

Filed on behalf of: Wiz, Inc.

By: Matthew A. Argenti (margenti@wsgr.com)
Michael T. Rosato (mrosato@wsgr.com)
Wesley E. Derryberry (wderryberry@wsgr.com)
Tasha M. Thomas (tthomas@wsgr.com)
Joseph M. Baillargeon (jbaillargeon@wsgr.com)
WILSON SONSINI GOODRICH & ROSATI
650 Page Mill Road
Palo Alto, CA 94304

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

WIZ, INC.,
Petitioner,

v.

ORCA SECURITY LTD.,
Patent Owner.

Case IPR2025-00093
Patent No. 11,582,257

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 11,582,257**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES UNDER 37 C.F.R. §42.8.....	2
III.	CERTIFICATIONS.....	3
IV.	IDENTIFICATION OF CHALLENGE; STATEMENT OF PRECISE RELIEF REQUESTED	3
V.	THE '257 PATENT.....	3
A.	Prosecution History	5
VI.	NO BASIS EXISTS FOR DENIAL UNDER 35 U.S.C. §325(D)	6
VII.	LEVEL OF ORDINARY SKILL	7
VIII.	CLAIM CONSTRUCTION	7
IX.	PRIOR ART.....	8
A.	Keren (U.S. Patent No. 11,374,982, EX1005).....	8
B.	Engelberg (U.S. Publication No. 2022/0263855, EX1006)	12
X.	GROUND 1: CLAIMS 1-20 WERE OBVIOUS OVER KEREN.....	13
A.	Independent Claims 1, 11, and 20.....	14
1.	Preambles	14
2.	Element 1.i	19
3.	Elements 1.1, 11.1, and 20.1.....	19
4.	Elements 1.2, 11.2, and 20.2.....	27
5.	Elements 1.3, 11.3, and 20.3.....	28
6.	Elements 1.4, 11.4, and 20.4.....	31
7.	Elements 1.5, 11.5, and 20.5.....	38
8.	Elements 1.6, 11.6, and 20.6.....	43
9.	Elements 1.7, 11.7, and 20.7.....	47
B.	Dependent Claims	53
1.	Claims 2-3 and 12-13.....	53

2.	Claims 4 and 14.....	54
3.	Claims 5 and 15.....	55
4.	Claims 6 and 16.....	56
5.	Claims 7 and 17.....	57
6.	Claims 8 and 18.....	59
7.	Claims 9 and 19.....	60
8.	Claim 10.....	60
XI.	GROUND 2: CLAIMS 1-20 WERE OBVIOUS OVER KEREN AND ENGELBERG.....	64
A.	Reasons to Combine Keren and Engelberg.....	64
B.	Claims 1, 11, and 20.....	68
C.	Claim 10	70
XII.	CONCLUSION.....	71

LISTING OF CHALLENGED CLAIMS

1. A cloud-based cybersecurity system for assessing internet exposure of a cloud-based workload, the system comprising:

[1.i] at least one hardware processor configured to:

[1.1] access at least one cloud provider API to determine a plurality of entities capable of routing traffic in a virtual cloud environment associated with a target account containing the workload;

[1.2] query the at least one cloud provider API to determine at least one networking configuration of the entities;

[1.3] build a graph connecting the plurality of entities based on the networking configuration;

[1.4] access a data structure identifying services publicly accessible via the Internet and capable of serving as an internet proxy;

[1.5] integrate the identified services into the graph;

[1.6] traverse the graph to identify at least one source originating via the Internet and reaching the workload; and

[1.7] output a risk notification associated with the workload.

2. The system of claim 1, wherein the plurality of entities includes a virtual network appliance.

3. The system of claim 2, wherein the virtual network appliance is at least one of a load balancer, a firewall, a proxy, or a router.

4. The system of claim 1, wherein the networking configuration is at least one of a routing configuration, a proxy configuration, a load balancing configuration, a firewall configuration, or a VPN configuration.

5. The system of claim 1, wherein the graph includes a data structure sequentially connecting entities.

6. The system of claim 1, wherein the graph includes directional vectors indicating directions of dataflow.

7. The system of claim 1, to build the graph, the at least one processor is configured to: identify individual entities as nodes; and connect the nodes.
8. The system of claim 1, wherein the graph includes port numbers associated with the workload.
9. The system of claim 1, wherein the graph includes a path from the at least one source to the workload.
10. The system of claim 1, wherein the risk notification includes one or more resolution recommendations.
11. A method for assessing internet exposure of a cloud-based workload, comprising:
 - [11.1] accessing at least one cloud provider API to determine a plurality of entities capable of routing traffic in a virtual cloud environment associated with a target account containing the workload;
 - [11.2] querying the at least one cloud provider API to determine at least one networking configuration of the entities;
 - [11.3] building a graph connecting the plurality of entities based on the networking configuration;
 - [11.4] accessing a data structure identifying services publicly accessible via the internet and capable of serving as an internet proxy;
 - [11.5] integrating the identified services into the graph;
 - [11.6] traversing the graph to identify at least one source originating via the internet and reaching the workload; and
 - [11.7] outputting a risk notification associated with the workload.
12. The method of claim 11, wherein the plurality of entities includes a virtual network appliance.
13. The method of claim 12, wherein the virtual network appliance is at least one of a load balancer, a firewall, a proxy, or a router.

14. The method of claim 11, wherein the networking configuration is at least one of a routing configuration, a proxy configuration, a load balancing configuration, a firewall configuration, or a VPN configuration.

15. The method of claim 11, wherein the graph includes a data structure sequentially connecting entities.

16. The method of claim 11, wherein the graph includes directional vectors indicating directions of dataflow.

17. The method of claim 11, wherein building the graph comprises: identifying individual entities as nodes; and connecting the nodes.

18. The method of claim 11, wherein the graph includes port numbers associated with the workload.

19. The method of claim 11, wherein the graph includes port numbers associated with the workload.

20. A non-transitory computer-readable medium storing instructions that, when executed by at least one processor, are configured to cause the at least one processor to perform operations comprising:

[20.1] accessing at least one cloud provider API to determine a plurality of entities capable of routing traffic in a virtual cloud environment associated with a target account containing a workload;

[20.2] querying the at least one cloud provider API to determine at least one networking configuration of the entities;

[20.3] building a graph connecting the plurality of entities based on the networking configuration;

[20.4] accessing a data structure identifying services publicly accessible via the internet and capable of serving as an internet proxy;

[20.5] integrating the identified services into the graph;

[20.6] traversing the graph to identify at least one vector originating via the internet and reaching the workload; and

[20.7] outputting a risk notification associated with the workload.

I. INTRODUCTION

Petitioner Wiz, Inc. (“Wiz”) respectfully requests review of U.S. Patent No. 11,582,257 (“the ’257 patent”), currently assigned to Orca Security Ltd. (“Orca”). This petition demonstrates claims 1-20 are unpatentable.

The ’257 claims describe techniques for identifying internet exposure of a cloud-based workload. These techniques involve using a cloud provider API to determine entities that are capable of routing traffic in a virtual cloud environment and determining their networking configurations. Next, a graph is built that connects these entities based on the determined networking configurations. Services that are publicly accessible via the internet and capable of serving as an internet proxy are then identified and integrated into the graph. The graph may then be traversed to identify sources or vectors that originate via the internet and reach the workload. Finally, a risk notification associated with the workload is outputted.

These techniques of generating and traversing a network graph to determine a risk associated with a workload were already known prior to the ’257 patent’s priority date. In fact, a network graphing approach nearly identical to the one described in the ’257 patent had already been developed and claimed in a prior art Wiz patent (EX1005, “Keren”). As demonstrated below, Keren discloses, or would have made obvious to a POSA, all aspects of the challenged claims. The

claims are further obvious in view of Engelberg (EX1006) and its teachings regarding providing risk notifications with remediation recommendations.

Accordingly, Wiz respectfully requests institution.

II. MANDATORY NOTICES UNDER 37 C.F.R. §42.8

Real Party-in-Interest (37 C.F.R. §42.8(b)(1)): Petitioner Wiz is the real party-in-interest.

Related Matters (37 C.F.R. § 42.8(b)(2)): Wiz is involved in litigation involving Orca patents from a different patent family than that of the '257 patent in *Orca Security Ltd. v. Wiz, Inc.*, No. 1-23-cv-00758 (DDE), filed and served on July 12, 2023. Wiz has also filed the following IPR petitions challenging Orca patents unrelated to the '257 patent: IPR2024-00220, IPR2024-00863, IPR2024-00864, IPR2024-00865, IPR2024-01109, IPR2024-01190 and IPR2024-01191.

Lead and Back-Up Counsel (37 C.F.R. §42.8(b)(3)):

Lead Counsel: Matthew A. Argenti (Reg. No. 61,836)

Back-Up Counsel: Michael T. Rosato (Reg. No. 52,182); Wesley E. Derryberry (Reg. No. 71,594); Tasha M. Thomas (Reg. No. 73,207); Joseph M. Baillargeon (Reg. No. 79,685).

Service Information—37 C.F.R. §42.8(b)(4): Wiz consents to electronic service. Please direct all correspondence to lead and back-up counsel at the contact information below. A power of attorney accompanies this petition.

E-mail: margenti@wsgr.com; mrosato@wsgr.com; wderryberry@wsgr.com;
tthomas@wsgr.com; jbaillargeon@wsgr.com

Post: WILSON SONSINI GOODRICH & ROSATI, 650 Page Mill Road,
Palo Alto, CA 94304

Tel.: 650-354-4154 Fax: 650-493-6811

III. CERTIFICATIONS

The '257 patent is available for IPR, and Wiz is not barred or estopped from requesting IPR on these grounds.

IV. IDENTIFICATION OF CHALLENGE; STATEMENT OF PRECISE RELIEF REQUESTED

Wiz seeks cancellation of the challenged claims for the reasons stated below, which are supported with exhibits, including the Declaration of Dr. Angelos Stavrou (EX1002). The claims are unpatentable under 35 U.S.C. §311 and AIA §6 based on at least the following grounds:

Ground	Claims	Basis
1	1-20	§103(a): obviousness over Keren.
2	1-20	§103(a): obviousness over Keren and Engelberg.

V. THE '257 PATENT

The '257 patent issued from U.S. Application No. 17/716,491 ("the '491 Application"), filed April 8, 2022. EX1001, Face. The '491 application claims

priority to Provisional Application No. 63/180,048, filed April 26, 2021. *Id.* The '257 patent thus has an effective filing date no earlier than April 26, 2021, and is subject to AIA §102 and §103. *Id.*; EX1002, ¶20.

The '257 patent describes techniques for securing a cloud infrastructure, including techniques for “assessing internet exposure of a cloud-based workload.” EX1001, Abstract, 1:15-18, 33:46-61, Fig. 6; EX1002, ¶23. The specification describes accessing and querying a cloud provider API to determine entities capable of routing traffic in a virtual cloud environment and their associated networking configuration. EX1001, 33:49-55, 35:19-22; *see also id.*, 33:62-34:6, 34:18-25, 37:53-65, Fig. 1; EX1002, ¶24. The specification further describes building a graph with connections between the determined entities based on the determined networking configuration. EX1001, 36:14-19; *see also id.*, 36:22-29, 36:44-48, 37:66-38:4; EX1002, ¶25. The system may also access a data structure that identifies services publicly accessible via the internet and capable of serving as an internet proxy and integrate these identified services into the graph. EX1001, 36:60-37:4; *see also id.*, 37:8-11, 38:5-14; EX1002, ¶25. Finally, the cybersecurity system may traverse the graph to identify sources originating via the internet that reach the workload and also output a risk notification associated with the workload. EX1001, 37:12-15, 37:32-34; *see also id.*, 38:15-25; EX1002, ¶26.

The '257 patent includes 20 claims. Claims 1, 11, and 20 are independent. Claims 11 and 20 essentially mirror claim 1, but whereas claim 1 is written as a system claim, independent claim 11 is directed to a method, and independent claim 20 is directed to a computer-readable medium. The dependent claims add other conventional aspects of cybersecurity and cloud computing. EX1002, ¶¶27-28.

A. Prosecution History

During prosecution, the '257 patent never received a rejection under §102 or §103. The first office action rejected some of the pending claims under §101 as being “drawn to a computer program per se,” but allowed the remaining claims. EX1004, 406-07. To secure allowance, the rejected claims were amended to recite that the system includes “at least one hardware processor configured to” perform the claimed steps. *Id.*, 216, 222. In allowing the claims, the Examiner identified four references as the closest prior art but found that “none of the prior art of record alone or in combination teaches *accessing a data structure identifying services publicly accessible via the internet and capable of serving as an internet proxy; integrating the identified services into the graph.*” *Id.*, 407-08; EX1002, ¶29. As explained below, these limitations are found in Keren, which was not before the Examiner during prosecution.

VI. NO BASIS EXISTS FOR DENIAL UNDER 35 U.S.C. §325(D)

Under the two-part *Advanced Bionics* framework, §325(d) analysis considers several factors to determine:

(1) whether the same or substantially the same art previously was presented to the Office or whether the same or substantially the same arguments previously were presented to the Office; and (2) if either condition of [the] first part of the framework is satisfied, whether the petitioner has demonstrated that the Office erred in a manner material to the patentability of challenged claims.

Advanced Bionics, LLC v. Med-El Elektromedizinische Geräte GmbH, IPR2019-01469, Paper 6 at 8 (precedential); 35 U.S.C. §325(d).

Keren was not presented to the Office nor considered by the Office during prosecution. See EX1001, (56); see generally EX1004. The reference is also not cumulative of references considered during prosecution. Keren was also never considered in combination with Engelberg since that reference likewise was not disclosed. The Office thus did not consider any of the grounds presented herein. The Office also lacked additional evidence discussed herein, including the declaration provided by Wiz's expert, Dr. Stavrou.

Allowance of the claims also constituted material error under part two of the *Advanced Bionics* test. The Examiner allowed the claims of the '257 patent because the prior art did not disclose accessing a data structure identifying services

publicly accessible via the internet and capable of serving as an internet proxy and integrating the identified services into the graph. *See* EX1004, 407-08; *supra*, §V.A. However, as discussed in more detail below, Keren teaches or suggests each of these limitations as does the combination of Keren and Engelberg. *Infra*, §§X-XI. The claims therefore should not have issued, and they would not have issued if the Examiner had considered the present grounds.

VII. LEVEL OF ORDINARY SKILL

For purposes of this petition, Wiz assumes a priority date of April 26, 2021. A POSA as of April 2021 would have held at least a bachelor's degree in computer science, computer engineering, electrical engineering, or a related field, and would also have 2-3 years of professional experience working with cyber security analysis and virtualization. Additional experience could compensate for less education and vice versa. Relevant work experience includes, for example, malware analysis, security analysis of cloud computing systems, and security analysis of VMs. EX1002, ¶¶21-22. Dr. Stavrou meets these requirements and is qualified to credibly opine on the state of the art and the POSA's perspective. *Id.*, ¶¶1-19; *see also id.*, ¶¶2-5 (qualifications); EX1003 (Stavrou CV).

VIII. CLAIM CONSTRUCTION

Claim terms are given their ordinary and customary meaning, consistent with the specification, as a POSA understood them. 37 CFR §42.100(b); *Phillips v. AWH*

Corp., 415 F.3d 1303, 1312-13 (Fed. Cir. 2005) (en banc). This petition applies the ordinary and customary meaning of the claim terms. *See also* EX1002, ¶30.

IX. PRIOR ART

A. Keren (U.S. Patent No. 11,374,982, EX1005)

Keren was filed February 18, 2021, and is therefore prior art under 35 U.S.C. §102(a)(2).

Keren relates to cybersecurity methods and systems “for automated generation of network analysis insights” in a cloud environment. EX1005, 1:16-18; *see also id.*, Abstract, 3:6-39, 4:31-55, Figs. 1A-B. The cloud environment includes “multiple security objects,” which are “objects, systems, devices, components, applications, entities, and the like, configured to operate within the cloud platform 104 and provide various functionalities therein.” *Id.*, 4:35-43, 5:20-44. The security objects may include, for example, “network objects,” which are objects “configured to provide one or more traffic-management functionalities,” such as “proxies, firewalls, routing tables, and the like.” *Id.*, 5:32-39. The security objects may also be “computing objects,” which “provide one or more data management or processing functionalities,” such as “dedicated processing systems, file servers, databases, and the like.” *Id.*, 5:32-44. Additional security objects identified by Keren include load balancers, virtual machines, and containers. *Id.*, 6:13-19, 9:6-15; EX1002, ¶31.

Keren teaches that the cloud environment connects with a cybersecurity system 150 and a graph database 160. EX1005, 4:43-47, Figs. 1A-B. “The cybersecurity system 150 is a system, device, or component, configured to provide one or more network analysis functionalities including, without limitation, network analysis, traffic analysis, object querying, graph generation, and the like, as well as any combination thereof.” *Id.*, 6:47-51. The graph database 160 “is a storage or memory component, device, system, or the like, configured to provide one or more functionalities relevant to storage of graph-related data.” *Id.*, 7:3-6; *see also id.*, 7:6-11 (graph database stores “graph-related data features,” such as “raw data, graphs, graph edges, graph vertices, graph schemas, and the like”); EX1002, ¶32.

To perform its network analysis, Keren teaches that the cybersecurity system generates security graphs detailing the network topology of the security objects based in the cloud environment. *See* EX1005, Abstract, 10:27-30, Fig. 2; EX1002, ¶33. Security graphs are constructed by querying an API provided by the cloud environment to identify the security objects contained within the network and collect security object data. EX1005, 9:26-55; *see also id.*, 9:56-66 (“Security object data may include data describing, as examples and without limitation, object types, object names or unique identifiers (IDs), object network addresses, object port configurations, object status, such as online, offline, busy, and available, object input signal configurations, object output signal configurations, object

security or access configurations, object processing rules and the like, as well as any combination thereof.”), 10:11-14 (“collection of security object data at S210 may include collection of the identities of all objects included in a cloud platform by generation and transmission of an API query”), Fig. 2. This collected data allows the system to identify “the various objects included in, and adjacent to,” the environment and determine the various network connections, or relationships, between the security objects. *Id.*, 10:27-30, 10:49-57. By identifying the security objects and their relationships, the system constructs a security graph of the network, which can be in various formats, such as a table, chart, a list of objects, or as a “node and link” graph, in which objects are represented as nodes connected by links that represent the objects’ network relationships with one another. *Id.*, 7:18-24, 10:33-48, Figs. 4A (visual node and link graph), 4B (list of objects illustrating network route); EX1002, ¶33.

Keren further teaches that the network properties of an object, as determined by the collected security object data, may be integrated into the security graph by tagging security objects with data labels, tags, or the like. EX1005, 11:13-27 (data tags may describe “connection source and destination, connection type, connection direction, connection status, connection protocol, and the like”), 13:6-25 (data tags may provide “descriptions of an object’s open ports and counts thereof,

descriptions of an object's network address or addresses, descriptions of protocols relevant to an object"); EX1002, ¶34.

From this information, Keren teaches that additional properties can be determined and populated into the graph. EX1002, ¶35. For example, the system can calculate a "reachability property" of each security object contained in the graph. EX1005, Abstract, 11:61-12:10, Figs. 3A-B. A reachability property "defines if and how an object on the generated security graph can be reached from an external or internal network," such as the internet. *Id.*, 13:26-34; *see also id.*, 15:15-24 ("Reachable properties are object properties describing if, and how, a given path element is reachable through the selected path, and specifically, from an external/internal network."). The system calculates a reachability property for a security object by traversing a path in the security graph to the security object and determining the minimal set of network accessibility properties needed to reach that security object. *Id.*, 13:66-14:5 ("[P]aths can be identified by traversing the security graph. The traversal can start or end at objects that are connected to an external network (the internet)."), 15:25-33, 15:44-58. This allows the system to assess a security object's exposure to network traffic originating from the internet. *See id.*, 19:66-20:12, Fig. 5; EX1002, ¶35.

Keren further teaches that "security insights" may be generated "based on the known and calculated properties" of the security objects in the security graph.

EX1005, 12:11-12. These insights provide natural-language descriptions of aspects of the security graph, including descriptions of risky network configurations. *Id.*, 12:12-14, 12:45-13:5. For example, a security insight may notify a user that ““VM thirty normally connects to load balancer twenty and firewall eight but is currently only connected to load balancer twenty”” or ““objects E, a database, and F, an administrator interface, are currently exposed to external networks.”” *Id.*, 12:57-61, 12:66-13:5; EX1002, ¶36.

B. Engelberg (U.S. Publication No. 2022/0263855, EX1006)

Engelberg is a patent publication that is entitled to claim a right of priority under §119 to U.S. Provisional Application No. 63/150,664 (EX1007), filed on February 18, 2021 (*see* EX1006, (22), (60), ¶1), and is therefore prior art under 35 U.S.C. §102(a)(2). As demonstrated by the parallel citations to EX1007, the relevant subject matter of Engelberg relied upon in this petition is fully supported by equivalent disclosure in the provisional application. *Penumbra, Inc. v. RapidPulse, Inc.*, IPR2021-01466, Paper 34, 29-35 (Mar. 10, 2023) (precedential); *see also* EX1002, ¶37 & n.1.

Engelberg is related to cyber risk mitigation. EX1006, Title, Abstract; EX1007, Title, Abstract; EX1002, ¶37. As recognized by Engelberg, “[c]omputer networks are susceptible to attack by malicious users (e.g., hackers)” and “[t]o defend against such attacks, enterprises use security systems to monitor

occurrences of potentially adverse events occurring within a network, and alert security personnel to such occurrences.” EX1006, ¶¶2-3; *see also id.*, ¶26; EX1007, ¶¶1-2, 24. To aid security personnel in managing network environments, Engelberg teaches “recommending remedial actions for cyber security” as part of identifying risks associated with the network. EX1006, ¶30; *see also id.*, ¶59 (remediation recommendations are conveyed to user); EX1007, ¶¶28, 57; EX1002, ¶37.

X. GROUND 1: CLAIMS 1-20 WERE OBVIOUS OVER KEREN

Keren teaches, suggests, or makes obvious all limitations of claims 1-20 as a whole. EX1002, ¶38. Keren teaches cybersecurity systems and methods for assessing the internet exposure of a cloud-based workload by accessing and querying a cloud provider API to identify security objects, including those capable of routing traffic in a virtual cloud environment, and determining a networking configuration of those objects. *Supra*, §IX.A. Keren also teaches that a security graph is built based on the determined networking configuration. Moreover, Keren explains that properties associated with the security objects, including services provided by those objects, are integrated into the graph. The security graph is then traversed to assess whether and how a workload is reachable from the internet based on the determined network information. *Supra*, §IX.A; *see also infra*, §§X.A-B.

A. Independent Claims 1, 11, and 20

1. Preambles

To the extent the preambles are limiting, Keren teaches them. EX1002, ¶¶39-40. Claim 1 recites a “cloud-based cybersecurity system for assessing internet exposure of a cloud-based workload, the system comprising.” Claim 11 recites a “method for assessing internet exposure of a cloud-based workload, comprising.” The preamble of claim 20 recites a “non-transitory computer-readable medium storing instructions that, when executed by at least one processor, are configured to cause the at least one processor to perform operations comprising.”

Keren describes a “cloud-based cybersecurity system” and corresponding methods. EX1002, ¶41. Specifically, Keren teaches methods and systems for assessing “security objects” in a “cloud environment.” EX1005, Abstract; *see also id.*, 1:15-18 (“The present disclosure relates generally to network administration...”), 3:6-39, 4:31-43 (cloud environment includes cloud computing platforms that include “multiple security objects”), Figs. 1A-B, claims 1, 11. Keren describes using a cybersecurity system to assess the security objects and automatically generate network analysis insights of the cloud environment. *See id.*, 1:15-18, 2:34-55 (“network administrators, and the like, may seek a solution which allows for network analysis and includes granular, comprehensive representation of security object properties and insights”), 4:43-55 (cloud

environment connects with cybersecurity system for network analysis purposes), 21:34-37; EX1002, ¶41. The cybersecurity system may be cloud-based and configured to implement assessment methods that include, for example, generating security graphs, determining relationships between security objects, generating insights describing network configurations or events, and determining reachable properties of security objects. *See* EX1005, 3:54-62, 6:33-37, 6:47-56 (cybersecurity system provides “network analysis functionalities including, without limitation, network analysis, traffic analysis, object querying, graph generation, and the like”), 6:57-7:2 (“cyber-security system 150 may be included within any of the various elements of the network system 102, including the cloud environment 103, the various cloud platforms 104, and subparts thereof, and the network 108”), 9:22-25, 12:11-19, 13:26-34, 14:31-33, Figs. 2-3B, claim 11; EX1002, ¶41.

Keren further teaches that the cloud-based cybersecurity system and corresponding methods are “for assessing internet exposure of a cloud-based workload,” as recited. EX1002, ¶42.

First, the security objects based within the cloud environment include “a cloud-based workload.” EX1002, ¶43. According to the ’257 patent, workloads are broadly described as “systems, devices, or resources in a network (or available via a network) such as a cloud infrastructure.” EX1001, 16:56-59; *see also id.*,

39:44-46 (“A workload, as used herein, may refer to a specific application, service, capability, or a specific amount of work that can be run on a cloud resource, system, or infrastructure.”). Examples of workloads identified by the ’257 patent include virtual machines, databases, storage servers, load balancers, containers, and applications. *Id.*, 16:60-63, 39:44-50, 54:59-55:3, Fig. 1.

Keren teaches that the security objects based within the cloud environment include the same types of workloads described by the ’257 patent. EX1002, ¶44. For example, like the ’257 patent, Keren generally explains that the security objects are “objects, systems, devices, components, applications, entities, and the like, configured to operate within the cloud platform 104” of the cloud environment. EX1005, 5:20-32. These security objects may include “computing objects,” which “are objects configured to provide one or more data management or processing functionalities” such as “dedicated processing systems, file servers, databases, and the like.” *Id.*, 5:32-44, Fig. 5. Keren explains that the security objects may also include virtual machines, containers, and load balancers. *Id.*, 6:10-19, 7:46-56, 9:9-15, 10:20-26, 14:59-61, 20:34-42, Figs. 4A-B; EX1002, ¶44.

Keren’s system and method also assess the “internet exposure” of a workload. EX1002, ¶45. As discussed in more detail below (*infra*, §§X.A.5, X.A.8), the cybersecurity system is configured to construct “[a] security graph describing the various objects included in, and adjacent to, a network, as well as

the relationships between such objects.” EX1005, 10:27-30; *see also id.*, 11:11-23 (stored security object relationships include “connection source and destination, connection type, connection direction, connection status, connection protocol, and the like”), 13:35-47, 16:54-67 (security graph provides information representing connections between objects). The security graph is then used to determine reachable properties of security objects. *Id.*, 13:26-14:30; EX1002, ¶45. “A reachable property defines if and how an object on the generated security graph can be reached from an external or internal network, and/or an external or internal object.” EX1005, 13:28-31. As Keren explains, “[e]xternal means outside of the cloud environment of an organization,” which includes the internet. *Id.*, 13:28-34, 13:46-14:1 (teaching network paths that “start or end at objects that are connected to an external network (the internet)”). By determining these reachable properties, Keren’s system assesses the ways in which a security object, such as a “workload” described above, may be accessible from the internet (and thus exposed to traffic originating from the internet). *See id.*, 12:62-13:5 (insights may describe whether objects are exposed to external networks), 15:44-58 (reachable properties “describe the properties of an object with respect to the object’s internet accessibility”), 20:34-42 (scope of reachable property may be the internet), 19:34-20:12 (analyzing internet accessibility of a database), Figs. 4B (illustrating network path between

container and internet), 5 (determining network routing between public proxy and database); EX1002, ¶45.

Finally, Keren teaches a “non-transitory computer-readable medium storing instructions that, when executed by at least one processor, are configured to cause the at least one processor to perform” the elements of claim 20. Keren explains that the cybersecurity system used to perform the operations of claim 20 “may be configured to execute one or more instructions, methods, processes, and the like.” EX1005, 6:51-56, Figs. 2-3B; *infra*, §§X.A.3-9. The cybersecurity system, in particular, includes a “processing circuitry 610 coupled to ... a storage 430.” EX1005, 20:61-67; *see also id.*, 4:3-4 (“Fig. 6 is a hardware block diagram depicting a cyber-security system, according to an embodiment.”), Fig. 6 (illustrating processing circuitry 610). Keren teaches that the processing circuitry “may be realized as one or more hardware logic components and circuits,” such as a processor. *Id.*, 21:1-12 (hardware logic components can be “general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), and the like”). Moreover, the storage may be a non-transitory computer-readable medium (e.g., “magnetic storage,” “optical storage,” or “any other medium which can be used to store the desired information”), which stores software instructions that “when executed by the processing circuitry 610, cause the processing circuitry 610 to perform the various processes” taught by Keren. *Id.*, 21:16-33; EX1002, ¶46. In

the context of this ground, a POSA therefore would have understood that execution of the instructions contained on the medium would “cause the at least one processor to perform” each of the operations recited in claim 20. EX1002, ¶¶46-47.

2. Element 1.i

Keren teaches this element, which recites “at least one hardware processor configured to....” As discussed above, regarding claim 20, Keren teaches that the cybersecurity system includes a hardware processor used to execute instructions to perform the described methods. EX1005, 20:61-67, 21:1-12, 21:16-33; EX1002, ¶¶48-49.

3. Elements 1.1, 11.1, and 20.1

Keren teaches these elements, which recite “[access/accessing] at least one cloud provider API to determine a plurality of entities capable of routing traffic in a virtual cloud environment associated with a target account containing [the/a] workload.” EX1002, ¶50.

Keren describes “a plurality of entities capable of routing traffic in a virtual cloud environment,” as recited. EX1002, ¶51. As described above, Keren teaches a cybersecurity system that provides network analysis of security objects within a cloud computing environment, which may be a virtual cloud environment. EX1005, 5:4-13 (cloud platforms may be “a virtual network” and may “simulate or

emulate” other networks such as “the Internet”), 8:29-53; *supra*, §X.A.1. Keren explains that the security objects may also include the same types of entities that are capable of routing traffic in a virtual cloud environment as identified in the ’257 patent. EX1002, ¶51. For example, the ’257 patent explains that “devices and systems capable of routing and filtering traffic” include load balancers, routers, switches, firewalls, security groups, API Gateways, and proxies. EX1001, 34:18-21; *see also id.*, 34:36-51 (entities capable of routing traffic include virtual network appliances such as “a firewall, WAN optimizer, application delivery controllers, routers, load balancers, IDS/IPS, proxies”), 34:51-53, claims 2-3, 12-13.

Like the ’257 patent, Keren teaches the security objects may include “network objects,” which “are objects configured to provide one or more traffic-management functionalities,” which may be “proxies, firewalls, routing tables, and the like.” EX1005, 5:32-39. Other security objects providing traffic-management functionalities described by Keren include gateways, load balancers, and security groups. *Id.*, 5:66-6:4 (“[S]ecurity objects may include virtual entities, devices, and the like, to process layer-7 (application layer) traffic, such as ... Amazon Load Balancer® (ALB) layer seven services and applications.”), 6:14-19 (security objects include “firewalls, network interface cards, proxies, gateways ... hubs, [and] virtual private networks (VPNs)”), 9:9-15 (additional security object

examples include “peering connections, load balancers, [and] route tables”), 10:14-20, Figs. 4A-B, 5 (illustrating security group 530); EX1002, ¶51.

Further, Keren teaches “accessing at least one cloud provider API” to determine the plurality of entities described above. EX1002, ¶52. Specifically, to “construct[] security graphs, including network and computing objects,” Keren teaches that “security objects are identified, and security object data is collected.” EX1005, 9:21-27, Fig. 2. The “security objects may be identified by querying a cloud platform,” which can “include generation of one or more queries through an application programming interface (API).” *Id.*, 9:27-44 (“Through the API, security objects’ identities and description data are provided in response to such API queries.”); *see also id.*, 9:56-10:10, 10:11-14, 10:14-26; EX1002, ¶52. Keren explains that the API may be “platform-specific APIs,” such as those provided by “Amazon AWS®, Microsoft Azure®, and the like,” which a POSA would have understood to be a cloud provider API. EX1005, 9:47-55; EX1002, ¶52.

Finally, Keren teaches that the plurality of entities in the virtual cloud environment is “associated with a target account containing [the/a] workload” in two ways. EX1002, ¶53. As detailed more below, Keren first teaches that the security objects, which include workloads and a plurality of entities, may be provided on a service basis through a commercially-available cloud provider, which a POSA would have understood would be accessed and managed through a

subscriber account. Second, Keren teaches that individual user accounts may be associated with the cloud environment, which a POSA would have understood allowed individual users access to particular workloads (e.g., a particular virtual machine) located within the cloud environment. *Id.*

First, a POSA would have understood that the security objects taught by Keren would have been associated with a subscriber/owner account (i.e., a target account). *Id.*, ¶54. As illustrated in Fig. 1A (reproduced below), Keren teaches that the cloud environment includes “a number of cloud platforms” that each contain “multiple security objects,” which, as discussed above, include both a plurality of entities (e.g., firewalls, proxies, gateways) and workloads (e.g., databases, containers, virtual machines). EX1005, 4:31-43, 5:20-6:19; *supra*, §X.A.1; EX1002, ¶54.

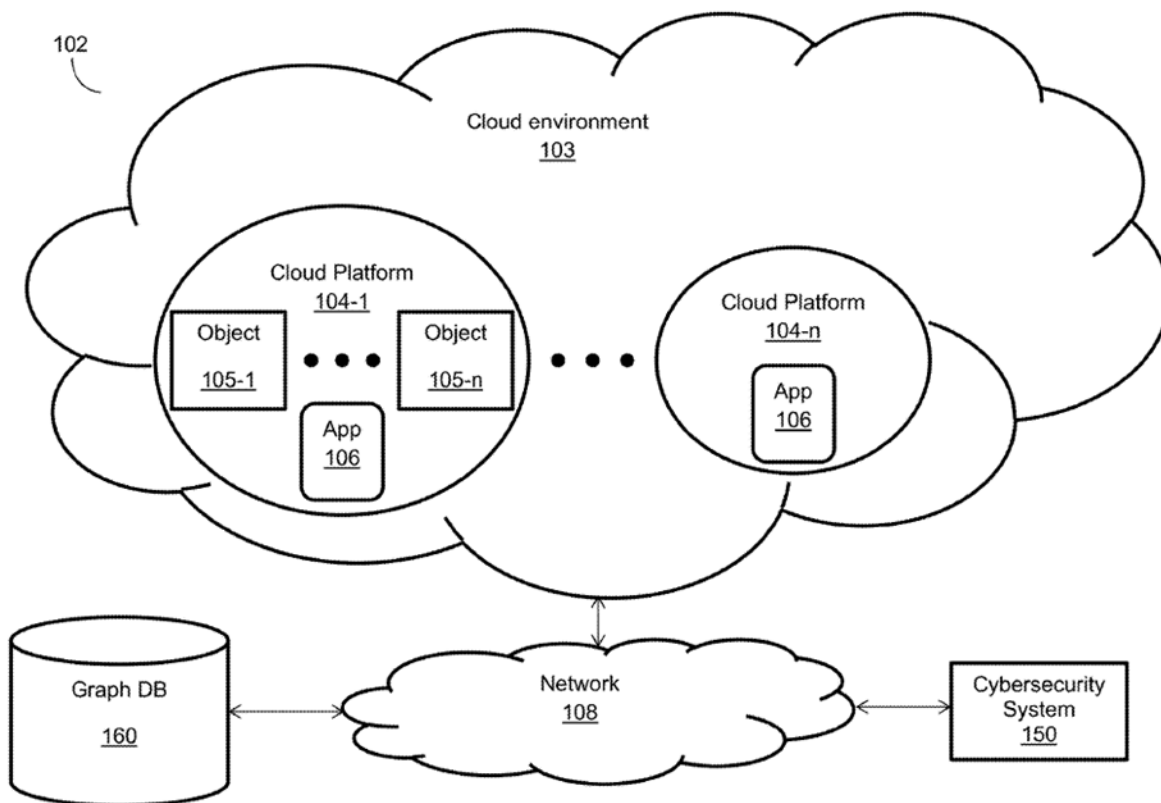


FIG. 1A

EX1005, Fig. 1A.

Keren teaches that a cloud platform “is a platform, architecture, or other, like, configuration providing for connectivity of the various objects” within the platform and “may be a commercially-available cloud system, provided on a service basis, such as ... Amazon AWS®, Microsoft Azure®, and the like.”

EX1005, 4:56-65; *see also id.*, 4:32-35 (cloud environment “represents an organization’s cloud-based resources, and the various connections between such resources”). It was well-known that when an entity, such as a business or organization, sought to use as a cloud platform the services of commercially-

available cloud providers, like Amazon AWS or Microsoft Azure, a subscriber account (or administrator/owner account) was required. EX1002, ¶55 (citing EX1008, vii, 1, 22-23). The subscriber account allowed the account owner to purchase resources, create virtual workloads and entities, and manage their network connections. EX1002, ¶55 (citing EX1008, 2, 5-6, 11, 74-78). Given Keren's teachings that a cloud platform may be provided on a service basis by commercially-available cloud providers, a POSA would have understood that the cloud platform, which contains the security objects (i.e., plurality of entities and workloads), would be associated with a target account containing the workload, as recited. EX1002, ¶55. This understanding is consistent with the '257 patent. EX1001, 7:57-67 (cloud infrastructure may comprise devices and systems part of "Amazon Web Services, Microsoft Azure, Google Cloud Platform," or "any other cloud platform provider"), 16:17-40 (source account may be "a domain, a location on a network server, or anything used to access system resources"), 34:33-37 ("A target account containing the workload, as used herein, may refer to a location on a network server selected as an aim of connection....").

Second, Keren also teaches that the security objects in the cloud environment may be associated with a specific user account containing a workload (another type of target account). EX1002, ¶56. For example, as shown in Fig. 4A below, a user account may be connected to proxy peering, gateway, and firewall

objects, which—as discussed above—are capable of routing traffic in the virtual cloud environment.

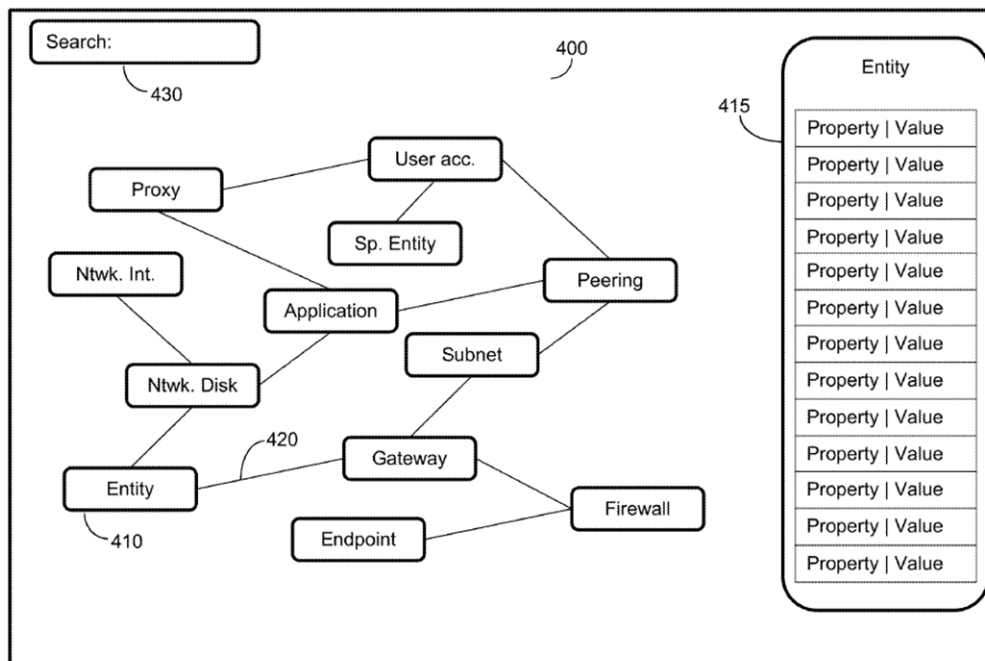


FIG. 4A

EX1005, Fig. 4A.¹

¹ As Dr. Stavrou notes, while only two nodes are labelled “Entity” in Fig. 4A, a POSA would not have understood the claimed entities to be limited to only those nodes. Rather, the entities described and claimed in the ’257 patent encompass many types of nodes depicted in Fig. 4A and described in Keren. EX1002, ¶56 n. 3 (noting Fig. 4A shows entities as described by the ’257 patent, such as firewall,

A POSA would have understood, or at least found it obvious, that a user account, such as that shown in Fig. 4A, represented a point of authorized access for a particular user to use computing resources (e.g., a given virtual machine) maintained in the virtual cloud environment. *See, e.g., id.*, 12:45-55 (detecting unauthorized network connection of user’s device), 15:44-58 (accessibility properties of security objects may be analyzed for “various given accounts”), 20:34-42 (same); EX1002, ¶57 (citing EX1008, 6-7, 9-11, 15-21). A POSA would have understood that this type of individual user account would provide access to particular workloads and be associated with a plurality of entities within the virtual cloud environment and would represent a “target account” consistent with the ’257 patent. EX1001, 34:33-37 (“A target account containing the workload, as used herein, may refer to a location on a network server selected as an aim of

proxy, and gateway) (citing EX1001, 34:18-21 (plurality of entities include load balancers, routers, switches, firewalls, security groups, API Gateways, and proxies), 34:36-53). Moreover, because Keren uses the term “entity” interchangeably with “security object” to refer to particular nodes of interest, even those two “Entity” nodes of Fig. 4A generally refer to the various types of security objects described throughout the reference. *See, e.g.,* EX1005, 2:42-53, 11:61-12:10, 13:62-65; EX1002, ¶56 n. 3.

connection....”); 39:67-40:3 (“A target account of the target system in this disclosure may refer to an account of a user of a cloud infrastructure (e.g., a cloud environment or a cloud storage environment.”); EX1002, ¶¶57-58.

4. Elements 1.2, 11.2, and 20.2

Keren teaches these elements, which recite “[query/querying] the at least one cloud provider API to determine at least one networking configuration of the entities.” EX1002, ¶59.

As discussed above, Keren teaches querying a cloud provider API to determine the security objects, such as the recited plurality of entities, based within the cloud environment. *Supra*, §X.A.3; *see also* EX1005, 9:38-55 (identifying security objects includes querying cloud platforms using “an application programming interface (API)”), Fig. 2 (step S210). Keren teaches that the cloud provider API is also queried to determine additional data relating to the security objects, including information relating to an object’s network configuration, such as an object’s network addresses, port configurations, or access configurations. *Id.*, 9:38-55 (“identification of security objects and collection of security object data ... includes querying each cloud platform” using an API), 9:56-10:10 (security object data includes network configuration information, such as “object network addresses,” “object port configurations,” “object [input/output] signal configurations,” and “object security or access configurations”), Fig. 2 (step S210);

EX1002, ¶60. As one example, Keren describes an API query in which information relating to the networking configuration of a firewall is collected, including “the collection of firewall rules, collection of firewall event logs, collection of firewall port configurations, and the like.” EX1005, 10:14-20; *see also id.*, 19:62-20:5 (describing use of determined public proxy port configuration for network path analysis); EX1001, 35:35-38 (“[T]he networking configuration is at least one of a routing configuration, a proxy configuration, a load balancing configuration, a firewall configuration, or a VPN configuration.”); EX1002, ¶¶60-61.

5. Elements 1.3, 11.3, and 20.3

Keren teaches these elements, which recite “[build/building] a graph connecting the plurality of entities based on the networking configuration.” EX1002, ¶62.

Keren teaches that, based on the collected data described above (*supra*, §§X.A.3-4), the system constructs a security graph (i.e., builds a graph), which is “a data feature describing the various objects included in, and adjacent to, a network, as well as the relationships between such objects.” EX1005, 10:27-33, Fig. 2; EX1002, ¶63. The “security graph may be constructed in one or more various formats,” such as “a table, chart, or other, nonvisual, data organization format, a list of objects,” or as a “visual ‘node and link’ graph.” EX1005, 10:33-

48; *see also id.*, Figs. 4A (illustrating node-and-link graph and reproduced below), 4B (illustrating object-to-object network path graph).

Keren also teaches that building the security graph connects the plurality of entities based on the determined network configuration of the entities. EX1002, ¶64. Specifically, Keren teaches determining “relationships between security objects,” which “are descriptions of the various connections between the [identified] security objects.” EX1005, 10:49-52, Fig. 2 (step S230); EX1002, ¶64. Keren describes using a static analysis process that analyzes the collected security object data, including the data collected on the network configuration of the entities, to determine the relationships between security objects. EX1005, 10:58-64 (“[T]he static analysis may include analysis of object and protocol code and rules, based on simulated network operation, as collected at S210, to provide for identification of security object relationships based on security object configurations.”); *supra*, §X.A.4; EX1002, ¶64. In an example of a static analysis determining connections between objects, Keren describes collecting object data from a firewall that “may specify, in the firewall’s port configurations, communication with a first device on a first port using a first protocol” and “transmission of a specific log file, via a third port, to a connected repository.” EX1005, 10:64-11:5. Based on this networking configuration, Keren teaches that security object relationships are determined to include, for example, “connections

between the firewall and the first device” and “connections between the firewall and the repository.” EX1005, 11:5-10; EX1002, ¶64.

Keren further teaches that the determined relationships are included in the constructed graphs. EX1005, 11:11-13 (“Determination of security object relationships at S230 may further include updating the graph or graphs constructed at S220 to include the determined relationships.”); *see also id.*, 11:14-27; EX1002, ¶65. The determined relationships may be included, for example, as “links” that connect “nodes” representing security objects located within the environment. EX1005, 11:28-37. Shown below is an example of a node-and-link security graph visualization built from the determined relationships between the security objects:

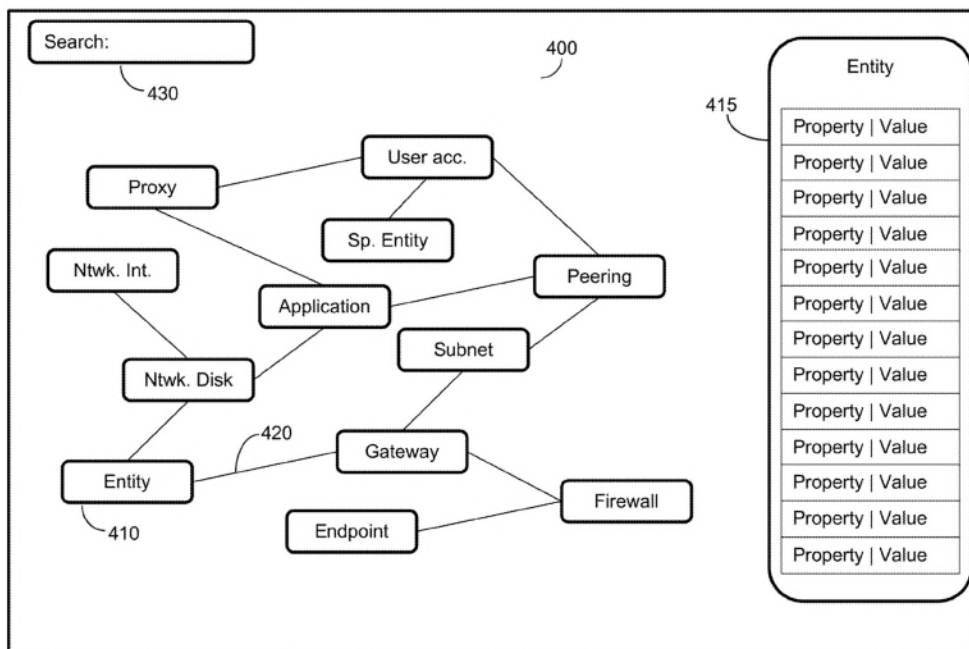


FIG. 4A

Id., Fig. 4A; *see also id.*, 16:41-67, Fig. 4B (showing object relationships in list format); EX1002, ¶¶65-66.

6. Elements 1.4, 11.4, and 20.4

Keren teaches these elements, which recite “[access/accessing] a data structure identifying services publicly accessible via the [I/i]nternet and capable of serving as an internet proxy.” The analysis below first addresses how a POSA would have understood from Keren, or at least would have found it obvious from its teachings, that the cybersecurity system accesses a data structure that stores security object data that is used for generating or updating constructed security graphs. The analysis then addresses how the stored security object data identifies services publicly accessible via the internet and capable of serving as an internet proxy. EX1002, ¶67.

First, as explained above, Keren teaches that the cybersecurity system identifies security objects and collects security object data that is then used to generate security graphs. EX1005, 9:26-27, 10:27-33, Fig. 2; *supra*, §§X.A.2-4; EX1002, ¶68. In addition to using this collected object data to determine the network relationships between the identified security objects and building a security graph connecting the objects based on those relationships (*supra*, §X.A.5), as explained more below in §X.A.7, Keren also teaches that the security graph may be updated (or integrated) with recordings (e.g., by using tags) that detail

properties of the security objects, which may be based on the collected security object data. *See, e.g.*, EX1005, 11:61-12:10, 13:6-25. A POSA would have understood that, to perform this analysis, the collected security object data would be stored in a memory storage or database (i.e., a data structure) so that the cybersecurity system could access and process the data for integrating into the security graph. EX1002, ¶68.

Keren suggests at least two means by which the collected security object data may be stored and accessed in a data structure. EX1002, ¶69. As noted above (*supra*, §X.A.I), the cybersecurity system generally provides “network analysis functionalities,” which includes functions relating to the querying and collection of security object data, network analysis, and graph generation. EX1005, 6:47-51; EX1002, ¶69. To perform these functions, Keren teaches that the cybersecurity system includes “a storage 630.” EX1005, 20:61-67, Fig. 6. A POSA would have understood Keren as disclosing, or at minimum rendering obvious, storing the security object data collected by the system in this storage so that the cybersecurity system could later access the data for subsequent analysis and integration into a security graph. EX1002, ¶69; *see also, e.g.*, EX1005, 9:26-27 (“[S]ecurity objects are identified, and security object data is collected.”). In addition, Keren further teaches the use of a “graph database (graph DB),” which is “configured to provide one or more functionalities relevant to storage of graph-related data” and “store

graph-related data features of one or more types or formats,” such as “raw data, graphs, graph edges, graph vertices, graph schemas, and the like.” EX1005, 7:3-11; *see also id.*, 7:28-30, 7:67-8:3 (explaining graph DB “may be directly connected to, or realized as a component of, the graph analysis system 150”); EX1002, ¶70. Generated security graphs are stored in the graph database, which, as Keren explains, may be updated on an ongoing basis as security object data is processed and analyzed. EX1005, 11:11-17 (graphs “may be updated ... by associating one or more data labels, tags, or other, like, features with a graph entry for a security object”), 11:61-62, 15:36-43 (reachable properties may be populated into graph through labelling or tagging), 16:15-21 (graph may be updated to include calculated properties); EX1002, ¶70. These updates include the integration of tags, which may include descriptions of object details as determined by the collected security object data. EX1005, 13:6-25 (tags include “object details, such as may be collected at S210”), Fig. 2. In view of these teachings, a POSA would have also understood that collected security object data would be stored as “raw data” in the graph database (*see id.*, 7:3-11) that the cybersecurity system would access and process during the generation of security graphs, including updating stored security graphs with object-specific tag information. EX1002, ¶70. Thus, a POSA would have understood Keren teaching, or at minimum rendering obvious, “accessing a data structure” as claimed. EX1002, ¶70.

Second, a POSA would have understood that the stored security object data accessed by the system identifies services publicly accessible via the internet and capable of serving as an internet proxy. EX1002, ¶71. The '257 patent describes that “[s]ervices capable of serving as an internet proxy ... may refer to a system or router that provides a gateway between users and the internet” and “[a] proxy ... may refer to a device or system that translates traffic between networks and protocols.” EX1001, 35:8-18, 36:64-66. As Dr. Stavrou explains, consistent with the '257 patent, the security objects described by Keren include objects that provide services publicly accessible via the internet and capable of serving as an internet proxy (e.g., by acting as a gateway between users and the internet). EX1002, ¶71. For example, Keren explains that the security objects can include Nginx (an HTTP web server), web-access firewalls (WAFs), public proxies, and entities that “process layer-7 (application layer) traffic, such as objects relevant to Amazon AWS® layer seven services and applications, Amazon Load Balancer® (ALB) layer seven services and applications, Kubernetes ingress, and the like.” EX1005, 5:56-6:4, Figs. 4B (illustrating internet-facing proxy for routing traffic), 5 (illustrating public proxy 540 for routing internet traffic). A POSA understood that these security objects provided management and routing services of internet traffic to and from resources within the network, and thus, provided “services that are publicly accessible via the internet and capable of serving as an internet proxy,” as

recited. EX1002, ¶71 (citing EX1009 (Amazon Load Balancer provides load balancing services of HTTP and HTTPS traffic); EX1010 (Kubernetes Ingress provides routing services of HTTP traffic); EX1016 (Nginx is an HTTP web server)); cf. EX1001, 14:19-25 (identifying web server Nginx as a type of service), 35:8-18, 36:64-66.

Keren also teaches that the security object data collected and stored by the system identifies these services. EX1002, ¶72. For example, the collected security object data may include object types, object names, rules, event logs, object port configurations, protocols, and internal libraries of the security objects. EX1005, 6:5-9 (security objects include communication ports that “provide for connection of various objects according to one or more protocols, and at different communication layers of the OSI model”), 9:58-66, 10:11-26 (data collected includes rules, event logs, port configurations, and internal data resources/libraries); *see also supra*, §X.A.4. A POSA would have understood that this collected data would have identified services that were publicly accessible via the internet and capable of serving as an internet proxy. EX1002, ¶72. As an example, collecting object names of security objects would indicate whether a given object was a Nginx web server, an Amazon Load Balancer, or a Kubernetes Ingress. EX1005, 5:56-6:4. These object names would identify that the services provided by these objects were publicly accessible via the internet and capable of

serving as an internet proxy because, as noted above, these objects were known to service internet traffic for a network. EX1002, ¶72 (citing EX1009; EX1010; EX1016); *cf.* EX1001, 14:19-25 (identifying web server Nginx as a type of service), 36:64-66 (internet proxy services refer to systems providing a gateway between users and the internet), 35:8-18 (a proxy translates traffic between networks or protocols).

Similarly, collecting object type data would indicate whether an object was a web-access firewall or a public proxy, which were commonly used resources providing services for managing internet traffic. *See* EX1005, 5:56-6:4, 19:56-20:12, Fig. 5; EX1002, ¶73 (citing EX1011 (internet traffic management services of public proxies); EX1012, 1, 4 (same); EX1009, 4 (AWS WAF protects web applications); EX1013, 1 (WAFs inspect and filter internet traffic)). An object's port configurations and the protocols handled by the object would also identify whether the services provided by the object were publicly accessible via the internet and capable of serving as an internet proxy. EX1002, ¶73 (citing EX1014, 1 (identifying open ports can identify "what kind of services ... are running" on the system); EX1015, 1 ("Each port is associated with a specific process or service.")); *cf.* EX1001, 14:19-25, 36:51-55 (recognizing port numbers can "identify a specific process to which an internet or other network message is to be forwarded when it arrives at a server and/or which services are running on a server."). For instance, a

port configuration in which port 80 is in use would identify that the object services internet traffic. EX1002, ¶73 (citing EX1014, 1-2; EX1015, 2, 5).

Additionally, Keren teaches that security object relationships and connectivity may be determined by monitoring network traffic or other logging methods. EX1005, 10:14-20 (security object data may include “collection of firewall event logs”), 11:56-60 (“[O]bject relationships may be determined at S230 by application of observational or active logging methods, such as those methods providing for detection of object-to-object connections by monitoring traffic of a network in use.”); EX1002, ¶74. A POSA would have understood that these active logging methods may include, proxy logs, web-access firewall (WAF) logs, web server logs, load balancer logs, or logs from other services provided by a third party such as Amazon Web Services or Microsoft Azure. EX1002, ¶74. These types of logs identify services publicly accessible via the internet and capable of serving as an internet proxy, as recited. *Id.* (citing EX1017, 6 (proxy logs detail network events such as internet traffic denials); EX1018, 1 (web server logs record HTTP requests); EX1019, 2-5 (information captured by AWS WAF logs)). Thus, the stored security object data identifies services publicly accessible via the internet and capable of serving as an internet proxy as recited. EX1002, ¶¶74-75.

7. Elements 1.5, 11.5, and 20.5

Keren teaches these elements, which recite “[integrate/integrating] the identified services into the graph.” EX1002, ¶76.

Keren teaches that the security object data, including the data identifying the services discussed above and accessed in data structures such as Keren’s storage 630 and/or graph DB 160 (*supra*, §X.A.6), is integrated into the security graph. EX1002, ¶77. Specifically, Keren explains that the security objects contained within the generated security graph may “include a recording describing its properties,” such as “‘known network properties’ (information that can be retrieved from querying the entity or the cloud platform)—i.e., the collected security object data. EX1005, 11:61-12:10. Keren teaches this is accomplished by “[t]agging of security objects” such as by “association of one or more data labels, tags, or other, like, features, with graph entries for one or more security objects” (i.e., integrating the data into the graph). *Id.*, 13:6-11; EX1002, ¶77; *cf.* EX1001, 37:3-11 (“[S]canning system 101 may integrate information on services available in each node into the data structure, associated with each node in the data structure.”). “Examples of relevant descriptions” that may be integrated into the graph include “object details, such as may be collected at S210, ... descriptions of an object’s open ports and counts thereof,” and “descriptions of protocols relevant to an object,” which a POSA would have understood would include the security object

data identifying services discussed above. EX1005, 13:12-25; *see also id.*, 11:14-23 (data label descriptions may include “connection source and destination, connection type, connection direction, connection status, connection protocol, and the like.”), 13:43-46, 14:42-46; *supra*, §X.A.6; EX1002, ¶77.

Further illustrating that the identified services are integrated into the graph is Keren’s calculation of a security object’s reachability properties. EX1002, ¶78; *infra*, §X.A.8 (further detailing reachability properties). Specifically, Keren teaches that a network path within the security graph may be determined based on the generated graph and the integrated object properties. EX1005, 13:26-65. An example path determination is shown in Fig. 5 (reproduced below), in which the reachable properties of a database 510 are determined. *Id.*, 4:1-2, 15:15-35, 18:40-43, Fig. 5; EX1002, ¶78. The database’s reachable properties are calculated based on the properties of objects within the network path, which includes a public proxy 540 having a translation rule in which internet traffic into a frontend port 1234 is translated to a backend port 4321 (i.e., an identified service publicly accessible via the internet and capable of serving as an internet proxy). EX1005, 18:55-58, 19:62-20:12, Fig. 5; *see also infra*, §X.A.8; EX1002, ¶78.

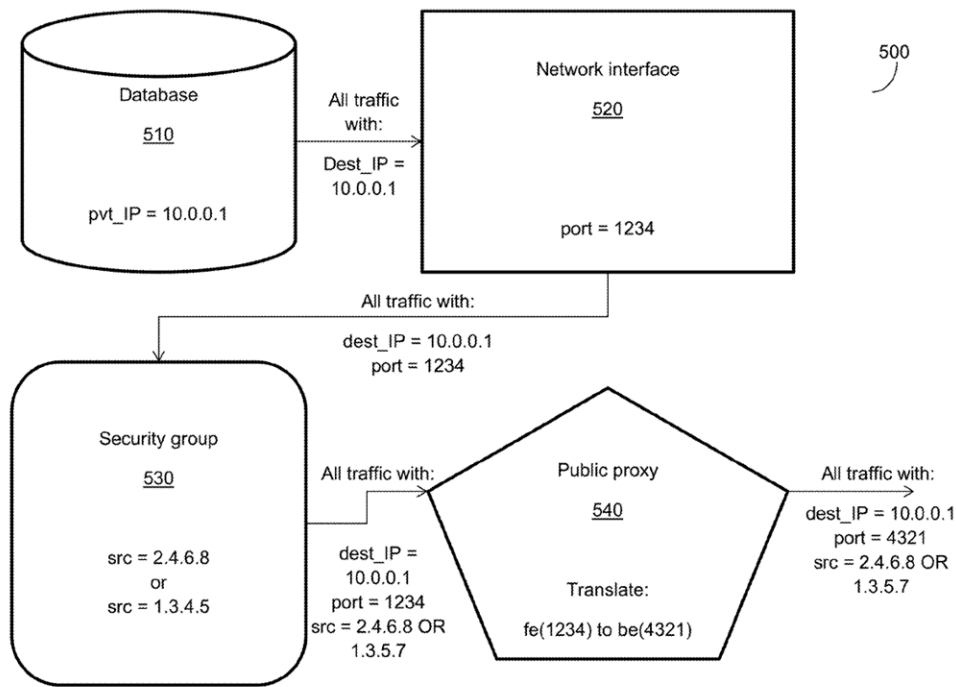


FIG. 5

EX1005, Fig. 5.

Keren's description of user visualization features, shown in Figs. 4A-4B, also demonstrates the integration of the identified services into the security graph. EX1002, ¶79. Keren teaches, for example, that a "security graph utility" (reproduced below) may be provided as "an application, interface, or other, like, means of providing a visual representation of a security graph" and may include various interactive functionalities, such as a display functionality where a user may click on a security object (represented by a node vertex visualization 410) to display a vertex overview pane 415. EX1005, 16:41-53, 17:5-8, Fig. 4A. The vertex overview pane "may be an information panel, including data relating to the

given vertex visualization 410 and describing various object data features, such as those object data features identified or determined during the execution of a process similar or identical to that described with respect to FIG. 2.” *Id.*, 15:8-13; *see also id.*, 15:13-18; EX1002, ¶79.

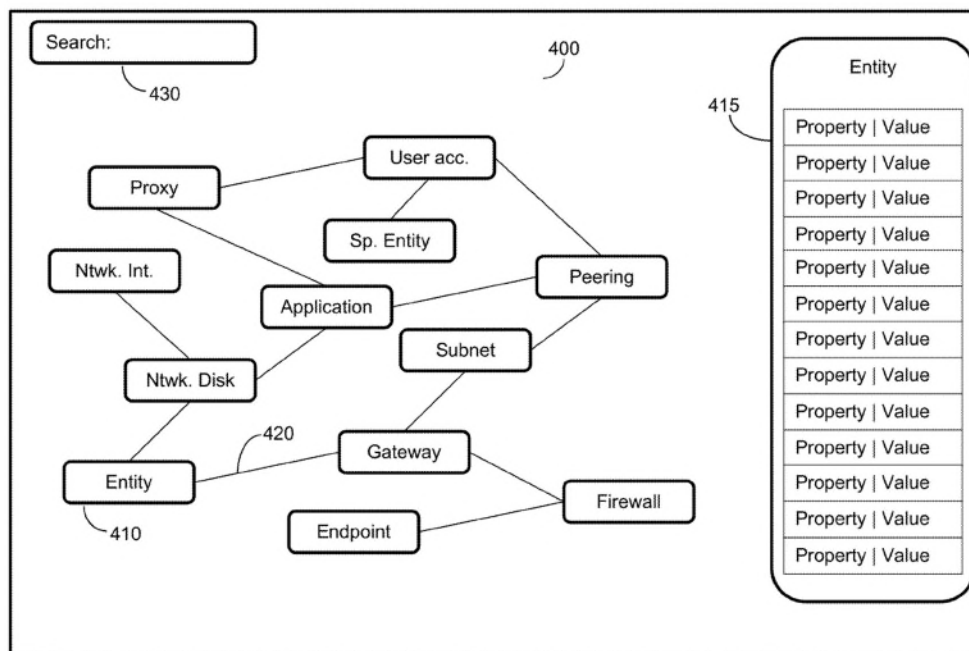


FIG. 4A

EX1005, Fig. 4A.

Keren also describes that a mode-specific feature may be provided, where a user may select “list-organization profiles,” which provides “for population of a security graph object list 450 in one or more configurations.” EX1005, 18:8-22; EX1002, ¶80. One example, as shown in Fig. 4B (reproduced below), is “a ‘route’ mode” that provides a security graph object list 450 representing “a data path between a first object 470-1 and a destination object 470-n.” EX1005, 18:23-27;

see also *id.*, 17:39-43. In the illustrated example, the mode-specific data display 465 displays “‘9 hops,’ indicating that a transmission from a first object 470-1,” a container, “makes nine ‘hops,’ or transmissions between objects, before reaching the destination object 470-n,” the internet. *Id.*, 18:27-39, Fig. 4B. As shown in Fig. 4B, a proxy services transmission from the container (and intervening security objects) to the internet. EX1002, ¶¶80.



FIG. 4B

EX1005, Fig. 4B.

Both cases thus provide further demonstration that the identified services are integrated into the constructed security graph so that the system may provide visualization features to a user. EX1002, ¶¶80-81.

8. Elements 1.6, 11.6, and 20.6

Keren teaches these elements, which recite “[traverse/traversing] the graph to identify at least one [source/vector] originating via the [I/i]nternet and reaching the workload.” EX1002, ¶82.

As an initial matter, claims 1 and 11 recite identifying “at least one source” originating via the internet, while claim 20 recites identifying “at least one vector” originating via the internet. The ’257 patent explains that a “source” originating via the internet may be “a potential malicious source,” where, in some embodiments, “the Internet at large may be considered to be a potentially malicious source.” EX1001, 37:25-31; *see also id.*, 38:15-20. The ’257 patent describes a “vector” as “connecting the plurality of systems and devices” in the graph, where the scanning system “may traverse the map to identify vectors originating in the Internet and reaching the devices and systems.” *Id.*, 12:65-13:10; *see also id.*, 36:30-39. Keren teaches both these elements. EX1002, ¶83.

Keren teaches that the system calculates a “reachability property” of the security objects represented in the graph based on the constructed graph and integrated object information discussed above. EX1005, 11:62-12:5, 13:35-46, Figs. 3A-B; *supra*, §§X.A.5, X.A.7; EX1002, ¶84. A security object’s reachability property “defines if and how an entity can be reached from an external or internal network,” where the external network may be the internet. EX1005, 12:5-7, 14:1

(“an external network (the internet)”); *see also id.*, 13:26-34, 15:15-24. As detailed more below, to calculate a security object’s reachability property, Keren’s system traverses the graph to identify a source or vector originating from the internet and reaching the security object. EX1002, ¶84. A security object, as explained above, includes workloads, such as virtual machines, containers, load balancers, storage, and databases. *Supra*, §X.A.1; EX1002, ¶84.

Specifically, Keren teaches accessing a security graph stored in the graph database, where each security object in the graph is associated with known properties of the object, such as those discussed above. EX1005, 13:26-46; *supra*, §§X.A.6-7; EX1002, ¶85. Based on this information, the system identifies “possible network paths within the obtained security information,” in which each “path shows a connection between security objects and/or computing objects that communicate over a network” (i.e., a vector). EX1005, 13:47-48, 13:62-65; *see also id.*, 13:51-61 (“[A] network path may include sequential representations of possible data/control flows between two or more objects in a graph.”). As Keren explains, “paths can be identified by traversing the security graph.” *Id.*, 13:66-67 (emphasis added). “The traversal can start or end at objects that are connected to an external network (the internet),” and can be performed using known means. *Id.*, 13:67-14:5; EX1002, ¶85.

Keren describes that each path is analyzed to calculate the reachability properties of the security objects, including workloads, contained within the path. EX1005, 14:15-26, 14:31-58, Fig. 3B; *see also id.*, 14:59-15:2; EX1002, ¶86. Keren explains that a reachability property identifies sources that may reach an object by determining a “minimal set of reachable properties of all other objects in the path.” EX1005, 15:25-27. For example, if a path includes two objects where a first object can “receive traffic from any source IP address through port 1515” and a second object “can receive traffic only from a source IP address of 173.54.189.188,” then the second object’s reachable property is “**source IP address 173.54.189.188** and port 1515.” *Id.*, 15:27-33 (emphasis added); EX1002, ¶86.

Furthermore, Keren expressly teaches that a reachability property for an object may be calculated based on whether and how an object is accessible from a source (or vector) originating from the internet. EX1002, ¶87. For example, Keren teaches that reachability properties for objects may be populated into the security graph as “object network access control lists (NACLs).” EX1005, 15:36-50. The NACLs may be “[s]cope-specific” and “describ[e] object, path, or network accessibility properties specific to a given scope, where the given scope may be the internet.” *Id.*, 15:50-53. Thus, an NACL having a scope of the internet would “describe the properties of an object with respect to the object’s internet

accessibility.” *Id.*, 15:53-58; *see also id.*, 20:34-42; EX1002, ¶87; *cf.* EX1001, 37:29-31 (the internet at large may be a source). As a further example, Fig. 5 (shown below) specifically illustrates a traversed graph path that identifies at least one source (or vector) originating via the internet and reaching a workload (i.e., database 510).

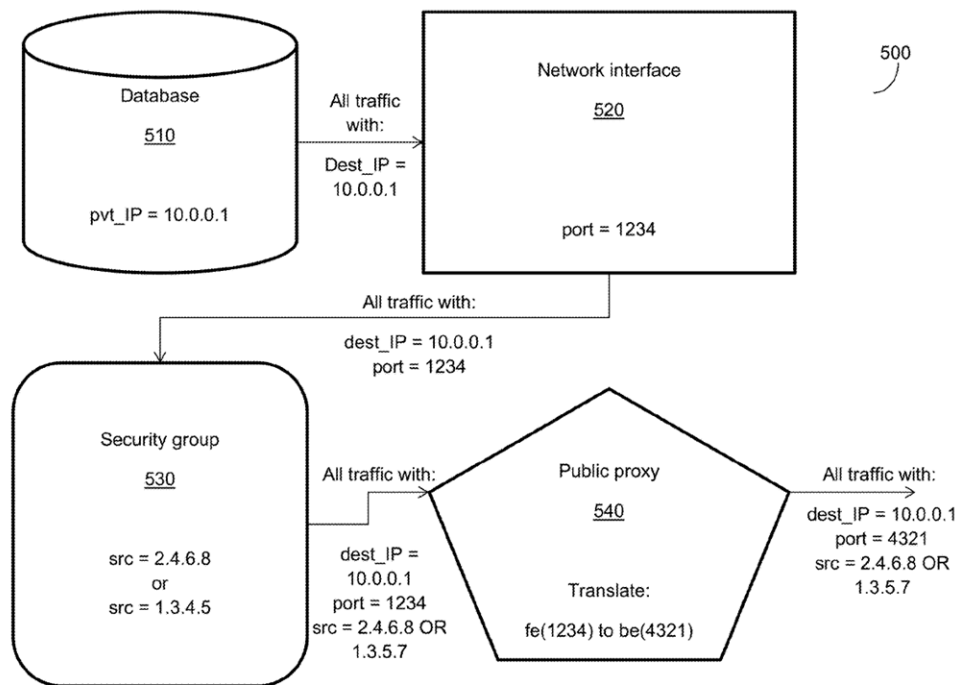


FIG. 5

EX1005, 18:40-43, Fig. 5; EX1002, ¶88.

As Keren explains, each of the objects within the illustrated network path of Fig. 5 includes accessibility properties that determine the source of internet-originating traffic that can be received at database 510. *See* EX1005, 18:59-19:18; EX1002, ¶89. In the example, by considering each of the object properties within

the path, the system determines that traffic matching only specific conditions (i.e., “all traffic with a destination IP address of 10.0.0.1, a port number of 4321, and a source of 1.3.5.7” and “all traffic with a destination IP address of 10.0.0.1, a port number of 4321, and a source of 2.4.6.8”) is allowed to reach the database 510.

EX1005, 18:44-58, 19:34-20:21; EX1002, ¶89. Based on this analysis, other reachability properties of the database may also be determined, such as the number of ports open to the internet (“numPortsOpenToInternet=1”) and whether the database is accessible from the internet (“accessibleFromInternet=True”).

EX1005, 20:5-12; EX1002, ¶89. Thus, Keren teaches both identifying a source originating via the internet (e.g., sources 1.3.5.7 and 2.4.6.8) and reaching the workload (e.g., database 510) and identifying a vector originating via the internet and reaching the workload (e.g., the data path entering public proxy 540 and reaching database 510), as recited. EX1002, ¶¶89-90.

9. Elements 1.7, 11.7, and 20.7

Keren teaches these elements, which recite “[output/outputting] a risk notification associated with the workload.” EX1002, ¶91.

The purpose of Keren is to provide risk notifications associated with workload objects in a network. *Id.*, ¶92. Keren explains that organizations relying on computer, network, and web technologies “may be increasingly vulnerable to cyber-threats.” EX1005, 1:22-25. Keren goes on to identify shortcomings of prior

network management systems, which “fail to provide network information which describes the role of an individual entity within such a network or a cloud environment as well as the vulnerabilities presented by the entities’ placements within the network configuration” to network administrators. *Id.*, 2:42-49. Thus, Keren provides a risk-notification solution “which allows for network analysis and includes granular, comprehensive representation of security object properties and insights,” including “reachability properties of security objects.” *Id.*, 2:49-55, 3:6-7; EX1002, ¶92.

Keren describes that a risk notification associated with the workload may be outputted by the system in the form of security insights. EX1002, ¶93. For example, Keren teaches that, “based on the known and calculated properties” of the security objects, “security insights may be generated,” which “are natural-language representations of aspects of the security graph constructed at S220.” EX1005, 12:11-14; *see also id.*, 12:14-44; EX1002, ¶93. In providing “a solution which allows ... granular, comprehensive representation of security object properties and insights,” Keren teaches, or at the very least suggests, that these insights are outputted as notifications to a user. EX1005, 2:49-55; EX1002, ¶93.

Specifically, Keren describes various ways in which users may be notified of information relevant to the network analysis performed by the system, including the generated security insights. EX1002, ¶94. For example, Keren teaches that the

security insights may be tagged to security objects in the security graph. EX1005, 13:6-25 (“Tagging of security objects ... may include, without limitation, association of one or more data labels, tags, or other, like, features, with graph entries for one or more security objects....”); EX1002, ¶94. Keren also teaches that the security graph and its associated data may be outputted as a “security graph visualization” to the user, as shown below. EX1005, 16:41-53.

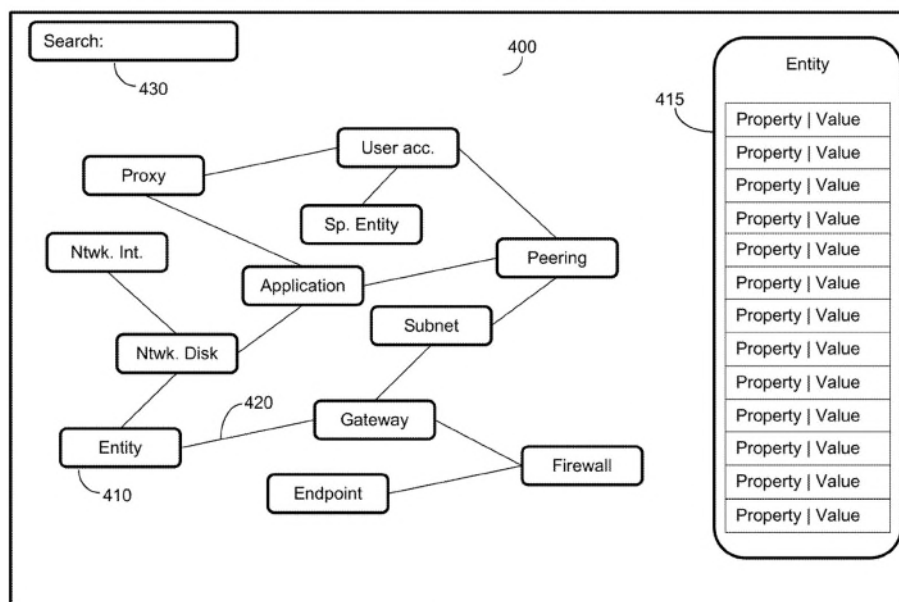


FIG. 4A

Id., Fig. 4A.

The security graph visualization can notify the user of relevant information, including tagged security insights, through various features, including an information panel and a search function. EX1002, ¶95. Keren explains that the security graph visualization includes a visual representation of the security objects (represented by vertex visualizations 410) and their relationships (represented by

edge visualizations 420). EX1005, 16:54-62; EX1002, ¶95. A user can select a security object in the graph (by, for example, clicking on a vertex visualization 410) to display a vertex overview pane 415, which is an information panel that notifies the user of “data relating to the given vertex visualization 410” and “describ[es] various object data features.” EX1005, 17:1-13. Keren teaches that the object data features may include “features identified or determined during the execution of a process similar or identical to that described with respect to FIG. 2” (*id.*, 17:10-18), which includes features provided in data labels or tags associated with the security graph. *Id.*, 11:14-17 (stating “[g]raphs may be updated at S230 by associating one or more data labels, tags, or other, like, features with a graph entry for a security object”); EX1002, ¶95. As noted above, Keren teaches that tags associated with the security graph include security insights generated by the system. EX1005, 13:6-25. Keren also explains that the security graph visualization may include “a search tool 430” that allows the user to search for user-specified object data features, such as “object names, types, IDs, statuses, labels or tags associated with various elements of the security graph visualization 400, and the like.” EX1005, 17:19-34; EX1002, ¶96. Thus, Keren teaches that a user may be notified of the generated security insights through an information panel associated with a given security object or through a search tool. EX1002, ¶96.

Moreover, Keren contemplates providing functionalities that generate alerts associated with the cloud environment. *See* EX1005, 8:54-63 (describing an orchestrator 115 that “generat[es] cloud platform 110 alerts”); EX1002, ¶97. Given that Keren teaches that an aim of its cybersecurity system is to provide network administrators with network information that “describes the role of an individual entity within such a network or a cloud environment, as well as the vulnerabilities presented by the entities’ placements within the network configuration” (EX1005, 2:42-55), a POSA would have found it obvious to also notify a user of security insights—particularly those describing a risk to a workload, explained more below—via an alert so that the user can be alerted to vulnerabilities presented by a current network configuration. EX1002, ¶97. Accordingly, in view of the above teachings, Keren teaches, or at the very least renders obvious, notifying users of the security insights generated by the system. EX1002, ¶97.

Keren further explains that the generated security insights may provide network descriptions that a POSA would have understood to be a notification of risk to workloads contained within the cloud environment. EX1002, ¶98; *see also* EX1005, 2:37-49 (teaching need to notify network administrators of “the vulnerabilities presented by the entities’ placements within the network configuration”). Keren explains that insights may include descriptions of “network

management or anomaly detection data, descriptions of network configurations or events which are rare or novel,” “descriptions of connections which are unauthorized,” and “descriptions of connections which display anomalous behavior.” EX1005, 12:45-55; EX1002, ¶98. As one specific example, Keren discloses generating an insight for a virtual machine (i.e., a workload) that provides notification of the fact that “VM thirty normally connects to load balancer twenty and firewall eight but is currently only connected to load balancer twenty.” EX1005, 12:57-61; EX1002, ¶98. A POSA would have understood that this insight provided a risk notification associated with the workload because the insight would notify a user that the virtual machine was no longer connected to a network security device designed to prevent unauthorized access to the workload (i.e., a firewall). EX1002, ¶98 (citing, e.g., EX1020, 17, 37 (explaining firewalls strengthen network security perimeter); EX1021, ¶33 (teaching use of firewall to prevent unauthorized asset exploitation); EX1025, ¶¶22-25 (discussing use of passive countermeasures, such as firewalls, to protect against vulnerability exploitation)). Keren additionally teaches generating “high-level insights,” such as a notification that ““objects E, a database, and F, an administrator interface, are currently exposed to external networks.”” EX1005, 12:62-13:5. A POSA would have understood such high-level insights would provide a risk notification that a workload containing sensitive information or providing privileged access (e.g., a

database or administrator interface) was exposed to an external network (e.g., the internet). EX1002, ¶¶99-101; *cf.* EX1001, 37:29-31 (“[T]he Internet at large may be considered to be a potentially malicious source.”).

Accordingly, claims 1, 11, and 20 were obvious.

B. Dependent Claims

1. Claims 2-3 and 12-13

Claims 2 and 12 depend from claims 1 and 11, respectively, and further recite “wherein the plurality of entities includes a virtual network appliance.”

Claims 3 and 13 depend from claims 2 and 12, respectively, and further recite “wherein the virtual network appliance is at least one of a load balancer, a firewall, a proxy, or a router.” EX1002, ¶102.

As explained above, Keren teaches a plurality of entities capable of routing traffic, such as firewalls, proxies, and load balancers. *Supra*, §X.A.3; *see also* EX1005, 5:32-39, 6:10-19, 9:6-15, 10:11-20; EX1002, ¶103. Furthermore, Keren teaches that these objects may be “virtual entities or instances of systems, devices, or components, including virtual systems, devices, or components, or any combination thereof.” EX1005, 6:10-19; *see also id.*, 5:45-51 (security objects can provide OSI layer seven network communication support), 5:66-6:9 (security objects may include “virtual entities, devices, and the like, to process layer-7 (application layer) traffic, such as objects relevant to Amazon AWS® layer seven

services and applications, Amazon Load Balancer® (ALB) layer seven services”), 9:6-15 (security objects include virtual entities such as load balancers), 13:64-65 (an object may be “a virtual ... entity”); EX1002, ¶103. These teachings of virtual security objects, as described by Keren, are consistent with a “virtual network appliance” as described by the ’257 patent. *See* EX1001, 34:38-51 (virtual network appliance refers to “a machine focused on virtualizing network functionality” directed to OSI layers four through seven, such as “a firewall, WAN optimizer, application delivery controllers, routers, load balancers, IDS/IPS, [or] proxies”), 34:52-53; EX1002, ¶104. Thus, Keren teaches a plurality of entities that includes a “virtual network appliance,” such as a load balancer, firewall, or proxy, as recited. EX1002, ¶¶104-05.

Accordingly, claims 2-3 and 12-13 were obvious.

2. Claims 4 and 14

Claims 4 and 14 depend from claims 1 and 11, respectively, and further recite “wherein the networking configuration is at least one of a routing configuration, a proxy configuration, a load balancing configuration, a firewall configuration, or a VPN configuration.” EX1002, ¶106.

As discussed above, Keren teaches determining the networking configuration of the plurality of entities (e.g., security objects providing traffic-management functionalities), which may include “object network addresses,”

“object port configurations,” “object [input/output] signal configurations,” and “object security or access configurations.” EX1005, 5:32-37, 9:56-10:10; EX1002, ¶107; *supra*, §X.A.4. Also discussed above, Keren teaches that the entities may include firewalls, proxies, load balancers, routing tables, and VPNs. EX1005, 5:38-39, 5:66-6:4, 6:14-19, 9:9-15; *supra*, §X.A.3. Because Keren teaches determining the networking configuration of these entities, Keren teaches that “the networking configuration is at least one of a routing configuration, a proxy configuration, a load balancing configuration, a firewall configuration, or a VPN configuration,” as recited. *See, e.g.*, EX1005, 10:14-20 (describing collection of a firewall’s rules, event logs, and port configurations), 19:62-20:5 (describing determination of public proxy’s port configuration), Fig. 5; EX1002, ¶¶107-08.

Accordingly, claims 4 and 14 were obvious.

3. Claims 5 and 15

Claims 5 and 15 depend from claims 1 and 11, respectively, and further recite “wherein the graph includes a data structure sequentially connecting entities.” EX1002, ¶109.

As explained above, Keren teaches building a security graph, which is a data structure that describes the various security objects included within the virtual environment and the relationships between those objects. EX1005, 10:27-33, Fig. 2; *supra*, §X.A.5; EX1002, ¶110. Keren teaches that the security graph may be

constructed in various formats, including formats where the data structure sequentially connects the security objects. EX1005, 10:33-48; *see also* EX1001, 36:26-29 (“A data structure sequentially connecting entities, as used herein, may refer to an organization, management, and/or storage format that enables efficient access and modification of data such as entities.”); EX1002, ¶110. For example, the security graph may be a “visual ‘node and link’ graph,” in which security objects, represented by nodes, are sequentially connected along any given path via links. EX1005, 10:33-48, 11:28-37, Fig. 4A; *see also id.*, 7:15-24. Keren also teaches that the security graph may provide a routing display feature that sequentially connects security objects along a specific data path. *Id.*, 18:23-39, Fig. 4B. Moreover, Keren discloses that the network paths identified by the system “may include sequential representations of possible data/control flows between two or more objects in a graph.” *Id.*, 13:51-53, Fig. 5; *see also supra*, §X.A.8; EX1002, ¶¶110-11.

Accordingly, claims 5 and 15 were obvious.

4. Claims 6 and 16

Claims 6 and 16 depend from claims 1 and 11, respectively, and further recite “wherein the graph includes directional vectors indicating directions of dataflow.” EX1002, ¶112.

Keren teaches that the security graphs may “include one or more functional attributes,” such as “directionality.” EX1005, 7:30:36; EX1002, ¶113. “A graph including directionality may be configured to include connection between graph nodes or vertices, via graph edges, ... where the edges connecting such vertices may be uni-directional or bi-directional, providing for enhanced analysis of security object structures and relationships.” EX1005, 7:36-42; *see also id.*, 11:17-23 (graph may include labels indicating “connection direction”). For instance, Keren describes that a virtual machine configured to run a container may include a uni-directional edge (i.e., a directional vector) between the virtual machine and the container that would indicate how data flows between the virtual machine and the container (e.g., that the virtual machine runs the container). *Id.*, 7:46-56; EX1002, ¶¶113-14.

Accordingly, claims 6 and 16 were obvious.

5. Claims 7 and 17

Claims 7 and 17 depend from claims 1 and 11, respectively, and further recite similar limitations. Claim 7 recites “to build the graph, the at least one processor is configured to: identify individual entities as nodes; and connect the nodes.” Claim 17 recites “wherein building the graph comprises: identifying individual entities as nodes; and connecting the nodes.” EX1002, ¶115.

Keren teaches that the security graph may be constructed as “a visual ‘node and link’ graph,” as shown below. EX1005, 11:28-37, Fig. 4A. To construct this type of security graph, Keren teaches that the identified individual security objects (including the plurality of entities) are represented as nodes in the graph (e.g., node 410) and the determined object relationships are represented by links or edge connections between these nodes (e.g., link 420). *Id.*, 11:64-66, 13:36-42, 16:54-67; *supra*, §X.A.5; EX1002, ¶¶116-17.

Accordingly, claims 7 and 17 were obvious.

FIG. 4A

6. Claims 8 and 18

Claims 8 and 18 depend from claims 1 and 11, respectively, and further recite “wherein the graph includes port numbers associated with the workload.” EX1002, ¶118.

Keren teaches that the security object data collected by the system may include port configurations associated with the security objects (including workloads). EX1005, 9:56-66; *see also id.*, 10:20-26 (describing collection of “VM port configurations”); EX1002, ¶119. Keren further teaches that the security object relationships may be based on the port configurations of the security objects, and descriptions of object relationships may be integrated into the graph. EX1005, 10:52-57 (“Network relationships may describe aspects of the connections between objects including ... relevant ports of connected objects ...”), 11:11-17, 11:66-12:1, 12:11-12, 13:6-25, 14:42-46, 15:19-24 (reachable properties may include “ports by which an element is reachable”), 15:36-37. As one example, Keren describes an insight that specifies, in part, “a path from ‘virtual machine one to load balancer five, where port eighty is routed to port 1337 on virtual machine one....’” *Id.*, 12:26-44; *see also id.*, 12:22-25 (insight describing gate connected to a VM “via the second port”), 19:66-20:12; *supra*, §X.A.9 (insights are added as tags to the security graph); EX1002, ¶¶119-20.

Accordingly, claims 8 and 18 were obvious.

7. Claims 9 and 19

Claims 9 and 19 depend from claims 1 and 11, respectively, and further recite “wherein the graph includes a path from the at least one source to the workload.” EX1002, ¶121.

As explained above, Keren teaches that the system is configured to identify network paths between security objects, including workloads, by traversing the security graph. EX1005, 13:47-14:16; *supra*, §X.A.8. For example, as detailed above, Keren describes the analysis of a path in a security graph from a source originating via the internet to a database. EX1005, 18:40-43, 19:34-20:12, Fig. 5; *supra*, §X.A.8; EX1002, ¶¶122-23.

Accordingly, claims 9 and 19 were obvious.

8. Claim 10

Claim 10 depends from claim 1 and further recites “wherein the risk notification includes one or more resolution recommendations.” As described below, Keren renders this limitation obvious. EX1002, ¶124.

As explained above for the independent claims, Keren teaches outputting security insights that may notify a user of a risk associated with a workload, such as network configurations or events that put a workload at risk. EX1005, 12:45-13:5; *supra*, §X.A.9. For example, the security insights may provide notification of rare or new events, “such as connection of a new device to a network,”

unauthorized events, “such as re-connection of a user’s device to a subnet which the user is not permitted to access,” or connections that display anomalous behavior, “such as connections displaying spikes of network activity.” EX1005, 12:45-55. The security insights may also provide notification as to whether important computing resources, such as a database or administrator interface, are exposed to an external network, or if third-party networks have access to an internal network. *Id.*, 12:62-13:5; *see also supra*, §X.A.9; EX1002, ¶125. A POSA would have understood an insight noting a current abnormal network condition relative to a normal condition to be a recommendation to update the configuration to the identified normal condition. EX1002, ¶125. For example, the described insight specifying “VM thirty normally connects to load balancer twenty and firewall eight but is currently only connected to load balancer twenty” conveys to the user a recommendation to connect VM thirty to firewall eight. EX1005, 12:45-55; EX1002, ¶125.

Moreover, a POSA would have found it obvious from Keren’s teachings to include one or more explicit resolution recommendations with the security insights. It was well-known that a strong network security perimeter provided a good defense mechanism for mitigating the occurrence of risky network configurations and events described by Keren. EX1002, ¶126 (citing EX1020, 23-24 (“The network perimeter should be configured to deny all activity that is not expressly

permitted. This includes securing all connection points, such as virtual private networks (VPNs) and dedicated connections to other organizations.”)). POSAs were also aware of methods commonly employed to strengthen the security perimeter of a network, including, for example, installing firewalls to protect important assets and changing rulesets or access control lists of security devices to manage incoming network traffic. EX1002, ¶126 (citing EX1020, 17 (altering network security controls (e.g., firewall rulesets) for incident handling), 37 (incident prevention includes “tightening network perimeter security (e.g., firewall rulesets, boundary router access control lists)”); EX1021, ¶33 (changing installed firewall settings to protect against vulnerability exploitation); EX1025, ¶¶22-25). A POSA would have understood that these routine methods would aid in mitigating the network risks described in Keren. EX1002, ¶126. For example, connection of a firewall to resources, such as a virtual machine or database, would reduce the resource’s exposure to external networks, and updating access control lists of a security device would reduce the likelihood of unauthorized connections to an internal network. *Id.*

A POSA would have readily appreciated that including express resolution recommendations with the security insights taught by Keren would have been a straightforward and logical extension of Keren’s teachings. *Id.*, ¶127. Keren recognizes the benefits of providing management and alert functionalities to the

user and suggests combining these functionalities together. *See* EX1005, 8:55-63 (“The orchestrator 115 may be configured to provide one or more functionalities including, without limitation, monitoring of elements or components of the cloud platform 110, logging and reporting data relating to the cloud platform 110, managing cloud platform 110 updates and maintenance, generating cloud platform 110 alerts, as well as other, like, functionalities, and any combination thereof.”); EX1002, ¶127. Thus, to further achieve Keren’s aim in providing administrators with a system that can enable effective network management of an organization’s cloud environment (*see* EX1005, 1:36-49, 2:34-55), a POSA would have found it obvious to combine resolution recommendations relating to the management of a security object in relevant security insights generated for that object. EX1002, ¶127.

For example, in a security insight specifying that “VM thirty normally connects to load balancer twenty and firewall eight but is currently only connected to load balancer twenty” (EX1005, 12:57-61), a POSA would have readily appreciated that providing an express recommendation that VM thirty be re-connected to firewall eight in the security insight would beneficially aid the administrator in addressing the detected network risk. EX1002, ¶128. Similarly, in security insights notifying a user of unauthorized connections (e.g., third-party networks) to an internal network, a POSA would have understood that including as

part of the security insight an explicit resolution recommendation that access control lists governing access into the network should be updated would also improve the response to addressing the detected risk. *Id.* Thus, in view of Keren’s teachings, a POSA would have found it obvious to generate a risk notification that “includes one or more resolution recommendations,” as recited. *Id.*, ¶¶128-29.

Accordingly, claim 10 was obvious.

XI. GROUND 2: CLAIMS 1-20 WERE OBVIOUS OVER KEREN AND ENGELBERG

This ground builds upon the analysis of Keren discussed in Ground 1, further applying Engelberg’s disclosure of including one or more resolution recommendations in a risk notification. In §§XI.B-C, these teachings are applied to independent claims 1, 11, and 20 and dependent claim 10, with parallel citations to Engelberg and its provisional (EX1007) showing support in the provisional for the relied-upon disclosure in Engelberg. The combination with Engelberg does not change the previous explanation in Ground 1 of why the other limitations of 1-20 were obvious (*supra*, §X), since the applied teachings of Engelberg are consistent with the cybersecurity systems and methods taught by Keren. EX1002, ¶¶130-31.

A. Reasons to Combine Keren and Engelberg

As an initial matter, the references are analogous. EX1006, ¶6 (implementations “are directed to mitigating cyber security risk in enterprise networks”); EX1007, ¶5; *supra*, §§IX.A-B; EX1002, ¶¶132-33. Engelberg is

therefore in the same field as Keren and the '257 patent, and its teachings are pertinent to issues addressed by the '257 patent since those teachings involve detecting security risks in computer networks and resources and identifying resolutions to those risks. *See, e.g.*, EX1001, 33:49-51 (“Aspects of this disclosure may include a cloud-based cybersecurity system for assessing internet exposure of a cloud-based workload....”), 37:32-45 (“[T]he risk notification includes one or more resolution recommendations,” which “may refer to a suggestion or proposal as to the best course of action in view of the identified risk.”).

Moreover, arriving at the claimed invention involved using known elements according to their known functions to achieve predictable results. EX1002, ¶134; *see KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 416-17 (2007). Providing risk notifications that included resolution recommendations for risks identified in a computer network, as taught by Engelberg, was well-known and predictable. EX1002, ¶134 (citing EX1022, 9:14-26, 10:27-34; EX1023, ¶¶210-211; EX1024, ¶¶60-63). Thus, the combination of Keren and Engelberg would have merely involved using the familiar technique of including a resolution recommendation in a risk notification generated by a cybersecurity system. EX1002, ¶134.

In addition, a POSA would have been motivated to apply Engelberg’s teachings to Keren to alert an administrator that a security insight identified by Keren’s cybersecurity system presented a risk, and further to provide additional

information about that risk by presenting a potential remediation activity.

Engelberg itself teaches these benefits. EX1002, ¶135. Engelberg explains that “[c]omputer networks are susceptible to attack by malicious users (e.g., hackers),” especially as “[e]nterprises, in general, and industrial infrastructures, in particular, are increasingly connected to external networks such as the Internet,” making them increasingly vulnerable to external cyber-attacks. EX1006, ¶2; EX1007, ¶1. As a result of these risks, Engelberg teaches that it is helpful to identify “mitigation efforts to mitigate risk in enterprise networks” and provide those recommendations to security personnel so that personnel can “understand immediate remediation actions corresponding to each issue.” EX1006, ¶¶6, 59 (providing notifications of risks and recommendations “enable the cyber security personnel to react to and remediate security relevant events”); *see also id.*, ¶¶30, 42; EX1007, ¶¶5, 28, 40, 57; EX1002, ¶135.

In view of Engelberg’s teachings regarding the benefits of notifying administrators of risks present in a network and providing resolution recommendations for the identified risk, a POSA would have had reason to similarly explicitly identify security insights generated by Keren’s cybersecurity system as risks and include resolution recommendations. EX1002, ¶136. As explained above, Keren teaches generating security insights that provide notification to risks present within the network, such as rare or novel network

configurations or events, unauthorized connections, or exposed assets. *Supra*, §§X.A.9, X.B.8. Keren explains that, in managing large network environments, these insights aid security personnel in understanding “the role of an individual entity within such a network or a cloud environment, as well as the vulnerabilities presented by the entities’ placements within the network configuration.” EX1005, 2:42-55; EX1002, ¶136. Engelberg similarly teaches notifying a user of risks associated with network or system configuration conditions. EX1006, ¶¶57, 59; EX1007, ¶¶55, 57. Thus, a POSA would have appreciated that Keren’s generation of security insights could be readily improved by expressly identifying security insights describing risky network configurations as risks in a notification and also including resolution recommendations to aid in network-management decision-making for security personnel, as taught by Engelberg. EX1002, ¶136.

A POSA would have had a reasonable expectation of success in combining the above teachings of Keren and Engelberg to arrive at the claimed invention. EX1002, ¶137. Keren recognizes that object management functionality can be readily combined with alert features. *See* EX1005, 8:56-63 (orchestrator can combine multiple functionalities, including object monitoring, object maintenance, and alert generation). As explained above, providing resolution recommendations with notifications regarding risks present in a computer network was well-understood and routinely used. EX1002, ¶137 (citing EX1022, 9:14-26, 10:27-34;

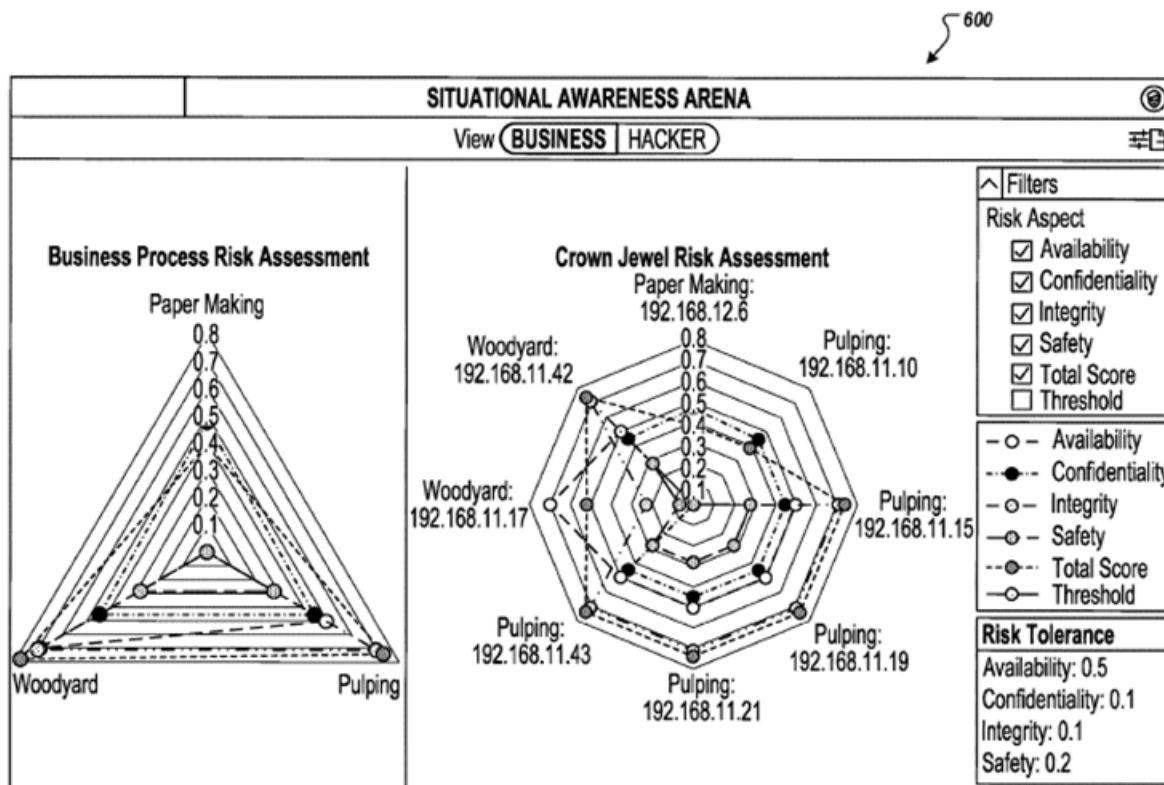
EX1023, ¶¶210-211; EX1024, ¶¶60-63). Moreover, a POSA was well-aware of the types of resolutions that would aid in mitigating risks such as those taught by Keren and would have found it routine to identify such resolutions for inclusion with the corresponding security insight. EX1002, ¶137 (citing EX1020, 17, 23-24, 37; EX1021, ¶33 (changing firewall settings to prevent vulnerability exploitation); EX1025, ¶¶22-25); *supra*, §X.B.8. As Dr. Stavrou explains, combining the references’ teachings as discussed above would not have presented meaningful technical challenges or produced unexpected results. EX1002, ¶137.

B. Claims 1, 11, and 20

Elements 1.7, 11.7, and 20.7 recite “[output/outputting] a risk notification associated with the workload.” As explained above, these elements are obvious in view of Keren, which teaches a system configured to output risk notifications associated with security objects (e.g., workloads) contained within a cloud environment. *Supra*, §X.A.9. The independent claims are further obvious in view of Engelberg’s similar teachings regarding risk notifications. EX1002, ¶138.

Engelberg is directed to “Automated Prioritization of Process-Aware Cyber Risk Mitigation” that involves “generating ... a risk assessment” to “mitigate cyber security risk in enterprise networks.” EX1006, Title, Abstract, ¶6; EX1007; Title, Abstract, ¶5; EX1002, ¶139. Engelberg teaches the straightforward proposition that it is beneficial to notify a system administrator of such a network risk once it

has been identified. EX1006, ¶42 (“[T]he agile security platform provides one or more dashboards, alerts, notifications, and the like to cyber security personnel that enable the cyber security personnel to react to and remediate security relevant events.”); EX1007, ¶40. As one example of such a dashboard, shown below in Fig. 6, Engelberg presents a display conveying various risk notifications such as “Business Process Risk Assessment,” “Crown Jewel Risk Assessment,” “Risk Aspect,” and “Risk Tolerance”:



EX1006, Fig. 6; *see also id.*, ¶¶3 (describing background art systems used to “alert security personnel” of adverse network events via dashboards), 7 (describing

assessing “asset vulnerability” based on “potential for asset breach and criticality of attack paths towards target(s)”), 20 (“FIG. 6 depicts an example UI displaying an example process-aware risk assessment.”), 46 (using dashboards “segmented to owners and relevant IT and security users ... security risk concern can be presented and evaluated in the operations-level”); EX1007, ¶¶2, 6, 18, 44, Fig. 6; EX1002, ¶139.

Thus, consistent with the teachings of Keren, Engelberg teaches that cyber security threats, vulnerabilities, and other network-related risks, once identified, are beneficially presented to a user as risk notifications. As discussed above in Section XI.A, a POSA would have been motivated to report Keren’s security insights to a user as risk notifications in order to alert the user to the security insight for remediation consideration. EX1002, ¶¶140-41.

C. Claim 10

Claim 10 depends from claim 1 and further recites “wherein the risk notification includes one or more resolution recommendations.” The combination of Keren and Engelberg teaches this limitation. EX1002, ¶142.

As explained in Ground 1, Keren teaches a system configured to output risk notifications associated with security objects (e.g., workloads) contained within a cloud environment. *Supra*, §X.A.9. Engelberg teaches analyzing risks associated with assets contained within a network environment, identifying “remediation

recommendations and courses of action (COA),” and conveying the information “to enable users (e.g., security teams) to understand immediate remediation actions corresponding to each issue.” EX1006, ¶¶30, 55, 57, 59; EX1007, ¶¶28, 53, 55, 57. Engelberg therefore teaches including one or more resolution recommendations with a risk notification. EX1002, ¶143. As explained above, a POSA would have had reason to include one or more resolution recommendations with a corresponding risk notification, as taught by Engelberg, as part of outputting generated security insights to a user, as taught by Keren. *Supra*, §XI.A. The combination of Keren and Engelberg thus teaches a risk notification that includes one or more resolution recommendations, as recited. EX1002, ¶¶144-45.

Claim 10 was therefore obvious.

XII. CONCLUSION

The challenged claims are unpatentable. Wiz respectfully requests that IPR be instituted.

Dated: November 1, 2024

Respectfully submitted,

/Matthew A. Argenti/

Matthew A. Argenti, Lead Counsel
Reg. No. 61,836

CERTIFICATION UNDER 37 C.F.R. §42.24(d)

Under the provisions of 37 C.F.R. §42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter Partes* Review totals 13,692 which is less than the 14,000 allowed under 37 C.F.R. 42.24(a)(i). In accordance with 37 C.F.R. 42.24(a), this word count does not include any table of contents, table of authorities, mandatory notices under §42.8, certificate of service or word count, or appendix of exhibits or claim listing.

Dated: November 1, 2024

Respectfully submitted,

/Matthew A. Argenti/

Matthew A. Argenti, Lead Counsel

Reg. No. 61,836

PAYMENT OF FEES UNDER 37 C.F.R. §§42.15(A) AND 42.103

The required fees are submitted herewith. If any additional fees are due at any time during this proceeding, the Office is authorized to charge such fees to Deposit Account No. 23-2415.

LIST OF EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 11,582,257 to Shua
1002	Declaration of Dr. Angelos Stavrou
1003	<i>Curriculum Vitae</i> of Dr. Angelos Stavrou
1004	File History of U.S. Patent Application No. 17/716,491 ('257 Pat.)
1005	U.S. Patent No. 11,374,982 to Keren
1006	U.S. Publication No. 2022/0263855 to Engelberg
1007	U.S. Provisional Application No. 63/150,664 to Engelberg
1008	Collier M. and Shahan R., "Fundamentals of Azure: Microsoft Azure Essentials," 2 nd Edition, Microsoft Press (2016), available at: https://web.archive.org/web/20210413101249/https://download.microsoft.com/download/6/6/2/662DD05E-BAD7-46EF-9431-135F9BAE6332/9781509302963_Microsoft%20Azure%20Essentials%20Fundamentals%20of%20Azure%202nd%20ed%20pdf.pdf , archived April 13, 2021
1009	Amazon AWS, Application Load Balancer, available at: https://web.archive.org/web/20210220224806/https://aws.amazon.com/elasticloadbalancing/application-load-balancer/ , archived February 20, 2021
1010	Kubernetes, Ingress, available at: https://web.archive.org/web/20210110103139/https://kubernetes.io/docs/concepts/services-networking/ingress/ , archived January 10, 2021
1011	freeCodeCamp, What is a Proxy Server? In English, Please., available at: https://web.archive.org/web/20210421111614/https://www.freecodecamp.org/news/what-is-a-proxy-server-in-english-please/ , archived April 21, 2021

Exhibit No.	Description
1012	Fortinet, Proxy Server, available at: https://web.archive.org/web/20210424234713/https://www.fortinet.com/resources/cyberglossary/proxy-server , archived April 24, 2021
1013	Rapid7, Web Application Firewall, available at: https://web.archive.org/web/20210410115503/https://www.rapid7.com/fundamentals/web-application-firewalls/ , archived April 10, 2021
1014	W3schools, Ports and Its Security, available at: https://web.archive.org/web/20210412193421/https://www.w3schools.in/cyber-security/ports-and-its-security/ , archived April 12, 2021
1015	Cloudflare, Inc., What is a computer port? Ports in networking, available at: http://web.archive.org/web/20210303124521/https://www.cloudflare.com/learning/network-layer/what-is-a-computer-port/ , archived March 3, 2021
1016	Nginx, available at: http://web.archive.org/web/20210410094516/https://nginx.org/en/ , archived April 10, 2021
1017	SecurityHQ, Notes from the Field. Using Proxy Logs to Detect Suspicious Activity – Interview with Aaron Hambleton, available at: https://web.archive.org/web/20210416073813/https://www.securityhq.com/blog/using-proxy-logs-to-detect-suspicious-activity/ , archived April 16, 2021
1018	Apache HTTP Server Project, Apache HTTP Server Version 2.4, Log Files, available at: http://web.archive.org/web/20210303195600/https://httpd.apache.org/docs/2.4/logs.html , March 3, 2021
1019	Amazon AWS, Logging web ACL traffic information, available at: http://web.archive.org/web/2020111183928/https://docs.aws.amazon.com/waf/latest/developerguide/logging.html , November 11, 2020
1020	Cichonski, “Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and

Exhibit No.	Description
	Technology,” NIST Special Publication 800-61, Rev. 2 (Aug. 2012) http://dx.doi.org/10.6028/NIST.SP.800-61r2
1021	EP Patent No. 3,360,071 B1 to Li
1022	U.S. Patent No. 9,692,778 B1 to Mohanty
1023	U.S. Publication No. 2014/0245376 A1 to Hibbert
1024	U.S. Publication No. 2023/0086187 to Tan
1025	U.S. Publication No. 2013/0191919 A1, published July 25, 2013, to Basavapatna

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§42.6(e) and 42.105, I certify that I caused to be served true and correct copies of the foregoing: **PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 11,582,257 and Exhibits 1001 - 1025** by Federal Express Next Business Day Delivery on November 1, 2024, on the Patent Owner's correspondence address of record for the subject patent as follows:

Finnegan, Henderson, Farabow,
Garrett & Dunner LLP
Attn: Christopher C. Johns
901 New York Avenue, N.W.
Washington, DC 20001-4413

Dated: November 1, 2024

/Matthew A. Argenti/
Matthew A. Argenti, Lead Counsel
Reg. No. 61,836