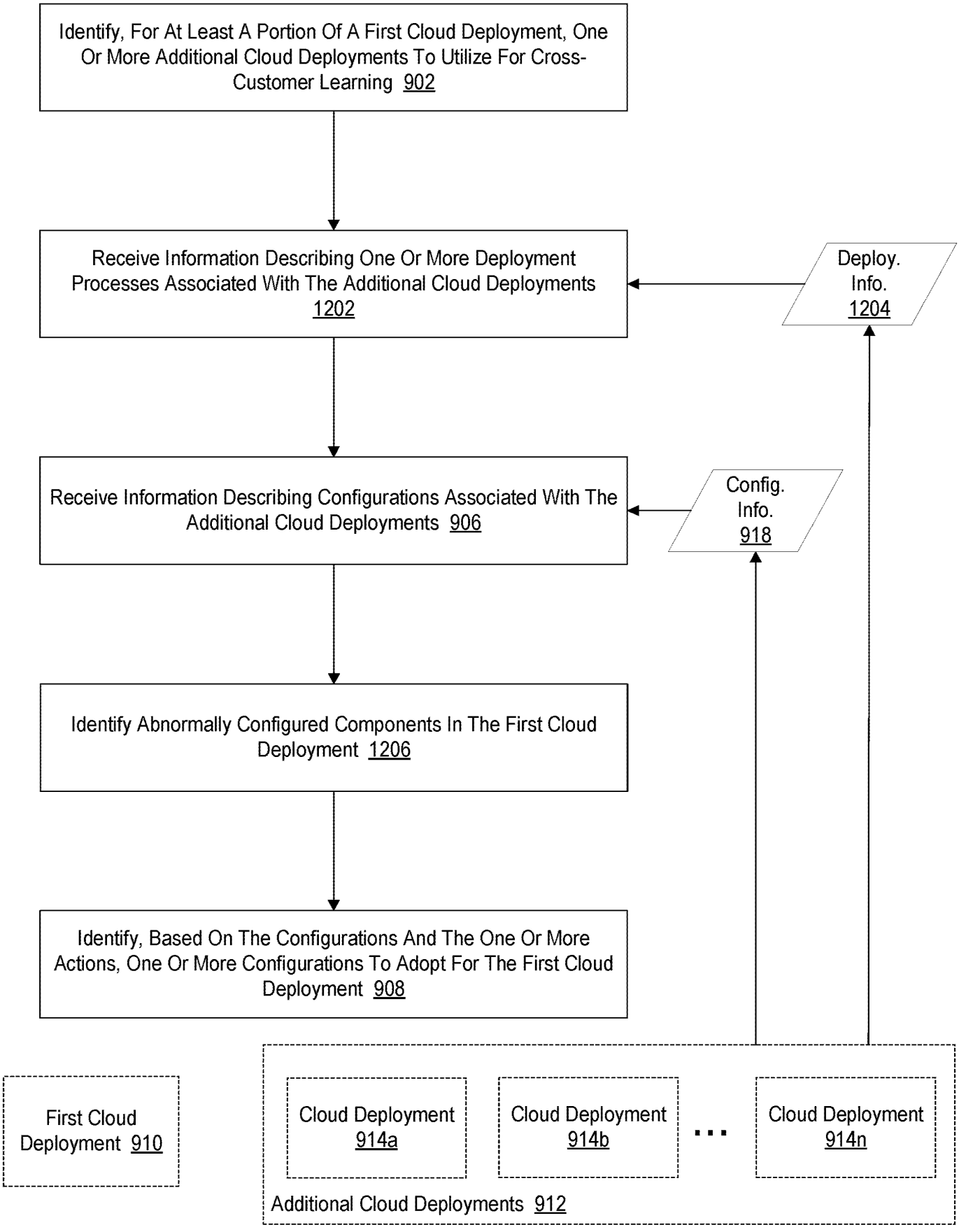

Identify, For At Least A Portion Of A First Cloud Deployment, One Or More Additional Cloud Deployments To Utilize For Cross-Customer Learning 902
Receive Information Describing One Or More Deployment Processes Associated With The Additional Cloud Deployments 1202
Deploy. Info. 1204
Receive Information Describing Configurations Associated With The Additional Cloud Deployments 906
Config. Info. 918
Identify Abnormally Configured Components In The First Cloud Deployment 1206
Identify, Based On The Configurations And The One Or More Actions, One Or More Configurations To Adopt For The First Cloud Deployment 908
First Cloud Deployment 910
Cloud Deployment 914a
Cloud Deployment 914b
...
Cloud Deployment 914n
Additional Cloud Deployments 912

Fig. 12

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS
First Named Inventor/Applicant Name:	ÚLFAR ERLINGSSON
Filer:	Edward Joseph Lenart/Kate Hill
Attorney Docket Number:	LACEP073

Filed as Large Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY APPLICATION FILING	1011	1	320	320
UTILITY SEARCH FEE	1111	1	700	700
UTILITY EXAMINATION FEE	1311	1	800	800
Pages:				
UTILITY APPL SIZE FEE PER 50 SHEETS >100	1081	1	420	420
Claims:				
Miscellaneous-Filing:				
Petition:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**2240**

Electronic Acknowledgement Receipt

EFS ID:	44992160
Application Number:	17671199
International Application Number:	
Confirmation Number:	1070
Title of Invention:	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS
First Named Inventor/Applicant Name:	ÚLFAR ERLINGSSON
Customer Number:	182115
Filer:	Edward Joseph Lenart/Kate Hill
Filer Authorized By:	Edward Joseph Lenart
Attorney Docket Number:	LACEP073
Receipt Date:	14-FEB-2022
Filing Date:	
Time Stamp:	17:02:39
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$2240
RAM confirmation Number	E20222DH03302101
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Application Data Sheet	2022-02-14_LACEP073_ADS.pdf	1307794 675fad5e0b27ce3eb19697591d1c00a8853a98d9	no	8
Warnings:					
Information:					
2	Power of Attorney	2022-02-14_LACEP073_POA.pdf	305399 7f2e712e3f3d3d809e00ccbd05035ca41b5e024a	no	3
Warnings:					
Information:					
3		2022-02-14_LACEP073_Application.pdf	699019 98938a9929e297b2f00a28cff7df982fb7c0be14	yes	122

Multipart Description/PDF files in .zip description

Document Description	Start	End
Specification	1	116
Claims	117	121
Abstract	122	122

Warnings:					
Information:					
4	Drawings-only black and white line drawings	2022-02-14_LACEP073_Figures.pdf	2100595 a4da3a3523bafb5df7d72a0c14f205bef5f394f0	no	49
Warnings:					
Information:					
5	Fee Worksheet (SB06)	fee-info.pdf	46288 057279eec12a1e1c468e53b94a2e58953be1ddfe	no	2

Warnings:	
Information:	
Total Files Size (in bytes):	4459095

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	ÚLFAR		ERLINGSSON	

Residence Information (Select One) ● US Residency Non US Residency Active US Military Service

City	PALO ALTO	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	6201 AMERICA CENTER DRIVE
Address 2	SUITE 200

City	SAN JOSE	State/Province	CA
Postal Code	95002	Country	US

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	YIJOU		CHEN	

Residence Information (Select One) ◉ US Residency Non US Residency Active US Military Service

City	CUPERTINO	State/Province	CA	Country of Residence	US

Mailing Address of Inventor:

Address 1	6201 AMERICA CENTER DRIVE
Address 2	SUITE 200

City	SAN JOSE	State/Province	CA
Postal Code	95002	Country	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	182115		
Email Address		Add Email	Remove Email

Application Information:

Title of the Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS		
Attorney Docket Number	LACEP073	**Small Entity Status Claimed**	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	49	**Suggested Figure for Publication (if any)**	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	● Customer Number	US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	182115		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation in part of	17504311	2021-10-18
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Claims benefit of provisional	63243013	2021-09-10

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1 — Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

● Assignee	Legal Representative under 35 U.S.C. 117	Joint Inventor
Person to whom the inventor is obligated to assign.	Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here. ☒

Organization Name	LACEWORK, INC.		
Mailing Address Information For Applicant:			
Address 1	6201 AMERICA CENTER DRIVE		
Address 2	SUITE 200		
City	SAN JOSE	**State/Province**	CA
Country	US	Postal Code	95002
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Remove

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

Signature:

Remove

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Edward J. Lenart/			Date (YYYY-MM-DD)	2022-02-14
First Name	Edward J.	Last Name	Lenart	Registration Number	63665

Additional Signature may be generated within this form by selecting the Add button. Add

PTO/AIA/14 (02-18)
Approved for use through 11/30/2020. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	LACEP073
	Application Number	
Title of Invention	LEARNING FROM SIMILAR CLOUD DEPLOYMENTS	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PATENT ASSIGNMENT COVER SHEET

Electronic Version v1.1
Stylesheet Version v1.2

EPAS ID: PAT7175011

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	ASSIGNMENT

CONVEYING PARTY DATA

Name	Execution Date
ÚLFAR ERLINGSSON	02/11/2022
YIJOU CHEN	02/11/2022

RECEIVING PARTY DATA

Name:	LACEWORK, INC.
Street Address:	6201 AMERICA CENTER DRIVE
Internal Address:	SUITE 200
City:	SAN JOSE
State/Country:	CALIFORNIA
Postal Code:	95002

PROPERTY NUMBERS Total: 1

Property Type	Number
Application Number:	17671199

CORRESPONDENCE DATA

Fax Number: (512)472-9887

Correspondence will be sent to the e-mail address first; if that is unsuccessful, it will be sent using a fax number, if provided; if that is unsuccessful, it will be sent via US Mail.

Phone: 512-472-9881
Email: kate@klspatents.com, kennedm5842@intake.appcoll.com
Correspondent Name: LACEWORK, INC. C/O KLS
Address Line 1: 797 SAM BASS ROAD #2559
Address Line 4: ROUND ROCK, TEXAS 78681

ATTORNEY DOCKET NUMBER:	LACEP073
NAME OF SUBMITTER:	KATE HILL
SIGNATURE:	/KATE HILL/
DATE SIGNED:	02/14/2022
	This document serves as an Oath/Declaration (37 CFR 1.63).

Total Attachments: 2
source=2022-02-14_LACEP073_Assignment#page1.tif
source=2022-02-14_LACEP073_Assignment#page2.tif

DOCKET NUMBER: **LACEP073**

ASSIGNMENT-STATEMENT

Title of Invention: **LEARNING FROM SIMILAR CLOUD DEPLOYMENTS**

As a below named inventor, I hereby declare that:

This declaration is directed to the attached application, or (if following box is checked):

[ ] United States application or PCT international application number _____ filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I have reviewed and understand the contents of the application, including the claims.

I am aware of the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in 37 CFR Section 1.56.

Whereas, I ("ASSIGNOR") have made certain inventions, improvements, and discoveries (herein referred to as the "Invention") disclosed in the above-identified patent application and further identified by the Docket Number provided above in the header of this document;

Whereas, Lacework, Inc., (herein referred to as the "ASSIGNEE"), a corporation organized and existing under the laws of the State of Delaware, with a principal office at 6201 America Center Drive, Suite 200, San Jose, CA 95002 (herein referred to as "Lacework"), desires to acquire, and each undersigned inventor desires to grant to Lacework, the entire worldwide right, title, and interest in and to the Invention and in and to any and all patent applications and patents directed thereto;

Now, therefore, for good and valuable consideration, the receipt and sufficiency thereof being hereby acknowledged, I hereby sell or have sold, assign or have assigned, and otherwise transfer or have transferred to the ASSIGNEE, its successors, legal representatives, and assigns, my entire worldwide right, title, and interest in and to the Invention, the above-identified United States patent application, and any and all other patent applications and patents for the Invention which may be applied for or granted therefor in the United States and in all foreign countries and jurisdictions, including all divisions, continuations, reissues, reexaminations, renewals, extensions, counterparts, substitutes, and extensions thereof, and all rights of priority resulting from the filing of such applications and granting of such patents. In addition, I hereby authorize and request the Director of the United States Patent and Trademark Office to issue any United States Patent, and foreign patent authorities to issue any foreign patent, granted for the Invention, to the ASSIGNEE, its successors, legal representatives, and assigns, my entire worldwide right, title, and interest in and to the same to be held and enjoyed by the ASSIGNEE, its successors, legal representatives, and assigns to the full end of the terms for which any and all such patents may be granted, as fully and entirely as would have been held and enjoyed by me had this Assignment not been made; and I agree to execute any and all documents and instruments and perform all lawful acts reasonably related to recording this Assignment or perfecting title to the Invention and all related patents and applications, in the ASSIGNEE, its successors, legal representatives, and assigns, whenever requested by the ASSIGNEE, its successors, legal representatives, or assigns.

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00779

I acknowledge my prior and ongoing obligations to sell, assign, and transfer my rights under this Assignment to the ASSIGNEE and am unaware of any reason why I may not have the full and unencumbered right to sell, assign, and transfer my rights hereby sold, assigned, and transferred, and have not executed, and will not execute, any document or instrument in conflict herewith. I also hereby grant the ASSIGNEE, its successors, legal representatives, and assigns, the right to insert in this Assignment any further identification (including, but not limited to, patent Application Number) which may be necessary or desirable for recordation of this Assignment. This Assignment is governed by the substantive laws of the State of Delaware, and any disputes will be resolved in a Delaware state court or federal court sited in Delaware.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. § 1001 by fine or imprisonment of not more than five (5) years, or both.

Signature: / Úlfar Erlingsson / Date: 02/11/2022

(1) Legal Name of Inventor: **ÚLFAR ERLINGSSON**

Signature: / Yijou Chen / Date: 02/11/2022

(2) Legal Name of Inventor: **YIJOU CHEN**

Netskope Ex. 1005
Netskope v. Fortinet IPR2026-00515
Page 00780