WIZ - EXHIBIT 1004 (2)
Wiz, Inc. v. Orca Security Ltd.

[00203] The method of the above disclosed embodiment may further determine network accessibility information 511 by determining the network accessibility information 511 based on the examined data sources. The data examined may include user data, system processing data, accessibility data, clock cycles, storage input/output, and similar processes. A query of the cloud provider API 501 may be automated or manually-initiated. Based on the said query, the network accessibility information 511 related to the workload may change based on the data sources associated with said workload.

[00204] In the method of one disclosed embodiment, network accessibility information 511 may include at least one of data from an external data source, cloud provider information, or at least one network capture log. These embodiments may include data from the operating environment of the cloud-based environment, an external operating system for a computer processing unit, or other similar computer readable media. Network capture logs may be automated or manually updated to include possible vulnerabilities and threats to the cloud-based storage medium.

[00205] A method of the disclosed embodiment may also include identifying the installed software application 507 by extracting data from at least one of OS packages, libraries, or program language libraries. This data may be extracted through a system query, random access algorithm, or similar automated process. Operating system packages may include systems operable on Microsoft, Apple, Linux, and similar operating systems. Libraries may consist of a series of files, folders, and databases of information stored on one of any indexed data repositories. A program language library may contain several of an exemplary program languages including but not limited to Javascript, Swift, Scala, Go, Python, Elm, Ruby, C#, C++ and other similar sources of software code.

[00206] Another method of the disclosed embodiment may identify the installed software application 507 based on the extracted data. The method of the processor 500 may perform only this function or this function among many to accomplish the layered cybersecurity defense described herein this disclosed cloud-based security environment.

[00207] A further disclosed method of the present embodiment may include at least one processor 500 further configured to identify a version 507 of the installed software application 505. The method of the processor may perform only this function or this function among many to accomplish the layered cybersecurity defense described herein this disclosed cloud-based security environment.

[00208] **Prioritization Techniques**

[00209] Fig. 6 is a block diagram of method 600 for risk prioritization, consistent with disclosed embodiments.

[00210] Aspects of this disclosure may include a cloud-based cybersecurity system for assessing internet exposure of a cloud-based workload with embodiments configured to access

at least one cloud provider API to determine a plurality of entities capable of routing and/or filtering traffic in a virtual cloud environment associated with a target account containing the workload. Routing traffic, as used herein, may refer to the process of selecting a path for traffic (e.g., cells, blocks, frames, packets, calls, messages, or other units of data) in a network or between or across multiple networks. The cloud-based cybersecurity system, in some embodiments, may comprise at least one processor configured to assess internet exposure as described herein.

[00211] A cloud provider API, as used herein, may refer to any application program interface that allows the end user to interact with a cloud provider's service. Workloads, as used herein, may refer to devices in a network. By way of example, workloads may include systems, devices, resources in a cloud infrastructure. By way of example, Fig. 1 illustrates examples of a workload, such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, or load balancer 115. In some embodiments, scanning system 101 of Fig. 1 may perform a scanning operation on any workload (e.g., systems, devices, resources, etc.) in cloud infrastructure 106.

[00212] In some embodiments, scanning system 101 of Fig. 1 may use an API to detect workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115 in the source account. In some embodiments, workloads may exist as virtual machines, while in other embodiments, workloads may exist as discrete, physical devices. For example, in some embodiments, a layer 3 routing system could be implemented as a physical device (e.g., a router or switch) or a virtual device (e.g., a routing system instantiated as a virtual device on a computer).

[00213] Scanning system 101 may query devices and systems capable of routing and filtering traffic (e.g., load balancer 115, routers, switches, firewalls, security groups, API Gateways and proxies) using an API provided through a cloud service provider's system to determine network configurations, and may evaluate them against known problematic configurations or other configurations. These devices can be cloud native and / or unmanaged (e.g. NGINX proxy)

[00214] A cloud environment, as used herein, may refer to a platform implemented on, hosted on, and/or accessing servers that are accessed over the Internet. A scanner account, as used herein, may refer to any type of account associated with a scanner. A scanner, as used herein, may refer to a device for examining, reading, or monitoring something. By way of example, Fig. 1 illustrates one example of a scanner, scanning system 101 of Fig. 1. A target account containing the workload, as used herein, may refer to a location on a network server selected as an aim of connection, for example. In some embodiments, scanning system 101 of Fig. 1 may determine entities capable of routing traffic.

[00215] In some embodiments, the plurality of entities includes a virtual network appliance. A virtual network appliance, as used herein, may refer to a machine focused on virtualizing network functionality. In some embodiments, a typical network virtual appliance may include functionality directed to various layer four to seven. That is, such appliances may implement functionality associated with OSI model layers 4-7 (Transport, Session, Presentation, and Application). Such functionality may include, for example, a firewall, WAN optimizer, application delivery controllers, routers, load balancers, IDS/IPS, proxies, or SD-WAN edge device. In some embodiments, scanning system 101 of Fig. 1 may determine a virtual network appliance capable of routing traffic.

[00216] In some embodiments, the virtual network appliance is at least one of a load balancer, a firewall, a proxy, or a router. A load balancer, as used herein, may refer to any device that acts as a reverse proxy and distributes network or application traffic across a number of servers. In some embodiments, load balancer 115 may comprise one or more systems that balance incoming requests between the different systems and devices of cloud infrastructure 106. For example, load balancer 115 may be configured to determine usage (e.g., processor load, used storage capacity) of systems or devices in cloud infrastructure 106 to assist in determining where to route an incoming request from network 105 to store data, perform processing, or retrieve data.

[00217] Load balancer 115 may be configured to receive an incoming request from user device 102. Upon receipt of the request, load balancer 115 may consult a data store (part of or separate from load balancer 115; not pictured) to determine usage or forecasted usage of various systems or devices in cloud infrastructure 106, and may forward the request to the systems or devices having the lowest usage or forecasted usage. A firewall, as used herein, may refer to a device or system that monitors incoming and outgoing network traffic and permits or blocks data packets based on a set of security rules. A proxy, as used herein, may refer to a device or system that translates traffic between networks or protocols. In some embodiments, a proxy may be implemented as an intermediary a device or system separating end-user devices from destinations devices (e.g., web servers) by routing traffic through them. Proxies (or proxy servers) may be configured to provide varying levels of functionality, security, and privacy depending on use case, needs, or company policy. A router, as used herein, may refer to a device or system that connects a local network to the internet.

[00218] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to query the at least one cloud provider API to determine at least one networking configuration of the entities. Querying, as used herein, may refer to a request for data or information from a resource. A cloud provider API, as used herein, may refer to any application program interface that allows an end user to interact with a cloud provider's service. A cloud provider may include any entity through whom information is accessible over

the Internet or other shared network. At least one networking configuration of the entities, as used herein, may refer to a manner in which components are arranged to make up an operating system, networking system, or a computer system. In some embodiments, scanning system 101 of Fig. 1 may send a request to determine at least one networking configuration of the entities.

[00219] In some embodiments, 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.

[00220] A routing configuration, as used herein, may refer to a manner in which components are arranged to make up a routing system. For example, a routing configuration may provide for dynamic or static "routes" between devices on a network or on networks.

[00221] A proxy configuration, as used herein, may refer to a manner in which components are arranged to make up a proxy system. For example, a proxy configuration may provide for a configuration of how a client can route traffic through a central point (e.g., a proxy server), in order to relay multiple clients' requests from a single point on a network.

[00222] A load balancing configuration, as used herein, may refer to a manner in which components are arranged to make up a load balancing system. For example, a loan balancing configuration may define how resources such as uplinks or distributed applications may be utilized by devices on the same (or another) network, in order to balance usage of multiple redundant uplinks or distributed application nodes.

[00223] A firewall configuration, as used herein, may refer to a manner in which components are arranged to make up a firewall system. For example, a firewall configuration may comprise rules that determine what traffic may be forwarded, which devices are permitted to communicate with one another, or what protocols may be used for communication between devices.

[00224] A VPN (virtual private network) configuration, as used herein, may refer to a manner in which components are arranged to make up a VPN system. A VPN may refer to a type of network that extends a private network across a public network and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. For example, a VPN configuration may provide rules or other information by which data is sent or received over a single, encrypted route, such that intermediate points cannot determine the content, source, or destination of the data.

[00225] In some embodiments, scanning system 101 of Fig. 1 may send a request to determine at least a routing configuration, a proxy configuration, a load balancing configuration, a firewall configuration, or a VPN configuration of the entities.

[00226] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to build a graph connecting the plurality of entities based on the networking configuration. A graph connecting the plurality of entities, as used herein, may

refer to a diagram showing the relation between variable quantities (e.g., entities). In some embodiments, the graph may include any data structures that includes the connections from one entity to another. In some embodiments, scanning system 101 of Fig. 1 may construct a graph or diagram linking the plurality of entities based on the networking configuration. In some embodiments, graph includes a data structure sequentially connecting entities. 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.

[00227] In some embodiments, a graph may include directional vectors indicating directions of dataflow. Directional vectors, as used herein, may refer to a quantity having direction as well as magnitude (indicating, for example, what kinds of traffic are allowed to flow, what ports are open, what protocols are allowed), especially as determining the position of one point in space relative to another. Directions of dataflow, as used herein, may refer to the movement of data through a system comprised of software, hardware or a combination of both along a particular course. In some embodiments, scanning system 101 of Fig. 1 may construct a graph including a data structure connecting entities sequentially or a graph including directional vectors showing the direction of data flow.

[00228] In some embodiments, to build the graph, the at least one processor may be configured to identify individual entities as nodes and connect the nodes. Nodes, as used herein, may refer to a device in a network (e.g., devices in cloud infrastructure 100). Connecting nodes, as used herein, may refer to linking or bonding devices or data points in a larger network. In some embodiments, the graph includes port numbers associated with the workload. Port numbers, as used herein, may refer to a way to 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. In some embodiments, the graph includes a path from the at least one source to the workload. A path from the at least one source to the workload, as used herein, may refer to a string of characters used to uniquely identify a location in a directory structure.

[00229] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to access a data structure identifying services publicly accessible via the Internet and capable of serving as an internet proxy. Services capable of serving as an internet proxy, as used herein, may refer to a system or router that provides a gateway between users and the internet (e.g., network 105). In some embodiments, scanning system 101 of Fig. 1 may access a configuration detecting services over the internet and capable of serving as an internet proxy.

[00230] Aspects of this disclosure may include embodiments configured to integrate the identified services into the graph. Integrating identified services into a graph, as used herein, may refer to the act of bringing together smaller components (the identified services) into a

single system that functions as one (the graph). For example, scanning system 101 may integrate information on services available in each node into the data structure, associated with each node in the data structure.

[00231] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to traverse the graph to identify at least one source originating via the Internet and reaching the workload. Traversing the graph, as used herein, may refer to a device, such as scanning system 101, accessing the systems represented by the nodes in the graph, to determine which systems are accessible and how. For example, scanning system 101 may analyze the graph to detect sources coming from the internet and reaching the workload, by accessing a first system, and traversing between systems in a depth-first, breadth-first, or other manner, until a path to an external network (e.g., the Internet) is reached.

[00232] In some embodiments, the at least one source may be a potential malicious source. In some embodiments, the source may include a code in any part of a software system or script that is intended to cause undesired effects, security breaches or damage to a system. (In some embodiments, the Internet at large may be considered to be a potentially malicious source.)

[00233] Aspects of this disclosure may include embodiments configured to output a risk notification associated with the workload. Outputting a risk notification, as used herein, may refer to sending out data such as a message indicating a possibility of danger. In some embodiments, scanning system 101 of Fig. 1 may produce a notification indicating a risk that is related with the workload. Examples of such risk notifications may include, for example, an electronic mail message, a visual alert, an audio alert, an audio-visual alert,

[00234] In some embodiments, the risk notification includes one or more resolution recommendations. Resolution recommendations, as used herein, may refer to a suggestion or proposal as to the best course of action in view of the identified risk.

[00235] **Fig. 6** illustrates a block diagram of method 600 performed by a processor (e.g., a processor associated with scanning system 101), consistent with disclosed embodiments. In some embodiments, the processor may operate on instructions stored in a non-transitory computer readable medium. In some embodiments, the method may include seven (or more or less) steps:

[00236] Block 602: 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. In some embodiments, scanning system 101 of Fig. 1 may determine entities capable of routing traffic, consistent with the above-disclosed embodiments.

[00237] Block 604: Query the at least one cloud provider API to determine at least one networking configuration of the entities. In some embodiments, scanning system 101 of Fig. 1

may send a request to determine at least one networking configuration of the entities, consistent with the above-disclosed embodiments.

[00238] Block 606: Build a graph connecting the plurality of entities based on the networking configuration. In some embodiments, scanning system 101 of Fig. 1 may construct a graph or diagram linking the plurality of entities based on the networking configuration, consistent with the above-disclosed embodiments.

[00239] Block 608: Access a data structure identifying services publicly accessible via the Internet and capable of serving as an internet proxy. In some embodiments, scanning system 101 of Fig. 1 may access a configuration detecting services over the internet and capable of serving as an internet proxy, consistent with the above-disclosed embodiments.

[00240] Block 610: Integrate the identified services into the graph. In some embodiments, scanning system 101 of Fig. 1 may incorporate the services into the graph, consistent with the above-disclosed embodiments.

[00241] Block 612: Traverse the graph to identify at least one source originating via the Internet and reaching the workload. In some embodiments, scanning system 101 of Fig. 1 may analyze the graph to detect sources coming from the internet and reaching the workload, consistent with the above-disclosed embodiments.

[00242] Block 614: Output a risk notification associated with the workload. In some embodiments, scanning system 101 of Fig. 1 may produce a notification indicating a risk that is related with the workload, consistent with the above-disclosed embodiments.

[00243] **Techniques for Malware Detection Using Secondary System**

[00244] Aspects of this disclosure may provide a technical solution to the challenging technical problem of malware detection on a primary system using a secondary system other than the primary system. In existing technologies, malware detection system may be run on a primary system that itself may be susceptible to malware. The malware may detect existence of the malware detection system and deceive it to avoid being detected. To resolve such a technical problem, a secondary system may be provided to host the malware detection system, which may isolate the malware detection system from potential exposure or influence from the malware that may reside on the primary system. The disclosed technical solutions may increase success rate of malware detection and thus enhance security level of the primary system.

[00245] A cyber security system, as used herein, may refer to a system including any combination of software and hardware for enhancing security of a device, a platform, or another system in a network environment. By way of example, the cyber security may be implemented as a system including scanning system 101 in Fig. 1. As another example, scanning system 101 may include the disclosed cyber security system as a subset. A cloud environment, as used herein, may refer to a computing environment running on a cloud. By way of example, the cloud environment may include or be part of cloud infrastructure 106 in Fig. 1.

[00246] Consistent with disclosed embodiments, at least one processor may be configured to utilize a cloud provider API to access a block storage volume of a workload maintained on a target account in a target system of a cloud storage environment. Utilizing, as used herein, may refer to an operation of using, deploying, enacting, enabling, activating, allocating, invoking, calling, or any operation of putting a thing into use in a computer environment. An API refers to an application programming interface herein. A cloud provider in this disclosure may refer to a cloud service provider (e.g., a service provider of cloud computing, cloud storage, or any services provided to users on demand over a network). A cloud provider API, as used herein, may refer to an API provided, prepared, enabled, activated, programmed, or written by a cloud provider. A block storage volume in this disclosure may refer to a storage volume in a block storage scheme. In some embodiments, a block storage volume may be connected, disconnected, or reconnected to a system (e.g., a physical computer, a virtual machine, a network interface, a cloud, or any combination of hardware or software modules) without interfering (e.g., shutting down, halting, rebooting, or any manner of interrupting) the operation status of the system or other running tasks on the system. For example, a block storage volume may be implemented as a virtual disk. In some embodiments, a block storage volume can be associated with an account of a user of the system, the system that it is connected to, or both.

[00247] Block storage, as used herein, may refer to a data storage scheme or model in which data is saved to storage media in fixed-sized raw data chunks (referred to as “blocks”). The raw data blocks may contain no metadata. Each block may be associated with a unique address as metadata assigned to the block. Storage blocks may be controlled by an operating system (OS) and may be accessed by a protocol (e.g., iSCSI, Fibre Channel, or Fibre Channel over Ethernet).

[00248] A storage volume, as used herein, may refer to an identifiable unit (e.g., a physical unit or a logical, virtual unit) of data storage. The storage volume may be mounted to a device via an operation system and be configured with a specific file system (e.g., New Technology File System) assigned a system-unique name or number that identifies the storage volume. The storage volume may represent a named, logical area of storage that enables users and applications to access data on the underlying device. By way of example, a storage volume may be a logical disk that represents a named, logical area of a physical storage device (e.g., a hard disk drive, solid-state drive, compact disc read-only memory, digital video disk, floppy disk, or any other type of storage device). In some embodiments, a logical storage volume may span multiple physical storage devices (e.g., hard disks) and appear as a single, contiguous storage area that works like a physical volume. Storage volumes may be flexibly configured, such as being expanded, contracted, mirrored, stripped, or adapted to support multiple disks (e.g., redundant array of independent disks).

[00249] 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. By way of example, a workload may be a virtual machine, a database, a container, a Hadoop node, an application, a storage object, a load balancer, or an IAM (Identity and Access Management) configuration. A cloud storage environment in this disclosure may refer to a computing environment of a cloud computing model (referred to as "cloud storage") that stores data on the Internet through a cloud computing provider that manages and operates data storage as a service. A cloud storage environment may be managed and controlled by an operating system. The operation system may be susceptible to malware and security vulnerabilities, and may become a primary system (or referred to as a "target system" herein) that needs malware detection. By way of example, the cloud storage environment may include or be part of cloud infrastructure 106 in Fig. 1.

[00250] A target system of a cloud storage environment in this disclosure may refer to an operating system or a computing system with an installed operating system, which manages and controls the cloud storage environment or a subsystem of the target system. The target system may maintain one or more accounts for users of the cloud storage environment. 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).

[00251] By way of example, to utilize the cloud provider API to access the block storage volume (e.g., existing in storage 111A-111D in Fig. 1) of the workload, the at least one processor (e.g., a processor in scanning system 101 of Fig. 1) may communicate with the cloud storage environment (e.g., cloud infrastructure 106 in Fig. 1) via sending and receiving data (e.g., data packets) over a network (e.g., network 105 in Fig. 1), and enable one or more APIs provided by the cloud provider for accessing the block storage volume. For example, the communication and API enabling may be performed through the process of integration as described in step 201 of Fig. 2. In some embodiments, the at least one processor may create a connection (e.g., by logging in) between a first account on scanning system 101 and a second account on cloud infrastructure 106, then cause the second account to authorize access privileges to the first account via the cloud provider API for accessing storage 111A-111D in cloud infrastructure 106.

[00252] Consistent with disclosed embodiments, the at least one processor may be configured to utilize a scanner at a location of the block storage volume and on a secondary system other than the target system. A scanner, as used herein, may refer to an application, a program, a service, a process, a thread, a function, or any executable codes for performing a scanning process on a computer system to obtain information (e.g., related to structures, vulnerabilities, security issues, or any information of the computer system).

[00253] A location of the block storage volume in this disclosure may refer to a system (e.g., a physical computer, a virtual machine, a network interface, a cloud, a data center, or any combination of hardware or software modules) or an identifier associated with the system that the block storage volume is connected to. In some embodiments, the location of the block storage volume may include at least one of the target account, a secondary system account, a cloud provider account, or a third party account. By way of example, the secondary system account may be an account connected with the target account (e.g., connected by an integration process similar to step 201 in Fig. 2A). The cloud provider account may be an account connected to at least one of the target account or the secondary system account. The third party account may be an account hold by a third party and connected to at least one of the target account, the secondary system account, or the cloud provider account.

[00254] In some embodiments, the at least one processor may determine the location of the block storage volume based on at least one of the target account, a secondary system account, a cloud provider account, or a third party account. By way of example, to determine the location of the block storage volume, the at least one processor may consult a lookup table, a list, or a database that stores a relationship record between an account (e.g., a target account, a secondary system account, a cloud provider account, or a third party account) and a system (e.g., a physical computer, a virtual machine, a network interface, a cloud, or any combination of hardware or software modules) that the block storage volume is connected to. Based on the relationship record, the at least one processor may identify the system as the location using the account as a key.

[00255] A secondary system, as used herein, may refer to a computer system running at an environment or conditions not affected by another system (referred to as a “primary system” or “target system” herein). In some embodiments, the secondary system may have an operating system different from an operating system of the target account. By way of example, the target account may have a WINDOWS® operating system, and the secondary system may have a LINUX® operating system.

[00256] In some embodiments, the secondary system includes at least one of a virtual machine (e.g., any of virtual machines 107A-107D in Fig. 1), a container, or a serverless function. A container, as used herein, may refer to an OS-level virtualization instance assigned with resources by a non-virtual computer operating system (OS). A non-virtual computer operating system (OS) may create multiple isolated user space instances that may function like real computers from the point of view of programs running in them. A computer program running on a non-virtual operating system can access all available resources (e.g., connected devices, files, folders, network shares, CPU power, or any other software or hardware capabilities) of the computer. In contrast, a computer program running inside of a container can only access the resources assigned to the container.

[00257] A serverless function, as used herein, may refer to a computer function hosted in a cloud environment that may allocate resources on demand to perform a function (e.g., scanning) for a user of the cloud environment. The user of the serverless function may be free from concerns of capacity planning, configuration, management, maintenance, fault tolerance, or scaling of containers, virtual machines, or physical servers. When a serverless function is not running, no computing resources may be allocated to it (e.g., using no provisioned server, thus named "serverless"). By way of example, a serverless function may be implemented as any combination of Lambda functions, event sources, and other computing resources. A serverless function may be invoked by the user of a cloud infrastructure (e.g., a cloud environment or a cloud environment).

[00258] By way of example, to utilize the scanner at the location of the block storage volume (e.g., existing in storage 111A-111D in Fig. 1), the at least one processor (e.g., a processor in scanning system 101 of Fig. 1) may communicate with the secondary system (e.g., one or more of virtual machines 107A-107D in Fig. 1) other than the target system (e.g., the main system operating and managing storage 111A-111D) and activate the secondary system for scanning. For example, to activate the secondary system, the at least one processor may transmit a scanner (e.g., in the form of a computer program or executable codes) to the secondary system, and the secondary system may allocate computing resources (e.g., CPU powers, memory space, network ports, or any other software or hardware resources) in preparation of running the scanner. The scanner may be programmed to be stored at the location of the block storage volume or to start the scanning at the location of the block storage volume. For example, the block storage volume may be connected to a particular system in cloud infrastructure 106 in Fig. 1, and the scanner may be stored at that system (e.g., on storage connected to the system). When the scanner begins scanning, the scanner may start the scanning at that system before scanning other systems.

[00259] In some embodiments, to utilize the scanner, the at least one processor may suspend an operation of the scanner after the scan of the block storage volume. Suspending, as used herein, may refer to an operation of pausing, stopping, ceasing, freezing, halting, hanging, terminating, killing, or any manner or temporarily or permanently setting aside an ongoing process or computer program. By way of example, after the scanner completes scanning the block storage volume, the at least one processor may detect the completion by an active manner (e.g., by periodically polling the secondary system for a status of the scanning) or a passive manner (e.g., by receiving a message from the secondary system indicating the completion of the scanning). The at least one process may then send data to the secondary system, the data including a signal or computer-executable codes for suspending the operation of the scanner. After receiving the data, the secondary system may suspend the scanner and release the computing resources previously allocated to the scanner.

[00260] In some embodiments, to utilize the scanner, the at least one processor may modify a pre-utilized scanner at the location of the block storage volume based on information related to the target account to obtain a modified scanner, and then utilize the modified scanner. A pre-utilized scanner in this disclosure may refer to a scanner that has been previously utilized (e.g., on the same secondary system) and not destroyed. In some embodiments, the information related to the target account may include data or parameters indicating a status or a change of the target account, such as, for example, newly created files, deleted files, changed files, or any data indicating a state of the target account. For example, the pre-utilized scanner at the block storage volume may use a configuration file. The configuration file may include data or parameters indicating a previous status (e.g., list of existing files, public network ports, Internet-accessible data channels, or any operation state parameters) of the target account. The information related to the target account may include data indicating a current status of the target system. The current status of the target system may be different from the previous status of the target system (e.g., including changed files, changed public network ports, changed Internet-accessible data channels, or any changed operation state parameters). The at least one processor may modify the pre-utilized scanner by modifying the configuration file to reflect the current status of the target system. Then, the at least one processor may utilize the modified scanner.

[00261] Consistent with disclosed embodiments, the at least one processor may be configured to scan the block storage volume for malicious code, using the secondary system. Scanning, as used herein, may refer to an operation of traversing, checking, inspecting, reading, examining, searching, or any manner of looking up. Malicious code, as used herein, may refer to an application, a computer program, a service, a process, a thread, a function, a script, or any type of executable codes that may be executed on a computer without authorization and with a malicious or harmful intent (e.g., manipulating the computer into dangerous behaviors, writing changes or add-ons to existing computer programs or files, stealing computer files, copying sensitive information without authorization, damaging the computer itself, disabling one or more functions of the computer, or attacking another computer using the computer as a disguise). For example, malicious code may include attack scripts, computer viruses, computer worms, Trojan horses, backdoors, or any other malicious executable content. In some embodiments, the malicious code may include a rootkit. A rootkit in this disclosure may refer to any type of computer software designed to enable access to a computer or an area of its software without authorization.

[00262] By way of example, to scan the block storage volume (e.g., existing in storage 111A-111D in Fig. 1) for malicious code, the secondary system (e.g., one or more of virtual machines 107A-107D in Fig. 1) may run the utilized scanner with allocated computing resources to look up databases (e.g., a blacklist) of known malicious files, look up databases storing

patterns of malicious files and identify the patterns, look up databases of cryptographic hashes that identify malicious files, run potential malicious code in a sandbox (e.g., a confined and isolated environment for running programs without causing changes outside the confined and isolated environment), determine whether there are scanned files that are not in a database (e.g., a whitelist) of known good files, detect changes over time in the scanned environment, or any combination thereof.

[00263] In some embodiments, scanning the block storage volume may include scanning disk-backed memory. A disk-backed memory in this disclosure may refer to a storage space implemented on a disk (e.g., a hard disk) and functioning like a volatile memory from the perspective of a running program. By way of example, scanning the block storage volume may include scanning files stored in the disk-backed memory. In some embodiments, the disk-backed memory may include at least one of a page file or a cache file. A page file, as used herein, may refer to a file containing one or more pages (also referred to as "memory pages" or "virtual pages"), each of the one or more pages being a fixed-length contiguous block of virtual memory and described by a single entry in a page table. A page file may be stored on a disk (e.g., a hard disk) for memory management of a virtual memory used by an operating system. A cache file, as used herein, may refer to a file storing data so that future requests for the stored data can be served faster. The data stored in a cache file may be the result of a computation or a copy of the data stored elsewhere.

[00264] Consistent with disclosed embodiments, the at least one processor may be configured to identify malicious code based on the scan. By way of example, if the secondary system scans the block storage volume by consulting databases of known malicious files and finds a scanned file matches a record in the databases, the at least one processor may identify the matched scanned file as the malicious code. In another example, if the secondary system scans the block storage volume by consulting databases storing patterns of malicious files and identifies a pattern of a scanned file matches one or more of the stored patterns, the at least one processor may identify the scanned file as the malicious code. In yet another example, if the secondary system scan the block storage volume by consulting databases of cryptographic hashes that identify malicious files and finds that a cryptographic hash of a scanned file matches one or more of the cryptographic hashes that identify malicious files, the at least one processor may identify the scanned file as the malicious code.

[00265] In yet another example, if the secondary system scans the block storage volume by running potential malicious code in a sandbox and finds that running a scanned file causes malicious or harmful changes or behavior (e.g., an attempt to modify a system setting, an attempt to access files outside of a set of directories, an attempt to perform a network mapping process), the at least one processor may identify the scanned file as the malicious code. In yet another example, if the secondary system scans the block storage volume to determine whether

there are scanned files that are not in a database of known good files and finds that a scanned file is not in the database of known good files, the at least one processor may identify the scanned file as the malicious code. As yet another example, if the secondary system scans the block storage volume to detect changes over time in the scanned environment and finding that a scanned file causes a malicious or harmful change in the scanned environment over time, the at least one processor may identify the scanned file as the malicious code. It should be noted that the identification of the malicious code based on the scan may be implemented in various manners and is not limited to the examples described herein.

[00266] Consistent with disclosed embodiments, the at least one processor may be configured to output from the secondary system, a notification of a presence of malicious code in the target system. In some embodiments, the notification may be a pop-up window, a pop-pup dialog, a flashing symbol, an email, a text message, a mobile application notice, a phone call, a warning sound, a tactile feedback (e.g., a vibration), or any type of communications. By way of example, the secondary system may trigger a communication channel between the secondary system and the target system and feed one or more parameters to the communication channel, in which the one or more parameters represent the presence of the malicious code. The secondary system may then send data representing the notification via the triggered communication channel including the one or more parameters to the target system. After receiving the data, the target system may parse the data and identify the one or more parameters. The target system may then display the notification (e.g., as a pop-up window on a screen) that present the presence of the malicious code.

[00267] Consistent with disclosed embodiments, the at least one processor may be further configured to identify a source of the malicious code. The notification of the presence of malicious code outputted from the secondary system in the target system may include information related to the identified source of the malicious code. In some embodiments, the source of the malicious code may include identity of a developer or distributor of the malicious code, the landing location (e.g., a network port or a vulnerable access point) where the malicious code is imported to the target system, source codes of the malicious code, or any information or data indicating the origination of the malicious code. The information related to the identified source of the malicious code may include a text, a picture, a symbol, a video, a link, a voice, or any type of data for presenting and describing the identified source of the malicious code.

[00268] By way of example, **Fig. 7** is a block diagram illustrating an exemplary process 300 of cyber security scanning for a cloud environment, consistent with the disclosed embodiments. While the block diagram may be described below in connection with certain implementation embodiments presented in other figures, those implementations are provided for illustrative purposes only, and are not intended to serve as a limitation on the block diagram.

For example, the steps in Fig. 7 may be executed in any order, steps may be duplicated, or steps may be omitted. In some embodiments, the process 300 may be performed by at least one processor (e.g., a CPU) of a computing device or system (e.g., scanning system 101 in Fig. 1) to perform operations or functions described herein, and may be described hereinafter with reference to Figs. 1-2D by way of example. In some embodiments, some aspects of the process 300 may be implemented as software (e.g., program codes or instructions) that are stored in a memory or a non-transitory computer-readable medium. In some embodiments, some aspects of the process 300 may be implemented as hardware (e.g., a specific-purpose circuit). In some embodiments, the process 700 may be implemented as a combination of software and hardware.

[00269] Fig. 7 includes process blocks 702-710. At block 702, at least one processor may utilize a cloud provider API to access a block storage volume of a workload maintained on a target account in a target system of a cloud storage environment.

[00270] At block 704, the at least one processor may utilize a scanner at a location of the block storage volume and on a secondary system other than the target system. In some embodiments, the secondary system may include at least one of a virtual machine, a container, or a serverless function. In some embodiments, the secondary system may have an operating system different from an operating system of the target account.

[00271] In some embodiments, to utilize the scanner, the at least one processor may suspend an operation of the scanner after the scan of the block storage volume. In some embodiments, to utilize the scanner, the at least one processor may modify a pre-utilized scanner at the location of the block storage volume based on information related to the target account to obtain a modified scanner, and the utilize the modified scanner.

[00272] At block 706, the at least one processor may scan the block storage volume for malicious code, using the secondary system. In some embodiments, to scan the block storage volume, the at least one processor may scan disk-backed memory. For example, the disk-backed memory includes at least one of a page file or a cache file. In some embodiments, the malicious code includes a rootkit.

[00273] At block 708, the at least one processor may identify malicious code based on the scan. At block 710, the at least one processor may output from the secondary system, a notification of a presence of malicious code in the target system.

[00274] Consistent with disclosed embodiments, besides blocks 702-710, the at least one processor may further determine the location of the block storage volume based on at least one of the target account, a secondary system account, a cloud provider account, or a third party account.

[00275] Consistent with disclosed embodiments, besides blocks 302-310, the at least one processor may further identify a source of the malicious code. The notification outputted in block 710 may include information related to the identified source of the malicious code.

[00276] **Forward and Rearward Facing Attack Vector Visualization Techniques**

[00277] **Fig. 8** is a schematic block diagram illustrating an exemplary embodiment of a system for performing visualization of forward and backward facing threats.

[00278] In one of many embodiments, the system may include a graphical user interface system for providing forward and backward facing attack vector visualizations. A graphical user interface system may include systems similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows 10 Mobile, PalmOS-Web OS, or Firefox OS. Forward facing attack vectors may be those external entities that may access an analyzed asset. Backward facing attack vectors may include the possible impacts to an analyzed asset and those impacts' possible effect. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of attack vectors.

[00279] In an embodiment, the system may comprise at least one processor configured to identify assets 801 in a cloud environment. The at least one processor may include a processor such as user device 102 of Fig. 1. The at least one processor may include a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service.

[00280] Another embodiment contemplated may consist of at least one processor configured to identify risks 803 associated with each of the identified asset. The identified risks 803 associated with each of the identified assets may be listed sequentially based on probability of risk, severity of risk, or similar method of ordering the risks. The identified risks 303 associated with an identified asset may be scored based on the level of risk to the asset and may be categorized as a high risk, medium risk, or low risk. Once the level of risk can be determined by scanning system 101, the level of risk can be conveyed and displayed to a user device 102 with a representative graphical display.

[00281] An exemplar embodiment may additionally comprise at least one processor configured to identify relationships 805 between at least some of the identified assets, the relationships including at least one of a trust 807A, a network connectivity 807B, or a mechanism of network proxying 807C. A trust 807A may establish a relationship and may

involve cryptography, digital signatures, electronic certificates, or similar method of establishing a trusting relationship. A network connectivity 807B may include wireless or wired connections and may be established once a relationship of trust 807A is established. A mechanism of network proxying 807C can be accommodated once trust 807A and network connectivity 807B are established through any number of connected servers to accomplish proxying.

[00282] A further embodiment may also comprise a processor configured to receive an identification 809 of a specific asset under investigation. The specific asset under investigation may be a physical or virtual machine that has software operating a specific version that is vulnerable to any number of associated risks. The identification 809 of a specific asset under investigation may involve an IP address, universally unique identifier, or similar system to uniquely or semi-uniquely identify the specific asset under investigation.

[00283] A system among several embodiments may include a processor configured to perform a forward analysis 811A of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment. A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. The analysis performed by scanning system 101 may determine the origin of the forward threat, the nature of the threat to the analyzed system, and the seriousness of the threat originating from the Internet. For example, scanning system 101 may identify at least one possible attack vector to the workload currently being analyzed. The network outside of the cloud environment may identify new or existing threats and attack vectors currently threatening the associated system and workload.

[00284] An exemplary embodiment may further comprise a processor configured to perform a backward analysis 811B of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset. A backward analysis 811B of the associated machine may look downstream to related workloads and systems connected to the analyzed systems. A scan downstream may involve, among several embodiments, the downstream exposure risk including an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset. The at least one exposure risk may include one or more entry point to access the exposed asset and any related machines potentially vulnerable to attack from the analyzed and exposed asset.

[00285] Another embodiment may comprise a processor configured to output a signal 813 to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. The display of a forward or

backward path to a specific asset may be one of several visual displays such as linear, graphical, or through computer-generated images. The visualization of a plurality of entry points and lateral movement risks with the plurality of entry points may include visualization of the plurality of paths may sequence the paths based on likelihood of access via that path or the severity of potential threat in a path of access to an asset.

[00286] An example embodiment may comprise a system wherein the network outside the cloud environment includes the Internet. The network outside the cloud environment may include access to the Internet in order for the system to access a plurality of analyzed and potentially vulnerable assets.

[00287] Another embodiment may comprise a system wherein the assets in the cloud environment include at least one of a virtual machine, a network appliance, a storage appliance, a compute instance, or an engine instance. A virtual machine may include an emulated version of a computer - including an operating system, memory, storage, graphics processing - such that it can be indistinguishable from a standard (non-virtual) machine to a running program. A network appliance may include a plurality of servers connected through the Internet in order to establish a system and network of stored assets and workloads. A storage appliance may be comprised by a server designed to store workloads and assets to be analyzed. A compute instance may involve input and processing of data entries that cause the analyzed asset to be accessed and run. An engine instance may include a search algorithm run to analyze and search the analyzed workloads or virtual machines to determine the software and data stored on the said virtual machine or stored instance.

[00288] One among many embodiments may comprise a system wherein identifying the assets in a cloud environment includes identifying the assets based on at least one of an identity and access management policy, an organization policy, or an access policy. An identity and access management policy may comprise a software tool or similar method of establishing identity of users attempting to access a system to verify identities of users and control access to an associated workload or system functionality. An organization policy may include a policy applied to all users by the controlling organization of a workload. An access policy may determine the authorized users of an associated system.

[00289] One possible embodiment may comprise a system wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream asset or a downstream asset. The indication will be via a visualization showing specific alternative paths connecting specific assets or it may visualize a plurality of alternative paths connecting specific assets. Alternative paths connecting between the specific asset and an upstream asset or a downstream asset may involve a direct connection, such as via the Internet, or an indirect connection via one or more series of servers acting as proxy servers between the analyzed asset and the upstream or downstream asset.

[00290] An exemplary embodiment may further comprise a system wherein the visualization includes a presentation of the alternative paths. The alternative paths may be presented visually through a graphical representation, sequential list of alternative paths, or a computer-generated image that represents the plurality of alternative paths.

[00291] Another embodiment may comprise a system wherein the presentation of the forward and backward paths indicates port numbers for each pathway. The port numbers may be indicated by a series of alphanumeric characters (e.g., “HTTP” or “80”) or other identifier.

[00292] One among several exemplary embodiments may further comprise a system wherein the visualization of the entry points indicates at least one entry point of risk. The visualization may indicate one or more points of entry to a given analyzed system. An entry point of risk may involve a port of access to the workload, and may involve a one or more entry points visualized for use by user device 102.

[00293] Another exemplary embodiment may comprise a system wherein the at least one processor is further configured to monitor network activities of the assets in a cloud environment. The processor may monitor continuously, intermittently, or as directed by an authorized user to direct monitoring of the analyzed network activities of the assets. The cloud environment may be, among several embodiments, the cloud infrastructure 106.

[00294] An embodiment contemplated may comprise a system further configured to detect a potential risk associated with the specific asset based on a network activity of the specific asset. The potential risk may be associated to a specific asset based on historical threats to the specific asset or a list of possible threats to the specific asset based on its current network activities.

[00295] Another embodiment may comprise a system further configured to detect a potential risk associated with the specific asset based on a network activity of an upstream asset of the specific asset. The potential risk associated with the specific asset may be identified by scanning system 101 and determine whether a potential risk can be classified as high risk, medium risk, or low risk. The potential risk may be identified based on the specific asset, the network activity performed by the specific asset, or the specific asset type or related software of the upstream asset.

[00296] One of many embodiments may comprise a system further configured to detect a potential risk associated with the specific asset based on a network activity of a downstream asset of the specific asset. The potential risk may be identified based on the specific asset, the network activity performed by the specific asset, or the specific asset type or related software of the upstream asset. For example, scanning system 101 may perform a scan on an asset being analyzed, and based on the one or more entry points of risk, may determine if the downstream asset of the specific asset poses a potential risk to the analyzed system.

[00297] Another exemplary embodiment may comprise a method for generating a graphical user interface for providing forward and backward facing attack vector visualizations. The method may include a method of generating a graphical user interface similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows 10 Mobile, PalmOS-Web OS, or Firefox OS. A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. Backward analysis 811B may include the possible impacts to an analyzed asset and those impacts' possible effect. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of attack vectors.

[00298] One exemplar embodiment may further comprise a method of identifying assets 801 in a cloud environment. A method may include at least one processor that may include a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service.

[00299] One of several embodiments may comprise a method of identifying risks 803 associated with each of the identified assets 801. The method of identifying risks 803 associated with each of the identified assets 801 may be listed sequentially based on probability of risk, severity of risk, or similar method of ordering the risks. The identified risks 803 associated with an identified asset may be scored based on the level of risk to the asset and may be categorized as a high risk, medium risk, or low risk. Once the level of risk can be determined by scanning system 101, the level of risk can be conveyed and displayed to a user device 102 with a representative graphical display.

[00300] Another embodiment contemplated may comprise a method for identifying relationships 305 between at least some of the identified assets 801, the relationships including at least one of a trust 807A, a network connectivity 807B, or a mechanism of network proxying 807C. A trust 807A may establish a relationship and may involve cryptography, digital signatures, electronic certificates, or similar method of establishing a trusting relationship. A network connectivity 807B may include wireless or wired connections and may be established once a relationship of trust 807A is established. A mechanism of network proxying 807C can be accommodated once trust 807A and network connectivity 807B are established through any number of connected servers to accomplish proxying.

[00301] An embodiment may further comprise a method of receiving an identification of a specific asset 809 under investigation. The specific asset under investigation may be a physical

or virtual machine that has software operating a specific version that is vulnerable to any number of associated risks 803. The identification 809 of a specific asset under investigation may involve an IP address, universally unique identifier, or similar system to uniquely or semi-uniquely identify the specific asset under investigation.

[00302] An embodiment may also comprise a method of performing a forward analysis 811A of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment. A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. A forward analysis may involve analysis of the specific asset under identification to identify at least one possible Internet-originating attack vector to the asset. The analysis performed by scanning system 101 may determine the origin of the forward threat, the nature of the threat to the analyzed system, and the seriousness of the threat originating from the Internet. For example, scanning system 101 may identify at least one possible attack vector to the workload currently being analyzed. The network outside of the cloud environment may identify new or existing threats and attack vectors currently threatening the associated system and workload.

[00303] One embodiment may comprise a method of performing a backward analysis 811B of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset. Backward analysis 811B may include the possible impacts to an analyzed asset and those impacts' possible effect. A scan downstream may involve, among several embodiments, the downstream exposure risk including an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset. The at least one exposure risk may include one or more entry point to access the exposed asset and any related machines potentially vulnerable to attack from the analyzed and exposed asset.

[00304] An embodiment may comprise a method of outputting a signal 813 to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. The display of a forward or backward path to a specific asset may be one of several visual displays such as linear, graphical, or through computer-generated images. The visualization of a plurality of entry points and lateral movement risks with the plurality of entry points may include visualization of the plurality of paths may sequence the paths based on likelihood of access via that path or the severity of potential threat in a path of access to an asset.

[00305] Another embodiment may comprise a method wherein the network outside the cloud environment includes the Internet. The network outside the cloud environment may include access to the Internet in order for the system to access a plurality of analyzed and potentially vulnerable assets.

[00306] One of several embodiments may comprise a method wherein the assets in the cloud environment include at least one of a virtual machine, a network appliance, a storage appliance, a compute instance, or an engine instance. A virtual machine may include an emulated version of a computer - including an operating system, memory, storage, graphics processing - such that it can be indistinguishable from a standard (non-virtual) machine to a running program. A network appliance may include a plurality of servers connected through the Internet in order to establish a system and network of stored assets and workloads. A storage appliance may be comprised by a server designed to store workloads and assets to be analyzed. A compute instance may involve input and processing of data entries that cause the analyzed asset to be accessed and run. An engine instance may include a search algorithm run to analyze and search the analyzed workloads or virtual machines to determine the software and data stored on the said virtual machine or stored instance.

[00307] An embodiment may further comprise a method wherein identifying assets in a cloud environment includes identifying the assets based on at least one of an identity and access management policy, an organization policy, or an access policy. An identity and access management policy may comprise a software tool to verify identities of users and control access to an associated workload or system functionality. An organization policy may include a policy applied to all users by the controlling organization of a workload. An access policy may determine the authorized users of an associated system.

[00308] An embodiment may also comprise a method wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream or a downstream asset. The indication may be via a visualization showing specific alternative paths connecting specific assets or it may visualize a plurality of alternative paths connecting specific assets.

[00309] Another exemplar embodiment may comprise a method wherein the visualization includes a presentation of the alternative paths. The alternative paths may be presented visually through a graphical representation, sequential list of alternative paths, or a computer-generated image that represents the plurality of alternative paths.

[00310] Another embodiment may comprise 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.

[00311] One embodiment may comprise a non-transitory computer-readable medium storing instructions that, when executed by at least one processor 800, are configured to

perform operations identifying assets 801 in a cloud environment. the instructions stored on a non-transitory computer-readable medium may include at least one processor 800 that may include a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service.

[00312] An embodiment may further comprise 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 300 to perform operations comprising identifying risks 803 associated with each of the identified asset. The instructions performed of identifying risks 803 associated with each of the identified assets 301 may be listed sequentially based on probability of risk, severity of risk, or similar method of ordering the risks.

[00313] Another embodiment may comprise 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 identifying relationships 805 between at least some of the identified assets 301, the relationships including at least one of a trust 807A, network connectivity 807B, or a mechanism of network proxying 807C.

[00314] An exemplar embodiment may comprise 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 receiving an identification of a specific asset 809 under investigation. The instructions may include receiving information identifying the specific asset 809. The specific asset 809 under investigation may be a physical or virtual machine that has software operating a specific version that is vulnerable to any number of associated risks 803.

[00315] One among many embodiments may comprise 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 performing a forward analysis 811A of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment. . A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. And the scanning system 101 may identify at least one possible attack vector to the workload currently being analyzed. The network outside of the cloud environment may identify new or existing threats and attack vectors currently threatening the associated system and workload.

[00316] An embodiment may further comprise 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 performing a backward

analysis 811B of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset. Backward analysis 811B may include the possible impacts to an analyzed asset and those impacts' possible effect. The at least one exposure risk may include one or more entry point to access the exposed asset and any related machines potentially vulnerable to attack from the analyzed and exposed asset.

[00317] An embodiment may comprise 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 outputting a signal 813 to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. The display of a forward or backward path to a specific asset may be one of several visual displays such as linear, graphical, or through computer-generated images. The visualization of a plurality of entry points and lateral movement risks with the plurality of entry points may include visualization of the plurality of paths may sequence the paths based on likelihood of access via that path or the severity of potential threat in a path of access to an asset.

[00318] **Passive Key Identification Techniques**

[00319] Aspects of this disclosure may provide a technical solution to the challenging technical problem of identifying access keys (e.g., passwords, Secure Shell (SSH) keys, or cloud keys) to compute resources (e.g., machines, containers, storages, or any hardware or software component) in a cloud environment without using the access keys to access the compute resources. In existing technologies, scanning systems may identify access keys to compute resources and may need to verify the identified keys by actually accessing the compute resources using the access keys. In such cases, the actual accesses may generate logs or records (e.g., for successful or failed accesses) that can constitute activity patterns similar to an attacker to the system. Such activity patterns may distract a system administrator (e.g., an individual or a computer program) to misidentify the scanning system as an attacker. Further, actual accesses may create loads on both the scanning system and the compute resources and may reduce the computing resources and power for other tasks. To resolve such a technical problem, a cryptographic analysis may be performed to identify a first set of fingerprints of the identified keys, and trust configurations of the compute resources may be analyzed to identify a second set of fingerprints of the compute resources. The first and second set of fingerprints may be compared to match keys with the compute resources without using the keys to access the compute resources. By doing so, the scanning system may perform its

function without raising false alarms to the system administrator, while reducing computing costs to the cloud environment.

[00320] A cyber security system, as used herein, may refer to a system including any combination of software and hardware for enhancing security of a device, a platform, or another system in a network environment. By way of example, the cyber security may be implemented as a system including scanning system 101 in Fig. 1. As another example, scanning system 101 may include the disclosed cyber security system as a subset. A compute resource, as used herein, may refer to a database, a virtual machine, a storage, a keystore, a scanning system, a load balancer, a server, a computer, a container, or any physical or virtual component integrated in or communicatively connected to a cloud infrastructure. By way of example, the compute resource may be scanning system 101, any of databases 103A-103D, any of virtual machines 107A-107D, any of databases 109A-109D, any of storage 111A-111D, any of keystores 113A-113D, or load balancer 115 in cloud infrastructure 106 in Fig. 1.

[00321] A key to access a compute resource, as used herein, may refer to computerized data that includes a credential for granting permission to an accessor to access the compute resource, or a service or function provided by the compute resource. By way of example, the key may include at least one of a password, a remote access key, a script containing a password, a cloud key (e.g., a credential file for accessing a cloud service, such as an AWS credential file), an Secure Shell (SSH) key, an AWS key, a private component of a private-public key pair, or any type of credential that provide checked or unchecked access to the compute resource. Matching a key to a compute resource, as used herein, may refer to confirming, verifying, validating, or any operation or process of determining that the key is valid for an accessor (e.g., another compute resource) to access the compute resource. For example, to match a key to a compute resource, an accessor may test by using the key to access the compute resource and determines that the key is matched to the compute resource if such access is successful. As another example, to match a key to a compute resource, an accessor may perform methods or algorithms to verify that the key is matched to the compute resource without using the key to access the compute resource.

[00322] Consistent with disclosed embodiments, at least one processor may be configured to analyze a cloud environment to identify a plurality of keys to the compute resources in the cloud environment. In some embodiments, the at least one processor may analyze the cloud environment to identify the plurality of keys to the compute resources in the cloud environment in any of steps 231, 237, or 239 in FIG. 2D. In some embodiments, the at least one processor may analyze the cloud environment to identify the plurality of keys to the compute resources in the cloud environment in any step described in association with FIGS. 2A-2D. In some embodiments, at least one of the plurality of keys may include at least one of a password (e.g., a combination of alphanumeric characters and symbols), a script (e.g., a

PYTHON® script or a shell script) containing a password, a cloud key, or an Secure Shell (SSH) key. For example, a cloud key may be a key for accessing a cloud resource.

[00323] A cloud environment, as used herein, may refer to a computing environment running on a cloud. By way of example, the cloud environment may include or be part of cloud infrastructure 106 in Fig. 1. Analyzing a cloud environment, as used herein, may refer to an operation or a process of separating or dividing the cloud environment into parts or components (e.g., physical components or logic components), then determining nature and relationship of the parts or components based on data associated with the parts or components.

[00324] By way of example, if the cloud environment is cloud infrastructure 106 in Fig. 1, to analyze cloud infrastructure 106, the at least one processor may access cloud infrastructure 106 (e.g., via network 105 or via internal connections in cloud infrastructure 106) and obtain a list of compute resources in cloud infrastructure 106 (e.g., scanning system 101, any of databases 103A-103D, any of virtual machines 107A-107D, any of databases 109A-109D, any of storage 111A-111D, any of keystores 113A-113D, or load balancer 115). For example, the at least one processor may obtain the list of compute resources by reading a configuration file from a database (e.g., one of databases 109A-109D), or visit available compute resources one after another through communicative connections between the compute resources to generate the list of compute resources. After obtaining the list of compute resources in cloud infrastructure 106, the at least one processor may determine the nature of the compute resources, such as their types, numbers, functions, or any other characteristics or features. The at least one processor may further determine the relationship between the compute resources, such as communicative connections, access privileges, data input/output directions, or controls between the compute resources.

[00325] Identifying a key to a compute resource, as used herein, may refer to an operation or a process of locating, recognizing, or any operation or process or analyzing computerized data or information to determine that the computerized data or information is a key to the compute resources. By way of example, the at least one processor may analyze cloud infrastructure 106 in Fig. 1 to recognize a list of compute resources in cloud infrastructure 106 (e.g., scanning system 101, any of databases 103A-103D, any of virtual machines 107A-107D, any of databases 109A-109D, any of storage 111A-111D, any of keystores 113A-113D, or load balancer 115). Then, the at least one processor may read computerized data stored in a compute resource and identify that some of the computerized data are one or more keys to other compute resources in cloud infrastructure 106. For example, to recognize the keys, the at least one processor may read and compare the computerized data with records stored in keystores 113A-113D, and determine that the computerized data is a key if it matches a record in any of keystores 113A-113D. As another example, the at least one processor may read the computerized data and check its syntaxes, text string patterns, file formats, file properties,

encryption manners, library versions, software versions, or any other characteristics or features of the computerized data, and determine that the computerized data is a key if its checked characteristics or features fit a predetermined pattern of a key or fit an entry in a dictionary of keys. For example, the dictionary of keys may be stored in a keystore (e.g., any of keystores 113A-113D).

[00326] In some embodiments, the plurality of keys may be stored in at least one workload. A workload, for example, 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. By way of example, a workload may be a virtual machine, a database server, a container, a Hadoop node, an application, a storage server, a load balancer, or an IAM (Identity and Access Management) configuration. By way of example, with reference to FIG. 1, cloud infrastructure 106 may include workloads such as scanning system 101, databases 103A-103D, virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115.

[00327] Consistent with disclosed embodiments, at least one processor may be configured to perform a cryptographic analysis on the plurality of keys to identify a first set of fingerprints that uniquely identify each of the plurality of keys. The first set of fingerprints may be non-functional. In some embodiments, the at least one processor may perform the cryptographic analysis in step 237 of FIG. 2D. In some embodiments, the at least one processor may perform the cryptographic analysis in any step described in association with FIGS. 2A-2D.

[00328] A cryptographic analysis on data, as used herein, may refer to an operation or process of identifying ciphertext, ciphers and cryptosystems contained in the data to discover hidden aspects of the data that may improve or weaken security of the data or compute resources associated with the data. In some embodiments, to perform the cryptographic analysis on the plurality of keys, the at least one processor may perform an algorithm on the plurality of keys to convert the plurality of keys into another form of data. For example, the at least one processor may perform a hashing process on the plurality of keys (e.g., by inputting values of the keys to a hash function) to obtain corresponding hash values of respective keys.

[00329] A fingerprint of a key, as used herein, may refer to non-functional data (e.g., a numeric value, an alphanumeric string, or any combination of letters, numbers, or symbols) generated based on the key and may uniquely identify the key. By way of example, a fingerprint may be generated by performing a hashing process on a key or a part of the key. The hashing process may include, for example, an MD5 algorithm, an SHA-1 algorithm, an SHA-2 algorithm, an SHA-3 algorithm, a RIPEMD-160 algorithm, a BLAKE2 algorithm, a BLAKE3 algorithm, or any type of cryptographic hash algorithms. In such cases, the fingerprint may be a hash value of the key, a part (e.g., a truncated part) of the hash value of the key, a hash value of a part of the key, or a part (e.g., a truncated part) of the hash value of the part of the key. The hash value

may be unique (e.g., non-overlap with any hash value of any other key) and non-functional (e.g., unable to be used as a key to access a compute resource). To identify the first set of fingerprints, the at least one processor may perform the hashing process on the plurality of keys to obtain a plurality of hash values, and search the cloud environment to check whether there exists any compute resource that stores the plurality of hash values.

[00330] In some embodiments, at least one of the first set of fingerprints may be non-identical to any key of the plurality of keys. For example, if the first set of fingerprints are hash values, they may be non-identical to the plurality of keys.

[00331] Consistent with disclosed embodiments, at least one processor may be configured to analyze trust configurations of the compute resources to identify a second set of fingerprints for each of the compute resources. In some embodiments, the at least one processor may analyze the trust configurations in step 233 of FIG. 2D. In some embodiments, the at least one processor may analyze the trust configurations in any step described in association with FIGS. 2A-2D.

[00332] A trust configuration, as used herein, may refer to data (e.g., a stored file, a database entry, a value stored in a data structure, or any computerized information) that stores a pre-established trust relationship between at least two compute resources. For example, a trust configuration may be a file storing a trust policy (e.g., a public key infrastructure (PKI), a digital signature system, or an AWS trust policy). Analyzing a trust configuration, as used herein, may refer to an operation or a process of separating or dividing compute resources associated with the trust configuration into units or groups, then determining nature and relationship of the units or groups based on relationship data associated with the units or groups. For example, the relationship data may include flags, indicators, or any type of value that indicates a first compute resource has an access privilege to a second compute resource.

[00333] A fingerprint for a compute resource, as used herein, may refer to non-functional data (e.g., a numeric value, an alphanumeric string, or any combination of letters, numbers, or symbols) generated based on a key for accessing the compute resource and may uniquely identify the key. By way of example, a fingerprint for a compute resource may be generated by applying a cryptographic hash function (e.g., an SHA-1 or SHA-2 function) to a key for accessing the compute resource to obtain a sequence of bytes (e.g., a hash value or a modified hash value), and the sequence of bytes may be the fingerprint for the compute resource. The key may be uniquely associated with the compute resource.

[00334] For example, to identify the second set of fingerprints, the at least one processor may analyze a trust configuration file (e.g., a file storing fingerprints and their associated data) by reading the byte stream of the trust configuration file to determine whether any sequence of bytes that fits to a predetermined pattern (e.g., an alphanumeric string pattern). The sequences

of bytes that fit to the predetermined pattern may be determined as the second set of fingerprints.

[00335] Consistent with disclosed embodiments, at least one processor may be configured to compare the first set of fingerprints with the second set of fingerprints to match keys with the compute resources without using the keys to access the compute resources. In some embodiments, the at least one processor may compare the first set of fingerprints with the second set of fingerprints in any step described in association with FIGS. 2A-2D. Comparing a first fingerprint and a second fingerprint, as used herein, may refer to performing a textual, numerical, or semantic comparison of the first fingerprint and the second fingerprint, or performing an algorithmic conversion of at least one of the first fingerprint or the second fingerprint to further perform a textual, numerical, or semantic comparison of the conversion results. By way of example, a first compute resource may store a key (e.g., a public key) for accessing a second compute resource. The at least one processor may analyze the cloud environment to identify the key from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. A third compute resource (e.g., may be the same as or different from the second compute resource) may store in its trust configurations a second fingerprint of the key. The second fingerprint may be generated by a different processor at a different time. The at least one processor may then analyze the trust configurations of the third compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other. If the first fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource. As can be seen, after comparison, if the first set of fingerprints are matched with the second set of fingerprints, keys may be matched with the compute resources, and the at least one processor needs not use the keys to access the compute resources to confirm that the keys are matched to the compute resources.

[00336] Consistent with disclosed embodiments, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the first set of fingerprints. A multi-machine interaction, as used herein, may refer to data input/output, operation control, status polling, scanning, or any type of an interaction between a plurality of machines in a cloud environment. Analyzing a multi-machine interaction, as used herein, may refer to an operation or a process of separating or dividing machines in the cloud environment into units or groups, then determining nature and relationship of the units or groups based on relationship data associated with the units or groups. For example, the relationship data may be the first set of fingerprints.

[00337] By way of example, to analyze the multi-machine interaction, the at least one processor may compare the first set of fingerprints with the second set of fingerprints. If a first fingerprint of the first set of fingerprints matches with a second fingerprint of the second set of fingerprints, the at least one processor may determine that a first compute resource associated with the first fingerprint may have an inter-machine interaction (e.g., having an ability to access the other) with a second compute resource associated with the second fingerprint. By way of example, a first compute resource may store a key (e.g., a private key) for accessing a second compute resource. The at least one processor may analyze the cloud environment to identify the key from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. A third compute resource (e.g., may be the same as or different from the second compute resource) may store in its trust configurations a second fingerprint of the key. The at least one processor may then analyze the trust configurations of the third compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other (e.g., being the same). If the first fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource, and may further determine an inter-machine interaction in which the first compute resource has ability to access the second compute resource.

[00338] Consistent with disclosed embodiments, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the plurality of keys. For example, the at least one processor may use the plurality of keys as the relationship data for determining nature and relationship of compute resources in the cloud environment. By way of example, to analyze the multi-machine interaction, the at least one processor may compare a first key for accessing a first one of the compute resources and a second key for accessing a second one of the compute resources. If the first key matches with (e.g., is identical to) the second key, the at least one processor may determine that the first one of the compute resources may have an interaction (e.g., having an ability to access the other) with the second one of the compute resources.

[00339] By way of example, **Fig. 9** is a block diagram illustrating an exemplary process 900 of matching keys with compute resources in a cloud environment, consistent with the disclosed embodiments. While the block diagram may be described below in connection with certain implementation embodiments presented in other figures, those implementations are provided for illustrative purposes only, and are not intended to serve as a limitation on the block diagram. For example, the steps in Fig. 9 may be executed in any order, steps may be duplicated, or steps may be omitted. In some embodiments, the process 900 may be performed by at least one processor (e.g., a CPU) of a computing device or system (e.g., scanning system

101 in Fig. 1) to perform operations or functions described herein, and may be described hereinafter with reference to Figs. 1-2D by way of example. In some embodiments, some aspects of the process 900 may be implemented as software (e.g., program codes or instructions) that are stored in a memory or a non-transitory computer-readable medium. In some embodiments, some aspects of the process 300 may be implemented as hardware (e.g., a specific-purpose circuit). In some embodiments, the process 300 may be implemented as a combination of software and hardware.

[00340] Fig. 9 includes process blocks 902-908. At block 902, at least one processor may analyze a cloud environment to identify a plurality of keys to the compute resources in the cloud environment. In some embodiments, the plurality of keys may be stored in at least one workload. In some embodiments, at least one of the plurality of keys may include at least one of a password (e.g., a combination of alphanumeric characters and symbols), a script (e.g., a PYTHON® script or a shell script) containing a password, a cloud key, or an Secure Shell (SSH) key.

[00341] At block 904, the at least one processor may perform a cryptographic analysis on the plurality of keys to identify a first set of fingerprints that uniquely identify each of the plurality of keys. The first set of fingerprints may, in some embodiments, be non-functional. In some embodiments, at least one of the first set of fingerprints may be non-identical to the key.

[00342] At block 906, the at least one processor may analyze trust configurations of the compute resources to identify a second set of fingerprints for each of the compute resources. As discussed above, to identify the second set of fingerprints, in some embodiments, the at least one processor may analyze a trust configuration file (e.g., a file storing fingerprints and their associated compute resources) by reading the byte stream of the trust configuration file to determine whether any sequence of bytes that fits to a predetermined pattern (e.g., an alphanumeric string pattern). The sequences of bytes that fit to the predetermined pattern may be determined as the second set of fingerprints.

[00343] At block 908, the at least one processor may compare the first set of fingerprints with the second set of fingerprints to match keys with the compute resources without using the keys to access the compute resources. As discussed above, in some embodiments, a first compute resource may store a key (e.g., a private component of a private-public key pair) to a second compute resource, and the second compute resource may store in its trust configurations a second fingerprint of the key. The at least one processor may analyze the cloud environment to identify the key from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. The at least one processor may then analyze the trust configurations of the second compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other (e.g., being identical). If the first

fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource.

[00344] Consistent with disclosed embodiments, besides blocks 902-908, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the first set of fingerprints. By way of example, to analyze the multi-machine interaction, the at least one processor may compare the first set of fingerprints with the second set of fingerprints. As discussed above, in some embodiments, a first compute resource may store a key (e.g., a private key) for accessing a second compute resource. The at least one processor may analyze the cloud environment to identify the private component from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. A third compute resource (e.g., may be the same as or different from the second compute resource) may store in its trust configurations a second fingerprint of the key. The at least one processor may then analyze the trust configurations of the third compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other (e.g., being the same). If the first fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource, and may further determine an inter-machine interaction in which the first compute resource has ability to access the second compute resource.

[00345] Consistent with disclosed embodiments, besides blocks 902-908, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the plurality of keys. By way of example, to analyze the multi-machine interaction, the at least one processor may compare a first key for accessing a first one of the compute resources and a second key for accessing a second one of the compute resources.

[00346] **Hybrid Ephemeral Scanner Techniques**

[00347] Aspects of this disclosure may include accessing a primary account maintained in a cloud environment. A primary account as used herein, may refer to a principal or main identity created for a person in a computer or computing system. A primary account may also be created for machine entities, such as service accounts for running programs, system accounts for storing system files and processes, and root and administrator accounts for system administration. A cloud environment, as used herein, may refer to a platform implemented on, hosted on, and/or accessing servers that are accessed over the Internet. An example of a cloud environment is cloud infrastructure 106 in Fig. 1. In some embodiments, scanning system 101 of Fig. 1 may gain access to an account within cloud infrastructure 106.

[00348] Aspects of this disclosure may include receiving information defining a structure of the primary account, wherein the structure includes a plurality of assets.

Information as used herein, may refer to data received. Structure of the primary account as used herein, may refer to a set up or of the primary account. A plurality of assets as used herein, may refer to one or more data, devices, or components within an organization's system. For example, the plurality of assets may include any of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, load balancer 115, log files or databases, API gateway resources, API gateway REST APIs, Autoscaling groups, CloudTrail logs, CloudFront services, volumes, snapshots, VPCs, subnets, route tables, network ACLs, VPC endpoints, NAT gateways, ELB and ALB, ECR repositories, ECS clusters, services, and tasks, EKS, S3 bucket and Glacier storage, SNS topics, IAM roles, policies, groups, users, KMS keys, and Lambda functions.

[00349] In some embodiments, scanning system 101 of Fig. 1 may receive information relating to the structure of an account. In such an embodiment, the structure of the account may include virtual machines 107A-D and databases 109A-109D, for example. In some embodiments, the information may be acquired from another device, received via a one way communication, received in response to a request, retrieved from a storage device, or generated.

[00350] In some embodiments, the information may exclude raw data of the primary account. Raw data as used herein, may refer to a collection of data as gathered before it has been processed, cleaned, or analyzed. For example, raw data may include usage data, passwords, and cache.

[00351] Aspects of this disclosure may include deploying, inside the primary account, at least one ephemeral scanner configured to scan at least one block storage volume and to output metadata defining the at least one block storage volume. In some embodiments, the output may exclude raw data of the primary account. In some embodiments, the ephemeral scanner may be deployed inside a secondary account for which trust has been established with a primary account. For example, a secondary account on another system may have an existing trust relationship (such as a cloud trust policy) with a first account. The ephemeral scanner may be operated inside of that secondary account to scan block storage associated with the first account. Each account may be hosted on the same system or on different systems.

[00352] Further, in some embodiments, the at least one ephemeral scanner may be configured to periodically scan at least one block storage volume. For example, the scanner may be configured to scan at least one block storage volume once per hour, once per day, once per week, or more or less often.

[00353] Further, in some embodiments, the at least one ephemeral scanner may be periodically deployed to scan at least one block storage volume. For example, the scanner may be deployed for scanning at least one block storage volume once per hour, once per day, once per week, or more or less often.

[00354] Outputting metadata as used herein, may refer to producing a set of data that describes and gives information about other data. In some embodiments, scanning system 101 of Fig. 1 may scan a block storage volume and output data related to it (e.g., vulnerability information, configuration information, malware information, risk analysis information, and sensitive information). Periodically scanning as used herein, may refer to conducting a scan occurring or recurring at regular intervals. In some embodiments, the ephemeral scanner is configured to perform vulnerability scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for security vulnerabilities on the device. In some embodiments, the ephemeral scanner is configured to perform configuration scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for configuration information related to the device. In some embodiments, the ephemeral scanner may scan the block storage volume in order to look for security issues. In some embodiments, the ephemeral scanner is configured to perform malware scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for malware information related to the device. As discussed above with respect to Fig. 2D, in step 235, scanning system 101 may perform a step of malware scanning. In some embodiments, scanning system 101 may perform malware scanning across all filesystems in the snapshot (e.g., gathered from virtual machines 107A-107D or storage 111A-111D). Scanning system 101 may use multiple malware scanning software solutions to perform a malware scan against the filesystems, including one sourced from another vendor, such as bucketAV, Trend Micro Cloud One, Sophos Cloud Optix, Crowdstrike Falcon CWP, or others. In some embodiments, malware scanning in step 235 comprises utilizing signatures, heuristics, or sandboxing capabilities to deduce whether there is an infection on the machine.

[00355] In some embodiments, the ephemeral scanner is configured to perform lateral-movement risk analysis of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for lateral-movement risk information related to the device. For example, in some embodiments, scanning system 101 may perform a “backward” analysis of the specific asset to identify exposure risk to assets downstream of the specific asset, wherein the downstream exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset.

[00356] Further, as discussed above with respect to Fig. 2D, in step 237, scanning system 101 may perform a step of lateral movement scanning. An attacker who establishes a network foothold usually attempts to move laterally from one resource to another in search of rich targets such as valuable data. Stolen passwords and keys unlock access to servers, files, and privileged accounts. In some embodiments, scanning system 101 may gather keys from

each scanned system or device (e.g., virtual machines 107A-107D or storage 111A-111D). In some embodiments, scanning system 101 searches for passwords, scripts, shell history, repositories, or other data that may contain passwords, cloud access keys, SSH keys, or other key/password/access information that provide unchecked access to important resources. In some embodiments, scanning system 101 searches for such keys/passwords/access information and calculates a "hash" (a mathematical fingerprint) of each string. Scanning system 101 then attempts to match the hashed strings to hashes of strings that that are stored on different systems or devices. This will be used to detect the potential lateral movement between assets.

[00357] In some embodiments, the ephemeral scanner is configured to perform sensitive information scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for sensitive information related to the device. As discussed above with respect to Fig. 2D, in step 241, scanning system 101 may perform a step of sensitive information scanning. In some embodiments, scanning system 101 may search the snapshot for sensitive information, such as personally identifiable information (PII), Social Security numbers, healthcare information, or credit card numbers. In some embodiments, scanning system 101 may search data repository history as well. This is because it is not uncommon for an entire production environment repository to be cloned, with no one remembering the copy contains sensitive information. In some situations, detecting sensitive data not secured is critical in adherence to data privacy regulations.

[00358] In some embodiments, the ephemeral scanner is configured to perform container scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for container information related to the device. As discussed with respect to Fig. 2D, in step 243, scanning system 101 may perform a step of container scanning. In some embodiments, scanning system 101 may apply one or more of the preceding steps of Fig. 2D against containerized environments. In some embodiments, in order to do so, scanning system 101 reconstructs a container runtime layered file system (LFS) before recursively running one or more of steps 231-241 on the reconstructed file system.

[00359] In some embodiments, the ephemeral scanner is configured to perform keys and password scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for keys and password information related to the device. As discussed above with respect to Fig. 2D, in step 239, scanning system 101 may perform a step of key/password scanning. As one example situation, suppose there is a weak or unprotected password stored (in plain text) in storage 111A. For example, if a personal email account has been compromised, the passwords may be known about in advance. Scanning system 101 may search the snapshot for similar usernames or login names, and, either using known dictionaries or the account owner's previously leaked

passwords (stored in, e.g., database 103A), may attempt to login to one or more systems or devices in cloud infrastructure 106, and may record the result thereof.

[00360] In some embodiments, defining the at least one block storage volume includes presenting risk data without sharing consumer data or data that was used to identify the risk data. Risk data as used herein, may refer to information relating to causing harm. Consumer data as used herein, may refer to a user's personal information or any information trail user's leave behind as a result of their computer or Internet use. Data used to identify the risk as used herein, may refer to information for determining possible issues. In some embodiments, scanning system 101 of Fig. 1 may present information without sharing a user's personal information or any information trail user's leave behind as a result of their computer or Internet use.

[00361] In some embodiments, the metadata defining the at least one block storage volume includes at least one of: an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, or a profile configuration. An indication of an installed application as used herein, may refer to a sign or signal of a stored program. A version of an installed application as used herein, may refer to an older or newer form of a stored program. An operating system configuration as used herein, may refer to one or more computer system settings that have been set by default automatically or manually by a given program or the user. An application configuration as used herein, may refer to one or more computer program settings that have been set by default automatically or manually by a given program or the user. A profile configuration as used herein, may refer to one or more computer program file settings that have been set by default automatically or manually by a given program or the user.

[00362] Aspects of this disclosure may include receiving a transmission of the metadata from the at least one ephemeral scanner. In some embodiments, the transmission may exclude raw data of the primary account. A transmission of metadata as used herein, may refer to movement of a set of data that describes and gives information about other data. In some embodiments, scanning system 101 of Fig. 1 may receive data excluding any raw data (e.g., data as gathered before it has been processed, cleaned or analyzed).

[00363] Aspects of this disclosure may include analyzing the received metadata to identify a plurality of cybersecurity vulnerabilities. A plurality of cybersecurity vulnerabilities as used herein, may refer to a weakness in a system. For example, a cybersecurity vulnerability may be exploited by cybercriminals to gain unauthorized access to a computer system. In some embodiments, scanning system 101 of Fig. 1 may analyze the received data in order to identify any weakness in a system (e.g., to avoid cybercriminals gaining unauthorized access to a computer system). In some embodiments, vulnerability information may include information stored in a system.

[00364] Aspects of this disclosure may include correlating each of the identified plurality of cybersecurity vulnerabilities with one of the plurality of assets. In some embodiments, scanning system 101 of Fig. 1 may connect a security risk with an asset in the cloud environment. In such embodiments, the connection or correlation of vulnerability to asset may allow the system to address the vulnerability.

[00365] Aspects of this disclosure may include generating a report correlating the plurality of cybersecurity vulnerabilities with the plurality of assets. A report as used herein, may refer to a document containing information. For example, a report may notify the administrator of a website or application about a problem such as a security issue or vulnerability in the system that should be addressed. In some embodiments, scanning system 101 of Fig. 1 may create a report providing information related to the vulnerability and asset. In such an embodiment, the report may allow for the vulnerability to be addressed and resolved.

[00366] In such embodiments, the system may allow scanners (e.g., ephemeral scanners, scanning system 101) to run inside an account, including an Amazon Web Services, Azure, or GCP account. In this mode, a cloud cybersecurity service may generate ephemeral scanners inside an account that performs the same actions as a SAAS node but are logically hosted inside the account.

[00367] In some embodiments, the at least one processor is further configured to receive a transmission of updated metadata defining the at least one block storage volume in response to at least one change to the at least one block storage volumes. Updated metadata as used herein, may refer to a renewed set of data that describes and gives information about other data.

[00368] **Fig. 10** is a block diagram of a method 1000 for deployment of ephemeral scanners, consistent with disclosed embodiments. In some embodiments, the method may include seven (or more or less) steps:

[00369] Block 1002: Access a primary account maintained in a cloud environment. In some embodiments, scanning system 101 of Fig. 1 may gain access to an account within cloud infrastructure 106 using a password stored in a database, for example.

[00370] Block 1004: Receiving information defining a structure of the primary account, wherein the structure includes a plurality of assets. In some embodiments, the information may exclude raw data of the primary account. In some embodiments, scanning system 101 of Fig. 1 may receive data relating to the structure of an account. In such an embodiment, the structure of the account may include virtual machines 107A-D and databases 109A-109D, for example.

[00371] Block 1006: Deploying, inside the primary account, at least one ephemeral scanner configured to periodically scan at least one block storage volume and to output metadata defining the at least one block storage volume. In some embodiments, the output may exclude raw data of the primary account. In some embodiments, scanning system 101 of Fig. 1

may scan a block storage volume and output data related to it (e.g., vulnerability information, configuration information, malware information, risk analysis information, and sensitive information).

[00372] Block 1008: Receiving a transmission of the metadata from the at least one ephemeral scanner. In some embodiments, the transmission may exclude raw data of the primary account. In some embodiments, scanning system 101 of Fig. 1 may receive data excluding any raw data (e.g., data as gathered before it has been processed, cleaned or analyzed).

[00373] Block 1010: Analyzing the received metadata to identify a plurality of cybersecurity vulnerabilities. For example, a cybersecurity vulnerability may be exploited by cybercriminals to gain unauthorized access to a computer system. In some embodiments, scanning system 101 of Fig. 1 may analyze the received data in order to identify any weakness in a system (e.g., to avoid cybercriminals gaining unauthorized access to a computer system).

[00374] Block 1012: Correlating each of the identified plurality of cybersecurity vulnerabilities with one of the plurality of assets. In some embodiments, scanning system 101 of Fig. 1 may connect a security risk with an asset in the cloud environment. In such embodiments, the connection or correlation of vulnerability to asset may allow the system to address the vulnerability.

[00375] Block 1014: Generating a report correlating the plurality of cybersecurity vulnerabilities with the plurality of assets. For example, a report may notify the administrator of a website or application about a problem such as a security issue or vulnerability in the system that should be addressed. In some embodiments, scanning system 101 of Fig. 1 may create a report providing information related to the vulnerability and asset. In such an embodiment, the report may allow for the vulnerability to be addressed and resolved.

[00376] **Risk Information Aggregation Techniques**

[00377] **Fig. 11** represents a schematic block diagram 1100 illustrating an exemplary embodiment of a method for providing a dashboard aggregating risk information.

[00378] Assets operating in a cloud environment face myriad cybersecurity risks that vary in type and nature of threat with increasing regularity. In order to combine the varied and multiple cybersecurity risks an asset may encounter in a cloud environment into a single view to allow for faster and more flexible processing, prioritization and mitigation, a system and method of aggregating cybersecurity risks into a single dashboard is needed. Aggregation of multiple cybersecurity risks into a single view for an administrator is needed. In one of many embodiments, an exemplary embodiment may include a graphical user interface system for providing comprehensive cloud environment risk inventory visualization. A graphical user interface system may include systems similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows

10 Mobile, PalmOS-Web OS, or Firefox OS. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of risk inventories. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D.

[00379] An exemplary embodiment may also include at least one processor. The at least one processor may be part of a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing logic operations.

[00380] In an embodiment may include a processor configured to cause a display to present a plurality of asset categories (step 1101). A display may be graphical, sequential, or multi-dimensional. A plurality may include at least one asset category to be displayed. Asset categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category.

[00381] An account category may further include listing of assets based on type of account associated with that user. A type of account may include active user account, passive user account, administrator account, maintainer account, a system account, superuser account, or a guest user account.

[00382] A container category may include one or more containers currently listed in the system. A container may involve specific versions of programming language runtimes, libraries required to run software, or another method of packaging applications abstracted from the cloud environment.

[00383] A database category may include one or more databases managed and maintained in relation to the system. Databases may include a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database.

Databases may be implemented using ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances.

[00384] An image category may contain one or more images based on image type, size, and content. Images may include images of one or more virtual machines.

[00385] A managed service category may include one or more managed services listed in the above system, to include at least client or customer-owned systems being managed by a third party entity.

[00386] A messaging service category may include systems/programs communicating with other systems/programs using services such as Google Cloud Pub and Sub communications, AWS SQS, or similar queues found in Information Systems.

[00387] A monitoring category may include centralized or decentralized methods of security monitoring systems.

[00388] A network category may include one or more systems of establishing a form of communicative connectivity between systems, e.g., TCP-IP services.

[00389] A storage category may list one or more methods of data storage, to include Direct Attached Storage, Network Attached Storage, SSD Flash Drive Arrays, Hybrid Flash Arrays, Hybrid Cloud Storage, Backup Software, Backup Appliances, Cloud Storage, or similar. Storage may include data structures, instructions, or any other data to be contained in a storage medium.

[00390] A user category may include one or more users of a given system along with identifying information relating to its unique or semi-unique identifier, level of authorized access and permissions, and storage volumes accessible by said user.

[00391] An access category may include one or more levels of access permission and a listing of users or devices granted access at a given level of access permission to a system.

[00392] A virtual machine category may list one or more virtual machines accessible to a system and the levels and types of access granted to a given virtual machine.

[00393] A serverless category may list one or more storage volumes stored on a serverless computing system such as Kubeless, Fission, Google App Engine, AWS Lambda, Google Cloud Functions, IBM Cloud Function, IBM Cloud Code Engine, Microsoft Azure Functions, Cloudflare Workers, and Compute@Edge.

[00394] In an embodiment, an exemplar may include a processor configured to receive, via an input device, a selection of a particular asset category (step 1103). An input device may include a personal computer, tablet, smartphone, or virtual machine. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing

logic operations. A selection of one or more asset categories may be received and updated as new visualizations are developed based on the current threats to an asset in that category. Asset categories may be selected or deselected by an end user. Once a particular asset category is selected, it may be added to a list of selected asset categories with a list for deselected categories as well. The list of selected asset categories may organized per category or may be compiled to include all current threats in a comprehensive selected asset threat list.

[00395] One exemplary embodiment may further include a processor configured to cause the display to present a list of assets in the selected category that have cyber security risks (step 1105). When an asset in a selected category has a cyber security risk associated with it, the asset may appear in a list of all listed categories having a cyber security risk associated with it, and may be organized by individual category, type of cyber security risk, or some combination or sequential ordering methods based on the information available to the processor and its end user.

[00396] In one of many exemplary embodiments a processor may be configured to retrieve workload component cybersecurity risk information (step 1107A) for each listed asset. When a category of assets is selected, a processor may be configured to order those assets according to available cybersecurity risk information available to said processor, and may organize by type of risk, type of asset, or similar type of sequentialization. Cybersecurity risk information may be determined from historical threats to the workload component, the current active cybersecurity threats to workload components of certain types, or some combination of the same.

[00397] In an embodiment, a processor may be configured to retrieve cloud component cybersecurity risk information (step 1107B) for each listed asset. Cloud component cybersecurity risk information may include identification of the risk based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset. Cybersecurity risk information to a cloud component may be determined from historical threats within the cloud environment 106, current active cybersecurity threats to similar workload components in the cloud environment 106 or the level of risk to assets within the cloud environment 106, as outlined below.

[00398] An exemplary embodiment may include a system configured to cause the display to present a common interface (step 1109) providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information. The common interface may be configured to access component cybersecurity risk information based on the nature of the risk, the likelihood of the risk, or the severity of the risk to the component, including recursive risk – risk that can happen from an attacker getting access to a system and using it as a mean to reach other system. On the same interface, a different interface, or a similar and related interface, a display may present cloud component cybersecurity risk information based

on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset.

[00399] As an exemplary embodiment, the system may be configured to cause the display to present, in the common interface, an interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information (step 1111). With this display presented, an exemplary embodiment may allow for a user to prioritize data from additional sources. The interconnection between workload component cybersecurity risk information and cloud component cybersecurity risk information may generate an interface between them and may be presented as an interconnection between them. The common interface presented may include direct interconnections between the component cybersecurity risk information and the cloud component cybersecurity risk information or links from one set of cybersecurity risk information. For instance, the component cybersecurity risk information may include a link to access the cloud component cybersecurity risk information. Conversely, the cloud component cybersecurity risk information may include a link to access the component cybersecurity risk information.

[00400] In one embodiment, a system may be configured to provide the plurality of asset categories include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. A category may list the currently identified entities of that category type as described below. An account category may include one or more listed accounts of users of the system. An account category may further include listing of assets based on type of account associated with that user. A type of account may include active user account, passive user account, administrator account, maintainer account, a system account, superuser account, or a guest user account.

[00401] A container category may include one or more containers currently listed in the system. A container may involve specific versions of programming language runtimes, libraries required to run software, or another method of packaging applications abstracted from the cloud environment.

[00402] A database category may include one or more databases managed and maintained in relation to the system. Databases may include a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database. Databases may be implemented using ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances.

[00403] An image category may contain one or more images based on image type, size, and content. Images may include images of one or more virtual machines displayed. A

managed service category may include one or more managed services listed in the above system, to include at least client or customer-owned systems being managed by a third party entity.

[00404] A messaging service category may include services of users communicating with at least one other user such as Google Cloud Pub and Sub communications, AWS SQS, or similar queues found in Information Systems.

[00405] A monitoring category may include centralized or decentralized methods of security monitoring systems.

[00406] A network category may include one or more systems of establishing a form of communicative connectivity between systems, e.g., TCP-IP services.

[00407] A storage category may list one or more methods of data storage, to include Direct Attached Storage, Network Attached Storage, SSD Flash Drive Arrays, Hybrid Flash Arrays, Hybrid Cloud Storage, Backup Software, Backup Appliances, Cloud Storage, or similar. Storage may include data structures, instructions, or any other data to be contained in a storage medium.

[00408] A user category may include one or more users of a given system along with identifying information relating to its unique or semi-unique identifier, level of authorized access and permissions, and storage volumes accessible by said user.

[00409] An access category may include one or more levels of access permission and a listing of users or devices granted access at a given level of access permission to a system.

[00410] A virtual machine category may list one or more virtual machines accessible to a system and the levels and types of access granted to a given virtual machine.

[00411] A serverless category may list one or more storage volumes stored on a serverless computing system such as Kubeless, Fission, Google App Engine, AWS Lambda, Google Cloud Functions, IBM Cloud Function, IBM Cloud Code Engine, Microsoft Azure Functions, Cloudflare Workers, and Compute@Edge.

[00412] In one of many exemplary embodiments, a system may further include the common interface configured to display information relating to at least one of an asset type, a risk, a region, or an account. A common interface may include the ability to configure and reconfigure the asset type, risk, region, or account to appear in any order and at multiple formats for displaying the information relating to those categories.

[00413] In another embodiment, a system may provide the common interface configured to display description for each listed asset. The display of a description for each listed asset may be listed vertically or horizontally. The display may be displayed as written text, representative graphical figures, or similar method of displaying the aforementioned types of information for an asset.

[00414] In another embodiment, a system may further provide the common interface is configured to display at least one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, or an authentication. A vulnerability may include vulnerability data from a vulnerability database such as NVD, WPVulnDB, US-CERT, Node.js Security Working Group, OVAL – Red Hat, Oracle Linux, Debian, Ubuntu, SUSE, Ruby Advisory Database, JVN, Safety DB(Python), Alpine secdb, PHP Security Advisories Database, Amazon ALAS, RustSec Advisory Database, Red Hat Security Advisories, Microsoft MSRC, KB, Debian Security Bug Tracker, Kubernetes security announcements, Exploit Database, Drupal security advisories, JPCERT. An insecure configuration may include software flaws or misconfigurations, non-encrypted files, improper file or directory permissions, unpatched security flaws in server software, enabled or accessible administrative and debugging functions, administrative account vulnerabilities, SSL certificates and encryption settings not properly configured, or a similar misconfiguration. These misconfigurations may be discovered by performing a scan, e.g., may query devices and systems capable of routing traffic (e.g., load balancer 115, routers, switches, firewalls, and proxies) using an API provided through a cloud service provider's system to determine network configurations, and may evaluate them against known problematic configurations or other configurations. Malware indicated may include Adware, Botnets, Cryptojacking, Malvertising, Ransomware, Remote Administration Tools (RATs), Rootkits, Spyware, Trojans, Virus Malware, Worm Malware, or similar attack vehicles. A neglected asset may include an asset that has been improperly maintained, patched, or similar cybersecurity security measure. A data at risk may include any packet of data that may be exposed to a cybersecurity threat. A lateral movement may include any pathway between assets where a cybersecurity risk can travel from one affected asset to another affected asset. An authentication may include a username, password, one-time password, two-factor authentication, or any other authentication mechanism to gain access to a cloud service provider's system. A common interface may display more than one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, and an authentication separately or concurrently for an asset. A display of said information may be ordered and listed in any order depending on user preference and input.

[00415] An exemplary embodiment may further include a system wherein the common interface is configured to display one or more possible attack vectors reaching the each listed asset. Generating this common interface display may include generating a map of an asset to include recording information such as a region identifier, site identifier, datacenter identifier, physical address, network address, workload name, or any other identifier which may be acquired via an API provided through a cloud service provider's system and demonstrating via a display the myriad attack vectors currently available to an analyzed asset. Display of the one or

more possible attack vectors reaching the each listed asset may be displayed as written text, representative graphical figures, or similar method of displaying the one or more possible attack vectors.

[00416] Another exemplary embodiment may further include a system wherein the common interface is configured to display a recommended mitigation tactic for the each listed asset. A recommended mitigation tactic may be generated based on historical data of the cybersecurity risk, the vulnerability attributed to the listed asset, or the data contained on the each listed asset. Mitigation tactics may be selected based on user input or may be implemented automatically based on the mitigation tactics likelihood of a successful mitigation of the cybersecurity risk. A mitigation tactic may include increased frequency of scanning, heightened access control measures, firewalls and antivirus software patches, increased patch management scheduling, continuous workload monitoring, or similar tactics.

[00417] In another exemplary embodiment is a system wherein the common interface is configured to display one or more workload metrics associated with the each listed asset. Workload metrics may be a statistical representation of the current performance of the workload, the capacity for additional processing of the workload, and the historical performance of the workload based on statistical trend.

[00418] Another exemplary embodiment may include a system wherein the at least one processor is further configured to provide a cybersecurity report for the each listed asset. A cybersecurity report may be generated by scanning system 101 following a complete scan of an analyzed asset, and may include a snapshot of one or more of the number of total cybersecurity threats to a listed asset, number of cybersecurity threats mitigated by recent performance, methods of cybersecurity risk mitigation in the listed asset, rate of cybersecurity threats detected in the listed asset, and possibility of failed cybersecurity mitigation tactics based on the frequency and nature of the historical cybersecurity threats. Data collected to generate a cybersecurity report may include, among many things, operating system packages, installed software applications, libraries, and program language libraries such as Java archives, Python packages, Go modules, Ruby gems, PHP packages, and Node.js modules, other software applications, library versions, software versions, and other identifying characteristics of software and operating systems, lists of users of each system or device (e.g., virtual machines 107A-D), each system's or device's services, password hashes, and application-specific configurations for software/services such as Apache, Nginx, SSH, and other services, bugs or other configuration risks, malware scan results, passwords, scripts, shell history, repositories, or other data that may contain passwords, cloud access keys, SSH keys, or other key/password/access information that provide unchecked access to important resources.

[00419] In another embodiment, a system may include the at least one processor is configured to identify a risk level distribution among the listed assets. The at least one processor

may be configured to list risk level distribution by highest level of risk, lowest level of risk, or largest number of cybersecurity risks attributed among the listed assets. A risk level may be high if the web service is connected to the Internet (e.g., has at least one public port forwarded through a firewall to the web service, is able to be accessed through a load balancer, is able to be accessed via reverse proxy). A risk level may be medium risk if the web service is only accessible internally (e.g., because of firewall configuration). A risk level may be low risk if access to the web server is blocked by a configuration of cloud infrastructure 106.

[00420] Another exemplary embodiment may include a system wherein the at least one processor is configured to receive a search query for a specific risk. A search query may be input by type of risk, name of a risk, or another semi-unique manner of identifying a specific cybersecurity risk. A search may be initiated by end user operating user device 102, by a maintainer or administrator of an analyzed asset, or by a virtual machine associated with operating scanning system 101, A search may be initiated by inputting a number of inputs, to include keywords, phrases, file type, or similar identifiers unique or semi-unique to a given cybersecurity risk. Upon initiating a search, the results generated by the search may be stored in the analyzed asset, transmitted to the operator of a processor running side scanning system 101, or generated for review by an end user operating user device 102. The search and display of a specific risk may be based on historical trends of cybersecurity risks in the listed asset, severity of cybersecurity risk to a listed asset, or a historical search for past cybersecurity threats to a listed asset.

[00421] Another embodiment may include a system wherein the at least one processor is configured to identify one or more assets vulnerable to the specific risk. For instance, a processor operating scanning system 101 may be configured to perform a scan of a set of one or more assets with potential exposure to a given risk, and upon identifying vulnerable pathways for an attack vector to reach a set of identified assets, may generate a list of assets that are potentially or currently exposed to one or more specific risk. Assets vulnerable to a specific risk may be determined to be vulnerable based on the type of the specific risk, the existence of specific risks in adjacent or similar workloads, and related risks that may also cause an asset to be vulnerable based on the specific risk determined.

[00422] In another embodiment is a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing a display to present a plurality of asset categories (step 1101). The method may include the use of a graphical user interface system and may include systems similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows 10 Mobile, PalmOS-Web OS, or Firefox OS. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and

the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of risk inventories. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D.

[00423] One of several embodiments may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising receiving, via an input device, a selection of a particular asset category. A method may include a personal computer, tablet, smartphone, or virtual machine. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing logic operations. A selection of one or more asset categories may be received and updated as new visualizations are developed based on the current threats to an asset in that category. Asset categories may be selected or deselected by an end user. Once a particular asset category is selected, it may be added to a list of selected asset categories with a list for deselected categories as well. The list of selected asset categories may organized per category or may be compiled to include all current threats in a comprehensive selected asset threat list.

[00424] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing the display to present a list of assets in the selected category that have cybersecurity risks (step 1105). A display may be graphical, sequential, or multi-dimensional. A list of selected categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. When an asset in a selected category has a cyber security risk associated with it, the asset may appear in a list of all listed categories having a cyber security risk associated with it, and may be organized by individual category, type of cyber security risk, or some combination or sequential ordering methods based on the information available to the processor and its end user.

[00425] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising retrieving workload component cybersecurity risk information 307A for each listed asset. When a category

of assets is selected, a processor may be configured to order those assets according to available cybersecurity risk information available to said processor, and may organize by type of risk, type of asset, or similar type of sequentialization. Cybersecurity risk information may be determined from historical threats to the workload component, the current active cybersecurity threats to workload components of certain types, or some combination of the same.

[00426] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising retrieving cloud component cybersecurity risk information 307B. Cloud component cybersecurity risk information may include identification of the risk based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset. Cybersecurity risk information to a cloud component may be determined from historical threats within the cloud environment 106, current active cybersecurity threats to similar workload components in the cloud environment 106 or the level of risk to assets within the cloud environment 106, as outlined above.

[00427] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing the display to present a common interface 309 providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information. The common interface may be configured to access component cybersecurity risk information based on the nature of the risk, the likelihood of the risk, or the severity of the risk to the component. On the same or a similar and related interface, a display may present cloud component cybersecurity risk information based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset.

[00428] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing the display to present, in the common interface, an interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information. The interconnection between workload component cybersecurity risk information 307A and cloud component cybersecurity risk information 307B may generate an interface between them 309 and may be presented as an interconnection between them 311. The common interface presented may include direct interconnections between the component cybersecurity risk information and the cloud component cybersecurity risk information or links from one set of cybersecurity risk information. For instance, the component cybersecurity risk information may include a link to access the cloud component cybersecurity risk information. Conversely, the cloud component cybersecurity risk information may include a link to access the component cybersecurity risk information.

[00429] In another embodiment, a method may include wherein the plurality of asset categories include at least one of an account category, an authentication category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. A category may list the currently identified entities of that category type as described below. An account category may include one or more listed accounts of users of the system. An account category may further include listing of assets based on type of account associated with that user. A type of account may include active user account, passive user account, administrator account, maintainer account, a system account, superuser account, or a guest user account.

[00430] A container category may include one or more containers currently listed in the system. A container may involve specific versions of programming language runtimes, libraries required to run software, or another method of packaging applications abstracted from the cloud environment.

[00431] A database category may include one or more databases managed and maintained in relation to the system. Databases may include a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database. Databases may be implemented using ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances.

[00432] An image category may contain one or more images based on image type, size, and content. Images may include images of one or more virtual machines displayed.

[00433] A managed service category may include one or more managed services listed in the above system, to include at least client or customer-owned systems being managed by a third party entity.

[00434] A messaging service category may include services of users communicating with at least one other user such as Google Cloud Pub and Sub communications, AWS SQS, or similar queues found in Information Systems.

[00435] A monitoring category may include centralized or decentralized methods of security monitoring systems.

[00436] A network category may include one or more systems of establishing a form of communicative connectivity between systems, e.g., TCP-IP services.

[00437] A storage category may list one or more methods of data storage, to include Direct Attached Storage, Network Attached Storage, SSD Flash Drive Arrays, Hybrid Flash Arrays, Hybrid Cloud Storage, Backup Software, Backup Appliances, Cloud Storage, or similar.

Storage may include data structures, instructions, or any other data to be contained in a storage medium.

[00438] A user category may include one or more users of a given system along with identifying information relating to its unique or semi-unique identifier, level of authorized access and permissions, and storage volumes accessible by said user.

[00439] An access category may include one or more levels of access permission and a listing of users or devices granted access at a given level of access permission to a system.

[00440] A virtual machine category may list one or more virtual machines accessible to a system and the levels and types of access granted to a given virtual machine.

[00441] Another embodiment may include a method wherein the common interface is configured to display information relating to at least one of an asset type, a risk, a region, or an account. A common interface may include the ability to configure and reconfigure the asset type, risk, region, or account to appear in any order and at multiple formats for displaying the information relating to those categories.

[00442] Another embodiment may include a method wherein the common interface is configured to display description for each listed asset. The display of a description for each listed asset may be listed vertically or horizontally. The display may be displayed as written text, representative graphical figures, or similar method of displaying the aforementioned types of information for an asset.

[00443] Another embodiment may include a method wherein the common interface is configured to display at least one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, or an authentication. A vulnerability may include vulnerability data from a vulnerability database such as NVD, WPVulnDB, US-CERT, Node.js Security Working Group, OVAL – Red Hat, Oracle Linux, Debian, Ubuntu, SUSE, Ruby Advisory Database, JVN, Safety DB(Python), Alpine secdb, PHP Security Advisories Database, Amazon ALAS, RustSec Advisory Database, Red Hat Security Advisories, Microsoft MSRC, KB, Debian Security Bug Tracker, Kubernetes security announcements, Exploit Database, Drupal security advisories, JPCERT. An insecure configuration may include software flaws or misconfigurations, non-encrypted files, improper file or directory permissions, unpatched security flaws in server software, enabled or accessible administrative and debugging functions, administrative account vulnerabilities, SSL certificates and encryption settings not properly configured, or a similar misconfiguration. These misconfigurations may be discovered by performing a scan, e.g., may query devices and systems capable of routing traffic (e.g., load balancer 115, routers, switches, firewalls, and proxies) using an API provided through a cloud service provider's system to determine network configurations, and may evaluate them against known problematic configurations or other configurations. Malware indicated may include Adware, Botnets, Cryptojacking, Malvertising,

Ransomware, Remote Administration Tools (RATs), Rootkits, Spyware, Trojans, Virus Malware, Worm Malware, or similar attack vehicles. A neglected asset may include an asset that has been improperly maintained, patched, or similar cybersecurity security measure. A data at risk may include any packet of data that may be exposed to a cybersecurity threat. A lateral movement may include any pathway between assets where a cybersecurity risk can travel from one affected asset to another affected asset. An authentication may include a username, password, one-time password, two-factor authentication, or any other authentication mechanism to gain access to a cloud service provider's system. A common interface may display more than one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, and an authentication separately or concurrently for an asset. A display of said information may be ordered and listed in any order depending on user preference and input.

[00444] Another embodiment may include a method wherein the common interface is configured to display one or more possible attack vectors reaching the each listed asset. Generating this common interface display may include generating a map of an asset to include recording information such as a region identifier, site identifier, datacenter identifier, physical address, network address, workload name, or any other identifier which may be acquired via an API provided through a cloud service provider's system and demonstrating via a display the myriad attack vectors currently available to an analyzed asset. Display of the one or more possible attack vectors reaching the each listed asset may be displayed as written text, representative graphical figures, or similar metho of displaying the one or more possible attack vectors.

[00445] Another embodiment may include a method wherein the common interface is configured to display a recommended mitigation tactic for the each listed asset. A recommended mitigation tactic may be generated based on historical data of the cybersecurity risk, the vulnerability attributed to the listed asset, or the data contained on the each listed asset. Mitigation tactics may be selected based on user input or may be implemented automatically based on the mitigation tactics likelihood of a successful mitigation of the cybersecurity risk. A mitigation tactic may include increased frequency of scanning, heightened access control measures, firewalls and antivirus software patches, increased patch management scheduling, continuous workload monitoring, or similar tactics.

[00446] Another embodiment may include a method wherein the common interface is configured to display one or more workload metrics associated with the each listed asset. Workload metrics may be a statistical representation of the current performance of the workload, the capacity for additional processing of the workload, and the historical performance of the workload based on statistical trend.

[00447] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing a display to present a plurality of asset categories 301. A plurality may include at least one asset category to be displayed. Asset categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category.

[00448] Another embodiment may include 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 for matching keys with compute resources, the operations comprising receiving, via an input device, a selection of a particular asset category (step 1103). An input device may include a personal computer, tablet, smartphone, or virtual machine. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing logic operations. A selection of one or more asset categories may be received and updated as new visualizations are developed based on the current threats to an asset in that category. Asset categories may be selected or deselected by an end user. Once a particular asset category is selected, it may be added to a list of selected asset categories with a list for deselected categories as well. The list of selected asset categories may organized per category or may be compiled to include all current threats in a comprehensive selected asset threat list.

[00449] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing the display to present a list of assets in the selected category that have cybersecurity risks (step 1105). A display may be graphical, sequential, or multi-dimensional. A list of selected categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. When an asset in a selected category has a cyber security risk associated with it, the asset may appear in a list of all listed categories having a cyber security risk associated with it, and may be organized by individual category, type of cyber security risk, or

some combination or sequential ordering methods based on the information available to the processor and its end user.

[00450] Another embodiment may include 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 for matching keys with compute resources, the operations comprising retrieving workload component cybersecurity risk information for each listed asset (step 1107A). When a category of assets is selected, a processor may be configured to order those assets according to available cybersecurity risk information available to said processor, and may organize by type of risk, type of asset, or similar type of sequentialization. Cybersecurity risk information may be determined from historical threats to the workload component, the current active cybersecurity threats to workload components of certain types, or some combination of the same.

[00451] Another embodiment may include 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 for matching keys with compute resources, the operations comprising retrieving cloud component cybersecurity risk information for each listed asset (step 1107B). Cloud component cybersecurity risk information may include identification of the risk based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset. Cybersecurity risk information to a cloud component may be determined from historical threats within the cloud environment 106, current active cybersecurity threats to similar workload components in the cloud environment 106 or the level of risk to assets within the cloud environment 106, as outlined above.

[00452] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing the display to present a common interface providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information for each listed asset (step 1109). The common interface may be configured to access component cybersecurity risk information based on the nature of the risk, the likelihood of the risk, or the severity of the risk to the component. On the same or a similar and related interface, a display may present cloud component cybersecurity risk information based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset.

[00453] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing the display to present, in the common interface, an

interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information (step 1111). The interconnection between workload component cybersecurity risk information (step 1107A) and cloud component cybersecurity risk information (step 1107B) may generate an interface between them (step 1109) and may be presented as an interconnection between them (step 1111). The common interface presented may include direct interconnections between the component cybersecurity risk information and the cloud component cybersecurity risk information or links from one set of cybersecurity risk information. For instance, the component cybersecurity risk information may include a link to access the cloud component cybersecurity risk information. Conversely, the cloud component cybersecurity risk information may include a link to access the component cybersecurity risk information.

[00454] In some embodiments, a processor may cause a display of one or more risks aggregated in the same visualization. **Fig. 12** is a schematic block diagram illustrating an exemplary embodiment of a visual representation of displaying aggregated cybersecurity risk information, consistent with disclosed embodiments. A visualization may be generated from one or more sources based on scanning system 101. For example, a visual display may include a cloud provider visualization 1201, displaying where the multiple sources of data may be derived from. A cloud service provider may include services such as Cloudflare, Amazon Web Services, Google Cloud, IBM Cloud, Oracle Cloud, Microsoft Azure, or similar. This is further displayed via a communicative connection through an access port 1203 to a data server 1205.

[00455] Data server 1205 may include servers such as 109A-D. Data server 1205 may then display risk information that may be communicatively passed through web server 1207, demonstrating one or more possible sources of the one or more cybersecurity risk information. Data server 1205 may be communicatively connected to a web server 1207 (through access port 1203, which may be the same or a different port from port 1203 between cloud service provider 1201 and data server 1205), which can then be communicatively connected to one or more users via user access points 1209. This may include, among other things, user access that may be verified or validated via a credential 1211. Each of these connections may demonstrate one or more risk paths that may be exploited in a single view to allow for prioritization of threats and analysis of possible users who may exploit those risks and vulnerabilities, as discussed above. This may allow an administrator to prioritize risks to a given storage volume as well as assess lateral storage volumes that may be similarly risked based on the assessed threats in a single image.

[00456] **Fig. 13** is a schematic block diagram illustrating an exemplary embodiment of a visual representation of a flow path for aggregated risk information, consistent with disclosed embodiments. In another embodiment, a processor may be configured to display a flow path of aggregated risk information. In one view, a displayed visualization may include user access

1301, which displays risk information such as service vulnerability 1303 and an insecure private key ~~1303~~ 1305. These vulnerabilities or risks may be displayed to an administrator via developer server 1307 to an administrator through administrator access 1309. As previously discussed, this will allow an administrator to visualize myriad risks to a storage volume in a single view that will allow an administrator to prioritize risks and analyze possible exploiters of this risk information on lateral systems and storage volumes.

[00457] In another embodiment, risk information may be shown in one or more views. This may involve a risk path similar to the exemplary embodiment of Fig. 13. In an exemplary embodiment, in a single visualization data from multiple sources may be displayed. This may involve identifying risk information such as the origin of the risk and potential exploiters of a risk. In some embodiments, this visualization may be done for a specific risk. In another embodiment, this visualization may be done for a combination of risks.

[00458] Disclosed embodiments may include any one of the following bullet-pointed features alone or in combination with one or more other bullet-pointed features, whether implemented as a method, by at least one processor, and/or stored as executable instructions on non transitory computer readable media:

- establishing a trusted relationship between a source account in a cloud environment and a scanner account;
- using the established trust relationship, utilize at least one cloud provider API to identify workloads in the source account;
- using the at least one cloud provider API to query a geographical location of at least one of the identified workloads;
- receiving an identification of the geographic location;
- using the cloud provider APIs to access block storage volumes of the at least one workload;
- determining a file-system of the at least one workload;
- mounting the block storage volumes on a scanner based on the determined file-system;
- activating a scanner at the geographic location;
- reconstructing from the block storage volumes a state of the workload; and
- assessing the reconstructed state of the workload to extract insights.
- wherein the geographic location includes an identifier of a physical site.
- wherein mounting includes selecting a driver corresponding to the determined file system.
- wherein the at least one processor is further configured to deploy a scanner at the geographical location.

- wherein the identification of the geographic location comprises an identification of a data center, at least one of a data center name, Internet Protocol (IP) address, name of the cloud provider, or a unique identity.
- wherein the reconstructed state of the workload includes at least two of an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, a profile configuration, a log, or a database content;
- wherein the at least one processor is further configured to update the reconstructed state of the workload based on at least one change to the block storage volumes.
- wherein the insights comprise at least one of a vulnerability associated with the workload or a composition of installed applications associated with the workload.
- wherein to mount the block storage volumes on the scanner, the at least one processor is configured to create a snapshot of the block storage volumes;
- wherein to mount the block storage volumes on the scanner, the at least one processor is configured to mount the snapshot of the block storage volumes on the scanner
- wherein the at least one processor is further configured to encrypt the snapshot of the block storage volumes;
- wherein the at least one processor is further configured to mount the encrypted snapshot of the block storage volumes on the scanner;
- using a cloud provider API, access a block storage volume of a workload maintained in a cloud storage environment;
- identifying an installed software application in the accessed block storage volume;
- analyzing the identified installed software application to determine an associated software version;
- accessing a data structure of known software vulnerabilities for a plurality of versions of software applications;
- performing a lookup of the identified installed software version in the data structure to identify known vulnerabilities; and
- performing at least one of query the cloud provider API to determine network accessibility information related to the workload, identify at least one port on which the vulnerable application is accessible, use network accessibility information and at least one port to identify one or more vulnerabilities susceptible to attack from outside the workload.
- implementing a remedial action in response to the identified one or more vulnerabilities.
- wherein the remedial measure includes transmitting an alert to a device associated with an administrator.

- wherein querying the cloud provider API to determine network accessibility information related to the workload further comprises examining data sources associated with the workload
- wherein querying the cloud provider API to determine network accessibility information related to the workload further comprises determining the network accessibility information based on the examined data sources.
- wherein querying the cloud provider API to determine network accessibility information related to the workload further comprises: wherein the network accessibility information includes at least one of: data from an external data source, cloud provider information, or at least one network capture log.
- identifying the installed software application comprises extracting data from at least one of operating system packages, libraries, or program language libraries;
- identifying the installed software application comprises identifying the installed software application based on the extracted data;
- identifying a version of the installed software application;
- wherein the identified installed software application includes one or more scripts;
- wherein the data structure includes aggregated vulnerability data;
- wherein the aggregated vulnerability data includes data from one or more third-party vendors;
- wherein the aggregated vulnerability data includes data collected by a scanner;
- wherein the aggregated vulnerability data includes at least one of an advisory, an exploit, a security announcement, or a known bug.
- querying the cloud provider API to determine network accessibility information related to the workload further comprises: wherein the network accessibility information includes at least one of: data from an external data source, cloud provider information, or at least one network capture log.
- 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;
- querying the at least one cloud provider API to determine at least one networking configuration of the entities;
- building a graph connecting the plurality of entities based on the networking configuration;
- 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;

- traversing the graph to identify at least one source originating via the internet and reaching the workload; and
- outputting a risk notification associated with the workload.
- wherein the plurality of entities includes a virtual network appliance.
- wherein the virtual network appliance is at least one of a load balancer, a firewall, a proxy, or a router.
- 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.
- wherein the graph includes a data structure sequentially connecting entities.
- wherein the graph includes directional vectors indicating directions of dataflow.
- wherein building the graph comprises identifying individual entities as nodes
- wherein building the graph comprises connecting the nodes.
- wherein the graph includes port numbers associated with the workload.
- wherein the graph includes a path from the at least one source to the workload.
- utilizing a cloud provider API to access a block storage volume of a workload maintained on a target account in a target system of a cloud storage environment;
- utilizing a scanner at a location of the block storage volume and on a secondary system other than the target system;
- scanning the block storage volume for malicious code, using the secondary system;
- identifying malicious code based on the scan; and
- outputting from the secondary system, a notification of a presence of malicious code in the target system.
- wherein the location of the block storage volume includes at least one of: the target account, a secondary system account, a cloud provider account, or a third party account.
- wherein scanning the block storage volume includes scanning disk-backed memory.
- wherein the disk-backed memory includes at least one of a page file or a cache file.
- wherein the secondary system includes at least one of a virtual machine, a container, or a serverless function.
- wherein the secondary system has an operating system different from an operating system of the target account.
- wherein the malicious code includes a rootkit.
- wherein utilizing a scanner includes suspending an operation of the scanner after the scan of the block storage volume.

- wherein utilizing a scanner includes modifying a pre-utilized scanner at the location of the block storage volume based on information related to the target account to obtain a modified scanner;
- wherein utilizing a scanner includes utilizing the modified scanner.
- identifying assets in a cloud environment;
- identifying risks associated with each of the identified asset;
- identifying relationships between at least some of the identified assets, the relationships including at least one of a trust, a network connectivity, or a mechanism of network proxying;
- receiving an identification of a specific asset under investigation;
- performing a forward analysis of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment;
- performing a backward analysis of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset; and
- outputting a signal to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points.
- wherein the network outside the cloud environment includes the Internet.
- wherein the assets in the cloud environment include at least one of: a virtual machine, a network appliance, a storage appliance, a compute instances, or an engine instance.
- wherein identifying the assets in a cloud environment includes identifying the assets based on at least one of: an identity and access management policy, an organization policy, or an access policy.
- wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream asset or a downstream asset.
- wherein the visualization includes a presentation of the alternative paths.
- wherein the presentation of the forward and backward paths indicates port numbers for each pathway.
- wherein the visualization of the entry points indicates at least one entry point at risk.
- monitoring network activities of the assets in a cloud environment.
- detecting detect a potential risk associated with the specific asset based on the monitored network activities.

- detecting a potential risk associated with the specific asset based on a network activity of the specific asset.
- detecting a potential risk associated with the specific asset based on a network activity of an upstream asset of the specific asset.
- detecting a potential risk associated with the specific asset based on a network activity of a downstream asset of the specific asset.
- analyzing a cloud environment to identify a plurality of keys to the compute resources in the cloud environment;
- performing a cryptographic analysis on the plurality of keys to identify a first set of fingerprints that uniquely identify each of the plurality of keys, the first set of fingerprints being non-functional;
- analyzing trust configurations of the compute resources to identify a second set of fingerprints for each of the compute resources; and
- comparing the first set of fingerprints with the second set of fingerprints to match keys with the compute resources without using the keys to access the compute resources.
- wherein the plurality of keys are stored in at least one workload.
- wherein at least one of the first set of fingerprints is not identical to any key of the plurality of keys.
- wherein at least one of the plurality of keys includes at least one of a password, a script containing a password, a private component of a private-public key pair, a cloud key, or an Secure Shell (SSH) key.
- testing validity of at least one of the plurality of keys.
- analyzing a multi-machine interaction in the cloud environment using the first set of fingerprints.
- analyzing the multi-machine interaction includes comparing the first set of fingerprints with the second set of fingerprints.
- analyzing a multi-machine interaction in the cloud environment using the plurality of keys.
- accessing a primary account maintained in a cloud environment;
- receiving information defining a structure of the primary account, wherein the structure includes a plurality of assets, and wherein the information excludes raw data of the primary account;
- deploying, inside the primary account or inside a secondary account for which trust has been established with the primary account, at least one ephemeral scanner configured to scan at least one block storage volume and to output metadata defining the at least one block storage volume, the output excluding raw data of the primary account;

- receiving a transmission of the metadata from the at least one ephemeral scanner, wherein the transmission excludes raw data of the primary account;
- analyzing the received metadata to identify a plurality of cybersecurity vulnerabilities;
- correlating each of the identified plurality of cybersecurity vulnerabilities with one of the plurality of assets; and
- generating a report correlating the plurality of cybersecurity vulnerabilities with the plurality of assets.
- wherein defining the at least one block storage volume includes presenting risk data without sharing consumer data or data that was used to identify the risk data.
- wherein the metadata defining the at least one block storage volume includes at least one of: an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, or a profile configuration.
- receiving a transmission of updated metadata defining the at least one block storage volume in response to at least one change to the at least one block storage volumes.
- wherein the ephemeral scanner is configured to perform vulnerability scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform configuration scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform malware scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform lateral-movement risk analysis of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform sensitive information scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform container scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform keys and password scanning of the at least one block storage volume.
- causing a display to present a plurality of asset categories;
- receiving, via an input device, a selection of a particular asset category;
- causing the display to present a list of assets in the selected category that have cyber security risks;
- for each listed asset, retrieving workload component cybersecurity risk information, and
- for each listed asset. retrieving cloud component cybersecurity risk information;

- for each listed asset, causing the display to present a common interface providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information; and
- for each listed asset, causing the display to present, in the common interface, an interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information.
- wherein the plurality of asset categories include at least one of an account category, an authentication category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category.
- wherein the common interface is configured to display information relating to at least one of an asset type, a risk, a region, or an account.
- wherein the common interface is configured to display description for each listed asset.
- wherein the common interface is configured to display at least one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, or an authentication.
- wherein the common interface is configured to display one or more possible attack vectors reaching the each listed asset.
- wherein the common interface is configured to display a recommended mitigation tactic for the each listed asset.
- wherein the common interface is configured to display one or more workload metrics associated with the each listed asset.
- wherein the at least one processor is further configured to provide a cybersecurity report for the each listed asset.
- wherein the at least one processor is configured to identify a risk level distribution among the listed assets.
- receiving a search query for a specific risk; or
- identifying one or more assets vulnerable to the specific risk.

[00459] Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the disclosed embodiments disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the disclosed embodiments being indicated by the following claims.

[00460] Computer programs based on the written description and methods of this specification are within the skill of a software developer. The various programs or program modules can be created using a variety of programming techniques. One or more of such

software sections or modules can be integrated into a computer system, non-transitory computer readable media, or existing software.

[00461] Moreover, while illustrative embodiments have been described herein, the scope includes any and all embodiments having equivalent elements, modifications, omissions, combinations (e.g., of aspects across various embodiments), adaptations or alterations based on the present disclosure. The elements in the claims are to be interpreted broadly based on the language employed in the claims and not limited to examples described in the present specification or during the prosecution of the application. These examples are to be construed as non-exclusive. Further, the steps of the disclosed methods can be modified in any manner, including by reordering steps or inserting or deleting steps. It is intended, therefore, that the specification and examples be considered as exemplary only, with a true scope and spirit being indicated by the following claims and their full scope of equivalents.

Docket Number: 15635.0004-00304

SYSTEMS AND METHODS FOR SIDE SCANNING

PRIORITY

[001] This application claims the benefit of U.S. Provisional Application No. 63/180,048 filed on April 26, 2021, the contents of which are hereby incorporated by reference.

TECHNICAL FIELD

[002] Disclosed embodiments include new systems and methods for securing cloud infrastructure to help meet compliance mandates, without disrupting business operations in live environments.

BACKGROUND

[003] Modern organizations typically depend on cloud infrastructure for data storage and processing. Data storage and processing nodes, among other things, power today's electronic infrastructure. In contrast to early network-centric days, an organization may have little in the way of "on-prem" (on-premises) systems and may run its entire business through systems having shared resources. Such an infrastructure may be the entire inventory of cloud assets for an organization, including running and stopped workloads of all types: virtual machines, containers, storage objects, load balancers, IAM (Identity and Access Management) configurations, and more. Organizations are searching for effective ways to scan their entire cloud estate to look for risks stemming from vulnerabilities, misconfigurations, malware, lateral movement risk, weak and leaked passwords, and improperly secured PII.

[004] The infrastructure of these organizations are plainly different than '90s-style physical servers running on bare metal. Existing agents and scanners from the on-prem days have been simply reused in cloud environments without reconfiguration. Before the cloud, it was required to secure physical hosts. That meant spending time installing multiple security agents—one for each server. IP addresses were assigned to physical assets and they seldom changed. Even then, however, agent integration was tedious and coverage rarely reached 100% of assets. In a cloud environment, scaling up and down is a frequent occurrence—possibly thousands of times per hour across multiple clouds—and all within a CI/CD pipeline that builds an organization's infrastructure. There are containers and virtual machines to contend with, and agents carry high operational costs.

[005] This causes problems of inefficiency and overuse of resources, including excess processor cycles on regular scans. In turn, such problems and inefficiencies can lead to a higher monetary cost for hosted resources as some cloud providers charge per processing cycle.

[006] Relying on agents for security visibility in the cloud also has drawbacks. Visibility is limited to only those assets that are already known and accessible to an agent, requiring that such assets are capable of having the agent installed and maintained in the first place. The assets also must have ongoing network connectivity to the backend, something that makes the

development cycle inefficient as developers tend not to be inclined to deploy agents on virtual machines, in containers, and in serverless configurations—let alone dealing with their never-ending maintenance—thus leading to insecure implementations.

[007] An authenticated scan allows for direct host access using remote protocols such as SSH or RDP. The scanner uses a privileged account to log in and determine how secure each host is from an inside vantage point. While authenticated scans can successfully discover potential vulnerabilities, they are limited as they require a privileged account on each scanned host. Furthermore, scans use significant system resources during the test procedures and require opening ports that by themselves pose a security risk.

[008] An unauthenticated scan can only examine publicly visible information and is not able to provide detailed information about assets. It essentially acts as a friendly attacker. An unauthenticated scan can easily miss identifying some assets and vulnerabilities, making it much less effective. For example, a website at mydomain.com/email_campaign not linked from a main website will not be scanned unless the scanner is manually configured. This can leave organizations exposed to vulnerabilities in areas where the scanner cannot reach.

[009] While unauthenticated scanners act like an attacker, they often get stuck where a real attacker would not. For example, a CAPTCHA can easily prevent automatic mechanisms (such as scanners) from registering. But such techniques would not have any effect on a human attacker.

[0010] Cloud security posture managers (CSPMs) are scanning tools developed specifically for the cloud. Rather than going inside the machine, a CSPM analyzes the cloud configuration itself for weaknesses. CSPMs are used to discover, assess, and solve cloud misconfigurations but provide shallow coverage where cloud security is concerned because CSPMs have difficulty detecting critical risks such as vulnerabilities, malware, and misconfigurations within the workloads themselves. Organizations choosing to combine agent-based solutions with a CSPM end up getting flooded with separate alerts that lack context which results in alert fatigue on behalf of security analysts.

[0011] Therefore, there is a need for improved security scanning systems that solve the problem with existing systems and are more adapted for use in cloud infrastructures.

[0012] The foregoing background is for illustrative purposes and is not intended as a discussion of the scope of the prior art or an admission of prior art.

SUMMARY

[0013] This summary is provided for the convenience of the reader to present a basic understanding of a few exemplary embodiments of this disclosure, and does not wholly define the breadth of the disclosure. This summary is not an extensive overview of all contemplated embodiments and is intended to neither identify key or critical elements of all embodiments nor to delineate the scope of any or all aspects. Its sole purpose is to present some features of one

or more embodiments in a simplified form as a prelude to the more detailed description presented later. For convenience, the term "embodiments" may be used herein to refer to one or more features or combinations of features described herein. A description associated with an embodiment does not necessarily apply to all embodiments.

[0014] Embodiments of the present disclosure may deliver instant-on, workload level visibility across cloud infrastructure assets without running opcodes on the target scanned workloads, helping organizations to: (1) Detect risks such as vulnerabilities, malware, misconfigurations, lateral movement risk, and unsecured sensitive data; (2) See cloud inventory at every layer: infrastructure, operating system, applications, and data; (3) Discover and see previously missed assets.

[0015] Embodiments of the present disclosure may further use an out-of-band process to reach cloud workloads through the runtime storage layer, combining this with information gathered from cloud provider APIs, including but not limited to K8S APIs.

DRAWINGS

[0016] The drawings and figures contained herein are a part of this specification, illustrate several embodiments consistent with the disclosure and, together with the description, serve to explain principles of the disclosure.

[0017] **Fig. 1** is a schematic block diagram illustrating an exemplary embodiment of a network including computerized systems, consistent with disclosed embodiments.

[0018] **Fig. 2A** is a schematic block diagram illustrating an exemplary embodiment of a process for integration, scanning, assessment, analyzing/reporting, and change reviewing, consistent with disclosed embodiments.

[0019] **Fig. 2B** is a schematic block diagram illustrating an exemplary embodiment of a process for integration, consistent with disclosed embodiments.

[0020] **Fig. 2C** is a schematic block diagram illustrating an exemplary embodiment of a process for scanning/mapping, consistent with disclosed embodiments.

[0021] **Fig. 2D** is a schematic block diagram illustrating an exemplary embodiment of a process for assessment, consistent with disclosed embodiments.

[0022] **Fig. 3** is a block diagram of method for insight extraction, consistent with disclosed embodiments.

[0023] **Fig. 4** depicts a cybersecurity system performing a side scanning function to protect against potential vulnerabilities, consistent with disclosed embodiments.

[0024] **Fig. 5** of the disclosed embodiments describes a method of operating a cybersecurity system performing a side scanning function to protect against potential vulnerabilities, consistent with disclosed embodiments.

[0025] **Fig. 6** is a block diagram of method for risk prioritization, consistent with disclosed embodiments.

[0026] **Fig. 7** is a block diagram illustrating an exemplary process of cyber security scanning for a cloud environment, consistent with the disclosed embodiments.

[0027] **Fig. 8** is a schematic block diagram illustrating an exemplary embodiment of a system for performing visualization of forward and backward facing threats, consistent with the disclosed embodiments.

[0028] **Fig. 9** is a block diagram illustrating an exemplary process of matching keys with compute resources in a cloud environment, consistent with the disclosed embodiments.

[0029] **Fig. 10** is a block diagram of a method for deployment of ephemeral scanners, consistent with disclosed embodiments.

[0030] **Fig. 11** represents a schematic block diagram illustrating an exemplary embodiment of a method for providing a dashboard aggregating risk information, consistent with disclosed embodiments.

[0031] **Fig. 12** is a schematic block diagram illustrating an exemplary embodiment of a visual representation of displaying aggregated cybersecurity risk information, consistent with disclosed embodiments.

[0032] **Fig. 13** is a schematic block diagram illustrating an exemplary embodiment of a visual representation of a flow path for aggregated risk information, consistent with disclosed embodiments.

DETAILED DESCRIPTION

[0033] Exemplary embodiments are described with reference to the accompanying drawings. The figures are not necessarily drawn to scale. While examples and features of disclosed principles are described herein, modifications, adaptations, and other implementations are possible without departing from the spirit and scope of the disclosed embodiments. For example, with this detailed description provides a few examples, these implementations are provided as examples only and are not restrictive of the claim concepts that follow or any of the descriptions herein. Also, the words “comprising,” “having,” “containing,” and “including,” and other similar forms are intended to be equivalent in meaning and be open ended in that an item or items following any one of these words is not meant to be an exhaustive listing of such item or items or meant to be limited to only the listed item or items. It should also be noted that as used herein and in the appended claims, the singular forms “a,” “an,” and “the” include plural references unless the context clearly dictates otherwise.

[0034] In the following description, various working examples are provided for illustrative purposes. However, it is to be understood that the present disclosure may be practiced without one or more of these details.

[0035] It is intended that one or more aspects of any mechanism may be combined with one or more aspect of any other mechanisms, and such combinations are within the scope of this disclosure.

[0036] Various embodiments are described herein with reference to a system, method, device, or computer readable medium. It is intended that the disclosure of one is a disclosure of all. For example, it is to be understood that disclosure of a computer readable medium described herein also constitutes a disclosure of methods implemented by the computer readable medium, and systems and devices for implementing those methods, via for example, at least one processor. It is to be understood that this form of disclosure is for ease of discussion only, and one or more aspects of one embodiment herein may be combined with one or more aspects of other embodiments herein, within the intended scope of this disclosure.

[0037] Embodiments described herein may refer to a non-transitory computer readable medium containing instructions that when executed by at least one processor, cause the at least one processor to perform a method. Non-transitory computer readable medium may include any medium capable of storing data in any memory in a way that may be read by any computing device with a processor to carry out methods or any other instructions stored in the memory. The non-transitory computer readable medium may be implemented as hardware, firmware, software, or any combination thereof. Moreover, the software may preferably be implemented as an application program tangibly embodied on a program storage unit or computer readable medium consisting of parts, or of certain devices and/or a combination of devices. The application program may be uploaded to, and executed by, a machine having any suitable architecture. Preferably, the machine may be implemented on a computer platform having hardware such as one or more central processing units ("CPUs"), a memory, and input/output interfaces. The computer platform may also include an operating system and microinstruction code. The various processes and functions described in this disclosure may be either part of the microinstruction code or part of the application program, or any combination thereof, which may be executed by a CPU, whether or not such a computer or processor is explicitly shown. In addition, various other peripheral units may be connected to the computer platform such as an additional data storage unit and a printing unit. Furthermore, a non-transitory computer readable medium may be any computer readable medium except for a transitory propagating signal.

[0038] Memory employed herein may include a Random Access Memory (RAM), a Read-Only Memory (ROM), a hard disk, an optical disk, a magnetic medium, a flash memory, other permanent, fixed, volatile or non-volatile memory, or any other mechanism capable of storing instructions. The memory may include one or more separate storage devices collocated or disbursed, capable of storing data structures, instructions, or any other data. The memory may further include a memory portion containing instructions for the processor to execute. The memory may also be used as a working scratch pad for the processors or as a temporary storage.

[0039] Some embodiments may involve at least one processor. A processor may be any physical device or group of devices having electric circuitry that performs a logic operation on input or inputs. For example, the at least one processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing logic operations. The instructions executed by at least one processor may, for example, be pre-loaded into a memory integrated with or embedded into the controller or may be stored in a separate memory.

[0040] In some embodiments, the at least one processor may include more than one processor. Each processor may have a similar construction, or the processors may be of differing constructions that are electrically connected or disconnected from each other. For example, the processors may be separate circuits or integrated in a single circuit. When more than one processor is used, the processors may be configured to operate independently or collaboratively. The processors may be coupled electrically, magnetically, optically, acoustically, mechanically or by other means that permit them to interact.

[0041] Consistent with the present disclosure, disclosed embodiments may involve a network. A network may constitute any type of physical or wireless computer networking arrangement used to exchange data. For example, a network may be the Internet, a private data network, a virtual private network using a public network, a Wi-Fi network, a LAN or WAN network, and/or other suitable connections that may enable information exchange among various components of the system. In some embodiments, a network may include one or more physical links used to exchange data, such as Ethernet, coaxial cables, twisted pair cables, fiber optics, or any other suitable physical medium for exchanging data. A network may also include a public switched telephone network ("PSTN") and/or a wireless cellular network. A network may be a secured network or unsecured network. In other embodiments, one or more components of the system may communicate directly through a dedicated communication network. Direct communications may use any suitable technologies, including, for example, BLUETOOTH™, BLUETOOTH LE™ (BLE), Wi-Fi, near field communications (NFC), or other suitable communication methods that provide a medium for exchanging data and/or information between separate entities.

[0042] Certain embodiments disclosed herein may also include a computing device for cloud cybersecurity, the computing device may include processing circuitry communicatively connected to a network interface and to a memory, wherein the memory contains instructions to be executed. The computing devices may be devices such as mobile devices, desktops, laptops, tablets, or any other devices capable of processing data. Such computing devices may include a display such as an LED display, augmented reality (AR), virtual reality (VR) display.

[0043] "Software" as used herein refers broadly to any type of instructions, whether referred to as software, firmware, middleware, microcode, hardware description language, or otherwise. Instructions may include code (e.g., in source code format, binary code format, executable code format, or any other suitable format of code). The instructions, when executed by the one or more processors, may cause the processing system to perform the various functions described in further detail herein.

[0044] The one or more processors may be implemented with any combination of general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), field programmable gate array (FPGAs), programmable logic devices (PLDs), controllers, state machines, gated logic, discrete hardware components, dedicated hardware finite state machines, or any other suitable entities that can perform calculations or other manipulations of information.

[0045] Aspects of this disclosure may provide technical solutions to challenges associated with cloud cybersecurity. Disclosed embodiments include methods, systems, devices, and computer-readable media. For ease of discussion, a system is described below with the understanding that the disclosed details may equally apply to methods, devices, and computer-readable media.

[0046] Embodiments of the present disclosure include technology referred to as "SideScanning." In contrast to some existing systems and techniques, these embodiments may provide a distinct advantage because the technology does not necessarily require entering into each workload to inspect data. Rather, some embodiments use an out-of-band process to reach cloud workloads through a runtime storage layer, combining this with metadata gathered from API provided through a cloud service provider's system, thus providing visibility of cloud environments both at a low level and with context, without the requirement for an agent or network scanner.

[0047] **Fig. 1** is a schematic block diagram 100 illustrating an exemplary embodiment of a network including computerized systems, consistent with the disclosed embodiments. Diagram 100 includes user device 102, network 105, and cloud infrastructure 106.

[0048] Cloud infrastructure 106 includes scanning system 101, databases 103A-103D, virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115. While particular numbers and arrangements of devices, systems, and connections, are depicted in exemplary Fig. 1, in some embodiments, each of the devices, systems, or connections may be omitted, duplicated, or modified. For example, in some embodiments, databases 109A-109D may exist as only a single database; in other embodiments, cloud infrastructure 106 may exist as one or more distinct or combined infrastructures (e.g., operated by the same or different cloud services). In some embodiments, scanning system 101 and/or databases 103A-103D may be part of cloud infrastructure 106 (and

may be connected to the various other systems and devices in cloud infrastructure 106); in other embodiments, scanning system 101 and/or databases 103A-103D may be separate from cloud infrastructure 106 (e.g., connected to the systems and devices in cloud infrastructure 106 through network 105).

[0049] Scanning system 101, in some embodiments, may include one or more computer systems. Each of the one or more computer systems may include memory storing instructions and at least one CPU configured to execute those instructions to perform operations as discussed herein. In some embodiments, the instructions cause the CPU to perform scanning operations. In some embodiments, scanning system 101 may perform a scanning operation on one or more workloads (e.g., systems, devices, resources, etc.) in cloud infrastructure 106.

[0050] User device 102, in some embodiments, may include a mechanism operated by a user to control scanning system 101. For example, in some embodiments, user device 102 may be any of a personal computer, a server, a thin client, a tablet, a personal digital assistant, a smartphone, a kiosk, or any other mechanism enabling data input. User device 102 may be operated to instantiate functionality, access data, or otherwise interact with scanning system 101 via network 105, as described herein.

[0051] Databases 103A-103D include data stores for use by scanning system 101. In some embodiments, one or more of databases 103A-103D may be implemented as a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database. In some embodiments, one or more of databases 103A-103D may be implemented as flat file stores, data stores, or other non-database storage systems. In some embodiments, databases 103A-103D may be implemented using one or more of ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances.

[0052] Network 105 may be implemented as one or more interconnected data networks. For example, network 105 may include one or more of any type of network (including infrastructure) that provides communications, exchanges information, and/or facilitates the exchange of information, such as the Internet, a Local Area Network, a near field communication (NFC) network, or other suitable connection(s) that enables the sending and receiving of information between the components of system 100. Network 105 may be implemented using wireless connections, wired connections, or both. In some embodiments, one or more components of system 100 can communicate through network 105. In some embodiments, one or more components of system 100 may communicate directly through one or more dedicated communication links. While particular devices and systems are shown as connected to network 105, in some embodiments, more or fewer devices and systems may be connected to network 105.

[0053] Cloud infrastructure 106 may be implemented as a set of devices and systems offered by a single cloud service provider. For example, cloud infrastructure 106 may comprise devices and systems that are part of Amazon Web Services, Microsoft Azure, Google Cloud Platform, IBM Cloud, Alibaba Cloud, or any other cloud platform provider. In some embodiments, one or more of the devices and systems in cloud infrastructure may require authentication or other identity validation for access. For example, to access virtual machine 107A, a user may be required to enter a password or provide a key. Systems (e.g., scanning system 101 or user device 102) may administer or interact with cloud infrastructure 106 using a cloud service provider's system (not pictured).

[0054] Virtual machines 107A-107D may include one or more devices and systems that implement a virtualized/emulated version of a computer. A virtual machine may be implemented as an emulated version of a computer - including an operating system, memory, storage, graphics processing - such that it can be indistinguishable from a standard (non-virtual) machine to a running program. A computer system, referred to as a "host," may operate virtual machines 107A-107D, referred to as "guests," by dividing the resources of the host between the virtual machines such that each virtual machine is isolated from one another. This means that in some embodiments, one virtual machine, and the operating system(s) and application(s) running thereon, is only able to access the resources that are allocated to that virtual machine and cannot access resources allocated to other virtual machines. For example, if a host has 32 gigabytes of random access memory (RAM), and is hosting three virtual machines, the host may segment 8 gigabytes of RAM to each virtual machine such that each virtual machine may only access data in that 8 gigabytes of RAM and not any of the other 24 gigabytes. Examples of commercial virtual machine software and services include VMWare Workstation, VMWare Server, VMWare ESXi, VirtualBox, Parallels Desktop, Parallels RAS, Amazon Machine Image, Amazon ECS, Kubernetes, Microsoft Hyper-V, and Xen.

[0055] Databases 109A-109D may include data stores for use by devices and systems in cloud infrastructure 106. In some embodiments, one or more of databases 109A-109D may be implemented as a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database. In some embodiments, one or more of databases 109A-109D may be implemented as flat file stores, data stores, or other non-database storage systems. In some embodiments, databases 109A-109D may be implemented using one or more of ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances. Databases 109A-109D may store data usable by devices or systems in cloud infrastructure 106. The data, in some embodiments, may include e-commerce data (e.g., shipments, orders, inventory), media data (e.g., pictures, movies, streaming data), financial data (e.g., banking data, investment data), or other data.

[0056] Storage 111A-111D may include storage systems for use by devices and systems in cloud infrastructure 106. In some embodiments, one or more of storage 111A-111D may be implemented as a hard drive, a RAID array, flash memory, optical storage, or any other kind of storage. Each of 111A-111D may include one or more filesystems (e.g., Amazon Elastic File System, GlusterFS, Google File System, Hadoop Distributed File System, OpenZFS, S3, Elastic Block Storage).

[0057] In some embodiments, systems and devices of cloud infrastructure 106 may use databases 109A-109D to store data that is accessed frequently (where, for example, access is required within a few milliseconds), and may use storage 111A-111D to store data that is accessed less frequently (where, for example, access is required within a few minutes or hours).

[0058] Keystores 113A-113D may include systems storing keys for accessing data and functionality. For example, to access certain data or systems, a system may require the use of passwords or keys in keystores 113A-113D for authentication. The data and functionality that the keys grant access to may be part of cloud infrastructure 106 or may be separate from cloud infrastructure 106. For example, keystores 113A-113D may include systems that store public and private keys (e.g., for use via SSH), may store passwords (e.g., login information for websites or programs), may store tokens (e.g., one-time passcodes), or the like. In some embodiments, keystores 113A-113D may be implemented as one or more of Amazon Web Services KMS, Azure Key Vault, or Google KMS.

[0059] Load balancer 115 may include one or more systems that balance incoming requests between the different systems and devices of cloud infrastructure 106. For example, load balancer 115 may be configured to determine usage (e.g., processor load, used storage capacity) of systems or devices in cloud infrastructure 106 to assist in determining where to route an incoming request from network 105 to store data, perform processing, or retrieve data.

[0060] Load balancer 115 may be configured to receive an incoming request from user device 102. Upon receipt of the request, load balancer 115 may consult a data store (part of or separate from load balancer 115; not pictured) to determine usage or forecasted usage of various systems or devices in cloud infrastructure 106, and may forward the request to the systems or devices having the lowest usage or forecasted usage.

[0061] **Fig. 2A** is a schematic block diagram illustrating an exemplary embodiment of a process 200 for integration, scanning, assessment, and review, consistent with the disclosed embodiments. In some embodiments, the steps in Fig. 2A are executed using software and hardware of scanning system 101. In some embodiments, the steps in Fig. 2A may be performed in an order other than those depicted in Fig. 2A. In some embodiments, steps may also be omitted, repeated, or modified. In some embodiments, information gathered in one step of Fig. 2A may be used to provide context or other information for use in another step of Fig. 2A. In other embodiments, the steps in Fig. 2A may be executed by other devices.

[0062] Process 200 begins with step 201. In step 201, scanning system 101 may execute a process of integration. The integration process may be performed by scanning system 101 with cloud infrastructure 106. In some embodiments, the integration process includes creating a connection between an account on scanning system 101 and an account on cloud infrastructure 106. The process of integration in step 201 may, in some embodiments, be implemented as described below with respect to Fig. 2B.

[0063] In step 203, scanning system 101 may execute a process of scanning/mapping. The scanning/mapping process may be performed by scanning system 101 with cloud infrastructure 106. In some embodiments, the scanning and mapping process may include analyzing data relating to cloud infrastructure 106 using scanning system 101, by reading cloud infrastructure 106 through the connection made in step 201 and generate a "map" (e.g., a data structure or data collection) representing systems and devices in cloud infrastructure 106. The process of scanning and mapping in step 203 may, in some embodiments, be implemented as described below with respect to Fig. 2C.

[0064] In step 205, scanning system 101 may execute a process of assessing. The assessing process may be performed by scanning system 101. In some embodiments, the assessing process may include reviewing vulnerabilities, infrastructure, interconnections, data, and other information, using scanning system 101. The process of assessing in step 205 may, in some embodiments, be implemented as described below with respect to Fig. 2D.

[0065] In step 207, scanning system 101 may execute a process of analyzing/reporting. The analyzing/reporting may be performed by scanning system 101. In some embodiments, the change reviewing process comprises scanning system 101 scanning cloud infrastructure 106 again to determine the differences between an earlier observed snapshot of cloud infrastructure 106 and the current state thereof. In some embodiments, scanning system 101 may analyze information (e.g., steps 203 and/or 205) to generate reports. The reports may list what each vulnerability is, where it is located, and its priority. In this way, security engineers and DevOps teams may be able to easily assess how to best allocate their time and attention.

[0066] In some embodiments, analyzing/reporting in step 207 may include scanning system 101 combining conclusions from different environmental perspectives (e.g., metadata) into a single model. For example, scanning system 101 may map the running services on cloud infrastructure 106 and consider collected vulnerability data. Scanning system 101, in some embodiments, may generate a visualization of the map for review, listing where the vulnerabilities are on a two-dimensional graphical representation of cloud infrastructure 106.

[0067] In some embodiments, scanning system 101 may indicate in the map whether an asset (e.g., a system or device) is Internet-facing and easily accessible to attackers, exposed only to internal assets or if it is private altogether (and may be less critical). For example, based on the contextual map, scanning system 101 may perform a "forward" analysis of the specific asset

under identification to identify at least one possible Internet-originating attack vector to the asset. Alternatively or additionally, scanning system 101 may perform a "backward" analysis of the specific asset to identify exposure risk to assets downstream of the specific asset, wherein the downstream exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset. Both backward and forward analysis can be recursively, for one or more hops, Scanning system 101 may present these analyses to a user via graphical user interface or expose them via an API.

[0068] As one example, scanning system 101 may determine a vulnerability in a web service. In some embodiments, scanning system 101 may score the vulnerability as: high risk if the web service is connected to the Internet (e.g., has at least one public port forwarded through a firewall to the web service, is able to be accessed through a load balancer, is able to be accessed via reverse proxy); medium risk if the web service is only accessible internally (e.g., because of firewall configuration); and low risk if access to the web server is blocked by a configuration of cloud infrastructure 106. As another example, if a machine is stopped (e.g., turned off, not running, not connected), a vulnerability, but one that is less likely to be exploited because the machine isn't running. This affects its risk score and other mitigating factors.

[0069] In step 209, scanning system 101 may execute a process of change reviewing. The change reviewing may be performed by scanning system 101. In some embodiments, the change reviewing process may include scanning system 101 scanning cloud infrastructure 106 again to determine the differences between an earlier observed snapshot of cloud infrastructure 106 and the current state thereof.

[0070] In some embodiments, scanning system 101 may monitor cloud change, or "delta," logs (e.g., Cloud event logs such as AWS CloudTrail, monitoring a network-related change in state, a trust-related change in state, or a disk configuration-related change in state in at least one of the primary asset group or a secondary asset group) and contextualize associated risks discovered within a customer cloud account. In some embodiments, scanning system 101 may generate a new map of assets based on a review of all systems or devices in cloud infrastructure 106 (e.g., as described above with respect to step 203), and compare the new map to the map generated earlier in step 203 to determine which assets have changed. In some embodiments, this comparison may require comparing data, metadata, network connections, software configurations, firewall configurations, or any other aspects related to cloud infrastructure 106, including, for example, determining new assets that are created or assets for which their configuration has changed.

[0071] After step 209, as depicted in Fig. 2A, process 200 may return to step 205. Scanning system 101 may then perform step 205 again. In some embodiments, steps 205, and 207 may be performed only on the devices or systems that have experienced a change that was

detected by step 209. In some embodiments, process 200 may return to step 203 after step 209.

[0072] **Fig. 2B** is a schematic block diagram illustrating an exemplary embodiment of a process 210 for integration, consistent with the disclosed embodiments. In some embodiments, the steps in Fig. 2B are executed using software and hardware of scanning system 101.Integration, in some embodiments, is a process for creating a trust relationship between scanning system 101 and cloud infrastructure 106.

[0073] In some embodiments, the steps in Fig. 2B may be performed in an order other than those depicted in Fig. 2B. In some embodiments, steps may also be omitted, repeated, or modified. In some embodiments, information gathered in one step of Fig. 2B may be used to provide context or other information for use in another step of Fig. 2B. In other embodiments, the steps in Fig. 2B may be executed by other devices.

[0074] Process 210 may begin with step 211 to initiate a connection to cloud infrastructure 106. In step 211, scanning system 101 may send a message to user device 102 instructing a user to authenticate, or log in, to a cloud service provider's system operating cloud infrastructure 106. For example, the user may use a username, password, one-time password, two-factor authentication, or any other authentication mechanism to gain access to a cloud service provider's system.

[0075] Concurrently with or after the first message, scanning system 101 may send a second message to user device 102, instructing the user to generate a role. The second message may include instructions for the user to follow to generate the role. In step 213, a user may provide (e.g., via a keyboard at user device 102) a role definition to the cloud service provider's system. In some embodiments, the role definition includes read-only permissions and permissions to read a block storage layer (containing block storage volumes). In some embodiments, scanning system 101 provides a role formation template (e.g., an Amazon Web Services CloudFormation Template) for use with cloud infrastructure 106 to create the necessary role. In step 213, the user may utilize user device 102, for example, by copying and pasting a URL of the template, downloading and uploading the template to the cloud service provider's system, or selecting the template from a list of templates.

[0076] In step 215, the cloud service provider's system may generate a string (e.g., a "key" or "resource name") for use by scanning system 101. In some embodiments, this string may be used to enable access by scanning system 101 to the workload of cloud infrastructure as permitted by the generated role. A user using user device 102 may copy the string and paste it into a user interface presented by scanning system 101 on user device 102. Other aspects of transmitting this string to scanning system 101 are possible as well.

[0077] In step 217, scanning system 101 determines that it is able to access cloud infrastructure 106. For example, scanning system 101 may attempt to perform a command such

as authenticating to cloud infrastructure 106 using the string received in step 215 or attempting to perform a command using cloud infrastructure (e.g., requesting a listing of files stored on storage 111A). Once scanning system 101 determines that it is able to access cloud infrastructure 106, the process may return to Fig. 2A and step 203.

[0078] **Fig. 2C** is a schematic block diagram illustrating an exemplary embodiment of a process 220 for scanning/mapping, consistent with the disclosed embodiments. In some embodiments, the steps in Fig. 2C are executed using software and hardware of scanning system 101. In some embodiments, the steps in Fig. 2C may be performed in an order other than those depicted in Fig. 2C. In some embodiments, steps may also be omitted, repeated, or modified. In some embodiments, information gathered in one step of Fig. 2C may be used to provide context or other information for use in another step of Fig. 2C. In other embodiments, the steps in Fig. 2C may be executed by other devices.

[0079] Process 220 may begin with step 221 where scanning system 101 initiates a process to access keys, such as those keys stored in keystores 113A-113D. Scanning system 101 may utilize the string (e.g., a key or resource name) received in process 210 to authenticate in order to retrieve keys stored in keystores 113A-113D. Once authenticated, scanning system 101 may access one, some, or all keys stored in keystores 113A-113D. In some embodiments, step 221 may comprise utilizing pre-established trust relationships, such as AWS trust policies, instead of creating a new relationship.

[0080] In step 223, scanning system 101 may generate a “snapshot” of devices and systems in cloud infrastructure 106. In some embodiments, generating a snapshot may include reading devices and systems in cloud infrastructure, such as storage 111A-111D, databases 109A-109D, and virtual machines 107A-107D, and copying the information read from those devices and systems to storage at scanning system 101. Generating a snapshot, in some embodiments, may include recording a reference count to data blocks in one or more of storage 111A-111D, databases 109A-109D, and virtual machines 107A-107D, and copying each block them like a copy and write operation. In some embodiments, scanning system 101 may generate an Elastic Block Storage snapshot in step 223.

[0081] In some embodiments, scanning system 101 uses one or more keys retrieved from keystores 113A-113D to encrypt “snapshots” of cloud infrastructure.

[0082] In step 225, scanning system 101 may apply “tags” to the snapshot. This process may include, in some embodiments, adding information to the snapshot to identify the snapshot as being associated with scanning system 101. Scanning system 101, in some embodiments, may be configured to delete only snapshots with the associated tags.

[0083] In step 227, scanning system 101 may generate a map of cloud infrastructure 106. In some embodiments, this map may be in a form of graph: a plurality of interconnected vectors connecting the plurality of systems and devices, based on the networking configuration.

Scanning system 101, or other devices, may traverse the map to identify vectors originating in the Internet and reaching the devices and systems in cloud infrastructure 106. In some embodiments, generating the map may include enumerating Internet-accessible services that are capable of serving as an Internet proxy. In some embodiments, a user may control scanning system 101 (e.g., via user device 102) to display the map on a graphical user interface.

[0084] In some embodiments, generating the map may include enumerating properties of all assets, including: virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, load balancer 115, log files or databases, API gateway resources, API gateway REST APIs, Autoscaling groups, CloudTrail logs, CloudFront services, volumes, snapshots, VPCs, subnets, route tables, network ACLs, VPC endpoints, NAT gateways, ELB and ALB, ECR repositories, ECS clusters, services, and tasks, EKS, S3 bucket and Glacier storage, SNS topics, IAM roles, policies, groups, users, KMS keys, and Lambda functions. In some embodiments, generating the map in step 227 may further include analyzing devices or systems for a subset of risk related situations, including determinations of compromise situations (e.g., where an attacker has already gained access), imminent compromise situations (e.g., a known attack vector exists and can be used, such as a data store that is exposed to the public Internet without authentication), hazardous situations (e.g., a serious security implication, but no full attack vector exists), or informational situations (e.g., when storage 111A-111D has a limited amount of free space, or an unexploitable vulnerability exists).

[0085] Generating the map in step 227 may include recording information such a region identifier, site identifier, datacenter identifier, physical address, network address, workload name, or any other identifier which may be acquired via an API provided through a cloud service provider's system.

[0086] In step 229, scanning system 101 may provide alerts, e.g., to user device 102, indicating any situations found during the process of generating the map. For example, scanning system 101 may send one or more of an email, a popup alert, a text message, or other notification to user device 102.

[0087] After performing step 229, the process may return to Fig. 2A and step 205.

[0088] **Fig. 2D** is a schematic block diagram illustrating an exemplary embodiment of a process 230 for assessing, consistent with the disclosed embodiments. In some embodiments, the steps in Fig. 2D are executed using software and hardware of scanning system 101. In other embodiments, the steps in Fig. 2D may be executed by other devices. In some embodiments, scanning system 101 may execute the steps in Fig. 2D against a snapshot created in step 223 of Fig. 2C and stored at scanning system 101. In some embodiments, the steps in Fig. 2D may be performed in an order other than those depicted in Fig. 2D. In some embodiments, steps may also be omitted, repeated, or modified. In some embodiments, information gathered in one

step of Fig. 2D may be used to provide context or other information for use in another step of Fig. 2D.

[0089] Process 230 begins with step 231. In step 231, scanning system 101 may perform a step of vulnerability scanning. In some embodiments, step 231 comprises extracting everything in the snapshot, including operating system packages, installed software applications, libraries, and program language libraries such as Java archives, Python packages, Go modules, Ruby gems, PHP packages, and Node.js modules, or other software applications.

[0090] In some embodiments, step 231 may determine library versions, software versions, and other identifying characteristics of software and operating systems in the snapshot. Scanning system 101 may then try to match them to known vulnerabilities stored in a vulnerability database (e.g., one of databases 103A-103D). The vulnerability database, in some embodiments, may include vulnerability data from: NVD, WPVulnDB, US-CERT, Node.js Security Working Group, OVAL – Red Hat, Oracle Linux, Debian, Ubuntu, SUSE, Ruby Advisory Database, JVN, Safety DB(Python), Alpine secdb, PHP Security Advisories Database, Amazon ALAS, RustSec Advisory Database, Red Hat Security Advisories, Microsoft MSRC, KB, Debian Security Bug Tracker, Kubernetes security announcements, Exploit Database, Drupal security advisories, JPCERT. The vulnerability database may also include other vulnerability data (including, e.g., manually-added vulnerability data or vulnerability sources not listed above).

[0091] In step 233, scanning system 101 may perform a step of configuration scanning. In step 233, scanning system 101 may gather configuration information—such as a list of users of each system or device (e.g., virtual machines 107A-D), each system's or device's services, password hashes, and application-specific configurations for software/services such as Apache, Nginx, SSH, and other services. In some embodiments, scanning system 101 may perform a first analysis on all information collected in step 233 to remove sensitive information (e.g., social security numbers, passwords, birthdates) before proceeding to review the configuration information. In some embodiments, step 233 may comprise verifying adherence of the systems or devices in cloud infrastructure 106 to standards or benchmarks established by an external entity, such as the Center for Internet Security.

[0092] In some embodiments, scanning system 101 may perform a benchmarking process to detect misconfigurations of any services based on the information gathered in step 233. For example, scanning system 101 may determine the software version of each service and examine it against known vulnerabilities (e.g., stored in database 103A).

[0093] In some embodiments, scanning system 101 may determine bugs or other configuration risks that might only be exploitable from internal machines, because such bugs and risks can facilitate an attacker's lateral movement.

[0094] In some embodiments, scanning system 101 may evaluate network misconfigurations and their implications. Scanning system 101 may query devices and systems capable of routing traffic (e.g., load balancer 115, routers, switches, firewalls, and proxies) using an API provided through a cloud service provider's system to determine network configurations, and may evaluate them against known problematic configurations or other configurations.

[0095] In step 235, scanning system 101 may perform a step of malware scanning. In some embodiments, scanning system 101 may perform malware scanning across all filesystems in the snapshot (e.g., gathered from virtual machines 107A-107D or storage 111A-111D). Scanning system 101 may use multiple malware scanning software solutions to perform a malware scan against the filesystems, including one sourced from another vendor, such as bucketAV, Trend Micro Cloud One, Sophos Cloud Optix, Crowdstrike Falcon CWP, or others. In some embodiments, malware scanning in step 235 comprises utilizing signatures, heuristics, or sandboxing capabilities to deduce whether there is an infection on the machine.

[0096] In step 237, scanning system 101 may perform a step of lateral movement scanning. An attacker who establishes a network foothold usually attempts to move laterally from one resource to another in search of rich targets such as valuable data. Stolen passwords and keys unlock access to servers, files, and privileged accounts. In some embodiments, scanning system 101 may gather keys from each scanned system or device (e.g., virtual machines 107A-107D or storage 111A-111D). In some embodiments, scanning system 101 searches for passwords, scripts, shell history, repositories, or other data that may contain passwords, cloud access keys, SSH keys, or other key/password/access information that provide unchecked access to important resources. In some embodiments, scanning system 101 searches for such keys/passwords/access information and calculates a "hash" (a mathematical fingerprint) of each string. Scanning system 101 then attempts to match the hashed strings to hashes of strings that that are stored on different systems or devices. This will be used to detect the potential lateral movement between assets.

[0097] In step 239, scanning system 101 may perform a step of key/password scanning. As one example situation, suppose there is a weak or unprotected password stored (in plain text) in storage 111A. For example, if a personal email account has been compromised, the passwords may be known about in advance. Scanning system 101 may search the snapshot for similar usernames or login names, and, either using known dictionaries or the account owner's previously leaked passwords (stored in, e.g., database 103A), may attempt to login to one or more systems or devices in cloud infrastructure 106, and may record the result thereof.

[0098] In some embodiments, scanning system 101 may perform a "fuzzy search" on any usernames found in a password database (e.g., database 103A) to determine existing targets for password testing. In some embodiments, the fuzzy search uses the Damerau-Levenshtein edit distance algorithm.

[0099] As one example, scanning system 101 may determine that a leaked username includes the email address johndoe@acme.com. Scanning system 101 may try to match passwords from john.doe@acme.net, johndoe@gmail.com, and other variations. Scanning system 101 may perform a similar process against leaked passwords (e.g., if a leaked password is "Victory@19," scanning system may attempt to log in using "Victory@20."

[00100] In step 241, scanning system 101 may perform a step of sensitive information scanning. In some embodiments, scanning system 101 may search the snapshot for sensitive information, such as personally identifiable information (PII), Social Security numbers, healthcare information, or credit card numbers. In some embodiments, scanning system 101 may search data repository history as well. This is because it is not uncommon for an entire production environment repository to be cloned, with no one remembering the copy contains sensitive information. In some situations, detecting sensitive data not secured is critical in adherence to data privacy regulations.

[00101] To be certain that such alerts do not constitute false positives, in some embodiments, scanning system 101 may perform statistical scans against the data. For example, it is possible for a random number to resemble a Social Security number, yet it is extremely unlikely for the majority of a file with thousands of numbers to be valid Social Security Number by pure chance.

[00102] In step 243, scanning system 101 may perform a step of container scanning. In some embodiments, scanning system 101 may apply one or more of the preceding steps of Fig. 2D against containerized environments. In some embodiments, in order to do so, scanning system 101 reconstructs a container runtime layered file system (LFS) before recursively running one or more of steps 231-241 on the reconstructed file system.

[00103] Scanning system 101 may read network information from the snapshot in order to determine which services within which containers are exposed externally and within which ports they are accessible. For example, scanning system 101 may identify a port on which a vulnerable application is accessible based on known software vulnerabilities for a versions of software application. Scanning system 101 may query network accessibility information via an API provided through a cloud service provider's system and may use it to identify specific vulnerabilities susceptible to attack.

[00104] After performing step 243 on one or more containers, the process may return to Fig. 2A and step 207 (as discussed above).

[00105] **Foundational Techniques**

[00106] Aspects of this disclosure may include establishing a trusted relationship between a source account in a cloud environment and a scanner account. A trusted relationship, as used herein, may refer to a secure communication channel between at least two accounts (e.g., domains). The secure communications channel may include, for example, an

administration link, communication link, a connection or communication network configured with security protocols, or any other secure relationship between two networked entities.. A trusted relationship between two accounts may enable user accounts and global groups to be used in a domain other than the domain where the accounts are defined. A source account, as used herein, may refer to a domain, a location on a network server, or anything used to access system resources. A cloud environment, as used herein, may refer to a platform implemented on, hosted on, and/or accessing servers that may be reached via the Internet or other shared network. A scanner account, as used herein, may refer to any type of account associated with a scanner. A scanner, as used herein, may refer to a device for examining, reading, or monitoring electronically accessible information. Fig. 1 illustrates one example of a scanner, which is depicted in an exemplary manner as scanning system 101.

[00107] By way of example, establishing a trusted relationship between a source account in a cloud environment and a scanner account may include an integration process (creating a connection between an account on scanning system 101 and an account on cloud infrastructure 106) discussed with reference to Fig. 2A. By way of one example, scanning system 101 of Fig. 1 may create a trusted relationship between a source account in a cloud environment and a scanner account.

[00108] Aspects of this disclosure may include using the established trust relationship, utilizing at least one cloud provider API to identify workloads in the source account. A cloud provider API, as used herein, may refer to any application program interface that allows an end user to interact with a cloud provider's service. A cloud provider, may include any entity through whom information is accessible over the Internet or other shared network. Workloads, as used herein, may refer to systems, devices, or resources in a network (or available via a network) such as a cloud infrastructure. By way of example, Fig. 1 illustrates examples of a workload: virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115. In some embodiments, scanning system 101 of Fig. 1 may perform a scanning operation on workload (e.g., systems, devices, resources, etc.) in cloud infrastructure 106.

[00109] In some embodiments, scanning system 101 of Fig. 1 may use an API to detect workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115 in the source account.

[00110] Aspects of this disclosure may involve using the at least one cloud provider API to query a geographical location of at least one of the identified workloads. Querying, as used herein, may refer to a request for data or information from a resource. A geographical location, as used herein, may refer to the data center where this workload is served, a position on Earth, a location in a network, an address of a server or other hardware, or any other information capable of identifying a locus, region, or position. Internet geolocation involves software capable

of deducing a geographic position of a device connected to the Internet. For example, the device's IP address may be used to determine the country, city, or ZIP code, determining its geographical location. By way of example, scanning system 101 of Fig. 1 may query any of workloads virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106 for their geographical location using a cloud provider API.

[00111] In some embodiments, the geographic location may include an identifier of a physical site. An identifier, as used herein, may refer to anything that allows for recognition of an object, location, or resource. A physical site, as used herein, may refer to an actual location. By way of example, scanning system 101 of Fig. 1 may query any of workloads virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106 for an identifier of their physical site (e.g., address, postal code, or any other physical descriptor) using a cloud provider API.

[00112] In some embodiments, scanning system 101 of Fig. 1 may use an out-of-band process to reach cloud workloads through a runtime storage layer, in combination with information gathered from cloud provider APIs.

[00113] Aspects of this disclosure may include receive an identification of the geographic location. An identification of the geographic location, as used herein, may refer to a recognition, association, or classification of the geographic location. By way of example, scanning system 101 of Fig. 1 may receive a geographical location, after querying for it, by interrogating one or more workloads, such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, or keystores 113A-113D of cloud infrastructure 106. For example, the scanning system 101 of Fig. 1 may receive address, postal code, or any other physical descriptor of any workloads in cloud infrastructure 106. In some embodiments, scanning system 101 of Fig. 1 may receive a listing locating the workloads of cloud infrastructure 106 on a two-dimensional graphical representation.

[00114] In some embodiments, the identification of the geographic location may include an identification of a data center, at least one of a data center name, an Internet Protocol (IP) address, a name of the cloud provider, or a unique identity. A data center, as used herein, may refer to a facility that centralizes an organization's shared IT operations and equipment for the purposes of storing, processing, and/or disseminating data and applications. A data center, for example, may include a dedicated space within a building, or a group of buildings used to house computer systems and associated components, such as telecommunications and storage systems. An identification of a data center, as used herein, may refer to a recognition, association, or classification of a data center. A data center name, as used herein, may refer to a series of letters or numbers, a word, or set of words or numbers, an address, or a code identifying the data center. An Internet Protocol (IP) address, as used herein, may refer to a

unique string of characters that identifies a computer using the Internet Protocol to communicate over a network. A cloud provider, as used herein, may refer to a company that offers any components of cloud computing (e.g., infrastructure as a service (IaaS), software as a service (SaaS) or platform as a service (PaaS)). A unique identity, as used herein, may refer to a distinctive numeric or alphanumeric string that is associated with a single entity within a given system.

[00115] By way of example, scanning system 101 of Fig. 1 may query for and receive a geographical location such as a data center name/city and state, data center address, IP address / specific cloud identifier of the workload, name of cloud provider (e.g., Amazon Web Services), or a specific identifier of information of any of workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106.

[00116] In some embodiments, scanning system 101 of Fig. 1 may receive a listing indicating locations of the workloads of cloud infrastructure 106 on a two-dimensional graphical representation.

[00117] In some other embodiments, scanning system 101 may receive additional information (in addition to geographic location information) such as whether the workload is Internet-facing and easily accessible to attackers, or if it is private (and may be less critical).

[00118] Aspects of this disclosure may include using the cloud provider APIs to access block storage volumes of the at least one workload. Block storage volumes, as used herein, may refer to data files maintained on Storage Area Networks (SANs) or in cloud-based storage environments. Each storage volume may act as an individual hard drive configured by a storage administrator. Accessing block storage volumes, as used herein, may refer to obtaining, examining, or retrieving data.

[00119] As described with relation to process 210 of Fig. 2B (beginning with step 211 to initiate a connection to cloud infrastructure 106), scanning system 101 may send a second message to user device 102, instructing the user to generate a role. A role may refer to a type associated with a node that may be assigned or defined. The second message may include instructions for the user to follow to generate the role. In step 213, a user may provide (e.g., via a keyboard at user device 102) a role definition to the cloud service provider's system. In some embodiments, the role definition includes read-only permissions and permissions to read a block storage layer (containing block storage volumes). In some embodiments, scanning system 101 provides a role formation template (e.g., an Amazon Web Services CloudFormation Template) for use with cloud infrastructure 106 to create the necessary role. In step 213, the user may utilize user device 102, for example, by copying and pasting a URL of the template, downloading and uploading the template to the cloud service provider's system, or selecting the template from a list of templates.

[00120] By way of example, scanning system 101 of Fig. 1 may access the block storage volume of any of workloads virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106.

[00121] Aspects of this disclosure may include determining a file-system of the at least one workload. A file-system, as used herein, may refer to a method and/or a data structure that the operating system uses to control how data is stored and retrieved.

[00122] By way of example, scanning system 101 of Fig. 1 may identify the type of file-system of any of workloads virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106. Some file systems may include NTFS, FAT32, ext3, exFAT, HFS, HFS+, HPFC, UFS, ZFS.

[00123] Aspects of this disclosure may include mounting the block storage volumes on a scanner based on the determined file-system. Mounting the block storage volumes, as used herein, may refer to a process by which the operating system makes files and directories (such as block storage volumes) on a storage device available for users to access via the computer's file system. A scanner, as used herein, may refer to a device for examining, reading, or monitoring electronically accessible information. By way of example, Fig. 1 illustrates one example of a scanner, scanning system 101 of Fig. 1. In some embodiments, mounting may include selecting a driver corresponding to the determined file system. In some embodiments, to mount the block storage volumes on the scanner, may include creating a snapshot of the block storage volumes; and mounting the snapshot of the block storage volumes on the scanner. A snapshot of the block storage volumes, as used herein, may refer to a current state or earlier state of the block storage volumes.

[00124] In some embodiments, the at least one processor may be further configured to encrypt the snapshot of the block storage volumes and mount the encrypted snapshot of the block storage volumes on the scanner. Encrypting a snapshot, as used herein, may refer to a process of encoding information (such as a snapshot). This process may convert the original representation of the information, known as plaintext, into an alternative form known as ciphertext.

[00125] In some embodiments, the scanner may use a privileged account to log in and determine how secure each host is from an inside vantage point. While authenticated scans can successfully discover potential vulnerabilities, they may be limited to the extent they require a privileged account on each scanned host. Furthermore, scans use significant system resources during the test procedures and require opening ports that by themselves pose a security risk.

[00126] Aspects of this disclosure may include activating a scanner at the geographic location. Activating a scanner, as used herein, may refer to starting up or initiating a device for examining, reading, or monitoring data. By way of example, scanning system 101 of Fig. 1 may

start up a scanner (a different scanner than scanning system 101) at the geographic location identified by scanning system 101.

[00127] Aspects of this disclosure may include reconstructing from the block storage volumes a state of the workload. Reconstructing, as used herein, may refer to rebuilding or reforming a state of the workload. A state of the workload, as used herein, may refer to a condition of a workload at a certain time. By way of example, scanning system 101 of Fig. 1 may rebuild the block storage volumes as an earlier version of any of workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106.

[00128] In some embodiments, the reconstructed state of the workload may include at least two of an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, a profile configuration, a log, or a database content. An indication of an installed application, as used herein, may refer to an identifier of a software. A version of an installed application, as used herein, may refer to a particular form, variant, edition, or revision of a piece or pieces of software. An operating system configuration, as used herein, may refer to a manner in which components are arranged to make up an operating system or a computer system. An application configuration, as used herein, may refer to a manner in which components are arranged to make up an application or a computer system. A profile configuration, as used herein, may refer to a manner in which components are arranged to make up a profile or a computer system. A log, as used herein, may refer to a detailed list of an application information, system performance, or user activities. Database content, as used herein, may refer to any collection of data, or information, that is specially organized for rapid search and retrieval by a computer. In some embodiments, scanning system 101 of Fig. 1 may reconstruct any workload, such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106 from the block storage volumes to an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, a profile configuration, a log, or a database content.

[00129] Aspects of this disclosure may include assessing the reconstructed state of the workload to extract insights. Assessing the reconstructed state of the workload, as used herein, may refer viewing, utilizing, or evaluating the reconstructed state of the workload. Insights, as used herein, may refer to information or knowledge attained based on an analysis of some data. Extracting insights, as used herein, may refer to pulling information or data about, for example, a vulnerability associated with the workload or a composition of installed applications associated with the workload. By way of example, scanning system 101 of Fig. 1 may evaluate the reconstructed state of any of workloads, such as virtual machines 107A-107D, databases 109A-

109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106 to pull valuable data and information.

[00130] In some embodiments, insights may include at least one of a vulnerability associated with the workload or a composition of installed applications associated with the workload. A vulnerability, as used herein, may refer to any weakness within an organization's information systems, internal controls, or system processes that can be exploited. A composition of installed applications associated with the workload, as used herein, may refer to a structure or group of accessible computer programs. By way of example, scanning system 101 of Fig. 1 may evaluate the reconstructed state of any workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106 to pull valuable data and information regarding a vulnerability with the workload.

[00131] In some embodiments, a scanner may be deployed at the geographical location. Deploying a scanner, as used herein, may refer to installing, activating, utilizing, or bringing a scanner into effective action . By way of example, Fig. 1 illustrates one example of a scanner, scanning system 101 of Fig. 1 that is deployed.

[00132] In some embodiments, scanning system 101 of Fig. 1 may update the reconstructed state of the workload based on at least one change to the block storage volumes . Updating the reconstructed state of the workload, as used herein, may refer to overhauling or refurbishing the reconstructed state of the workload. A change to the block storage volumes, as used herein, may refer to any shift, alteration, or modification to the block storage volumes.

[00133] **Fig. 3** illustrates a block diagram of method 300 for insight extraction, consistent with disclosed embodiments. In some embodiments, the method may include ten (or more or less) steps:

[00134] Block 302: Establish a trusted relationship between a source account in a cloud environment and a scanner account. In some embodiments, establishing a trusted relationship between a source account in a cloud environment and a scanner account may include an integration process (creating a connection between an account on scanning system 101 and an account on cloud infrastructure 106) discussed with reference to Fig. 2A. By way of one example, scanning system 101 of Fig. 1 may create a trusted relationship between a source account in a cloud environment and a scanner account.

[00135] Block 304: Using the established trust relationship, utilize at least one cloud provider API to identify workloads in the source account. In some embodiments, scanning system 101 of Fig. 1 may use an API to detect workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115 in the source account.

[00136] Block 306: Use the at least one cloud provider API to query a geographical location of at least one of the identified workloads. In some embodiments, scanning system 101 of Fig. 1 may query any of workloads virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106 for their geographical location using a cloud provider API.

[00137] Block 308: Receive an identification of the geographic location. In some embodiments, scanning system 101 may indicate in the map additional information regarding the workload (e.g., a system or device). For example, scanning system 101 may provide: an identification of a data center (e.g., a name or number associated with the data center where the workload is located), at least one of a data center name (e.g., a name or address associated with the data center where the workload is located), Internet Protocol (IP) address, name of the cloud provider, or a unique identity (e.g., any other information related to the workload that may be used).

[00138] Block 310: Use the cloud provider APIs to access block storage volumes of the at least one workload. In some embodiments, scanning system 101 of Fig. 1 may access the block storage volume of workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106. In some embodiments, accessing the block storage may include taking the mean of generating snapshots.

[00139] Block 312: Determine a file-system of the at least one workload. In some embodiments, scanning system 101 of Fig. 1 may identify the type of file-system any of workloads (virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106).

[00140] Block 314: Mount the block storage volumes on a scanner based on the determined file-system. In some embodiments, to mount the block storage volumes on the scanner, may include instructing the operating system to map a directory structure logically to a physical storage device. A storage volume may be mounted after it is attached and formatted for use by a server's operating system. Mounting may include creating a snapshot of the block storage..

[00141] Block 316: Activate a scanner at the geographic location. In some embodiments, scanning system 101 of Fig. 1 may start up a scanner (a different scanner than scanning system 101) at the geographic location identified by scanning system 101.

[00142] Block 318: Reconstruct from the block storage volumes a state of the workload. In some embodiments, scanning system 101 of Fig. 1 may rebuild the block storage volumes to an earlier version of workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106.

[00143] Block 320: Assess the reconstructed state of the workload to extract insights. In some embodiments, scanning system 101 of Fig. 1 may evaluate the reconstructed state of workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D of cloud infrastructure 106 to derive valuable data and information.

[00144] **Vulnerability Management Techniques**

[00145] **Fig. 4** depicts a cybersecurity system performing a side scanning function to protect against potential vulnerabilities. Referring to Fig. 4, a processor 400 configured to identify in a cloud environment one or more block storage volumes 401A-C communicatively connected to a cloud provider API where the block storage volumes 401A-C may be contained. When a block storage volume, e.g., 401A is identified, the processor 400 may be configured to perform the function of identifying the software installed 403A-C in or relating to block storage volume 401. Upon identifying installed software 403A-C, the processor may be further configured to identify the version of software 405A-C operating in the installed software 403A-C. Based on the installed software 403A-C and the version of the software 405A-C, the processor may be configured to identify a listing of known vulnerabilities 407A-C that may relate to the installed software 403A-C and the version installed 405A-C. When the listing of known vulnerabilities 407 is obtained, the system may list the known vulnerabilities 409A-C for that software version 411A-C, which can be communicated to the processor 400 of the end user. This information may be processed and displayed to an end user using processor 400.

[00146] **Fig. 5** of the disclosed embodiments describes a method of operating a cybersecurity system performing a side scanning function to protect against potential vulnerabilities. Referring to Fig. 5, a processor unit 500 may be configured to perform the method. When initiated either manually or automatically, processor unit 500 may be communicatively connected with a cloud provider API 501 to initiate a cybersecurity function as disclosed above. Contained digitally on a memory device and accessed by a cloud provider API 501 is a series of block storage volumes 503. Processor unit 500 may be capable of performing a vertical and horizontal scan for network access or security information of a block storage volume 503 to detect potential vulnerabilities to the block storage volume 503 or any processing unit 500 designed to access said block storage volume 503.

[00147] Based on the accessed block storage volume 500, processor unit 500 with the disclosed embodiments of a cybersecurity system may perform an identification of the type of software installed 505 connected with block storage volume 503. Following identification of the type of software installed 505, the processing unit may then identify an installed version 507 of said installed software 505 for a version comprised of a unique identifier based on a combination of letters, numbers, or similar unique identifiers.

[00148] Upon recognition of an installed software version 507 of the installed software 505, processing unit 500 with the disclosed embodiments of the disclosed scanning system 101

may provide a list of known vulnerabilities 509 for review by said scanning system 101 and/or its end user and maintainer of said block storage volumes 503. Based on the list of known vulnerabilities 509, the disclosed scanning system 101 or its end user may identify one or more ports of accessibility to said block storage volume 503 that may be accessed by a known, associated vulnerability 513 from listed vulnerabilities 509 to determine an avenue for potential vulnerability 513 to access and infiltrate block storage volume 503.

[00149] In some embodiments, a cyber security scanning system for a cloud environment 101 may include a processor 500 to operate said system. This processor 500 may include central processing units and other similar computer-enabling equipment for processing and executing commands based on the information inputted to said system. The processor 500 may be communicatively connected to a computer network or series of networks to accomplish said cyber security function.

[00150] As an example embodiment, a processor unit 500 may be configured to use a cloud-provider API that may communicate with one or more specified computer-readable media across a digital network. This can be accomplished through internet protocols, internet control message protocols, transmission control protocol, or user datagram protocol. Cloud provider API 501 may be one of several forms of middleware, interface, middle layer, or other systems of interfacing applications. A processor 300 may be one or more computer processing units, central processing unit, server, microcomputer, mainframe, and any other manifestation of digital computing.

[00151] Further to one of several possible embodiments, a cloud-provider API 501 may be configured to access a block-storage volume 503 of a workload maintained in a cloud-storage environment. This may be accomplished through a system of computer-readable media communicatively connected. Said block-storage volume 303 may be contained on a Storage Area Network (SAN) or similar cloud-based memory storage environment. The block storage volume 503 may be contained in smaller storage volumes with an associated identifier unique to that portion of said block storage volume 503. In some embodiments, the block-storage volume 503 of a workload may have multiple paths for the storage volume to be reaggregated and retrieved quickly.

[00152] Among several embodiments, a scanning system 101 may comprise a system for identifying an installed software application in the accessed block-storage volume 503. This identification of installed software may be accomplished by accessing installed software 505 files through signature verification, root license, or authorized user lists. The installed software 505 may be located and identified within applications such as file storage, database storage, and virtual machine file system volumes. The identification of said installed software application 507 may be processed, analyzed, and communicated to the scanning system 101 for

processing, cataloging, and protection through encryption and various methods of layered cyber defense.

[00153] Further, the scanning system 101 described herein may include functionality to analyze installed software applications to determine the associated software version. The software application version 507 may identify the software version based on unique version name, unique version number, and may be based on unique states of the currently installed computer software 505.

[00154] One of many embodiments disclosed above may include the scanning system 101 having the ability to access a data structure of known software vulnerabilities 509 for a plurality of versions of software applications. The known software vulnerabilities 509 may include, among others, missing data encryption, OS command injection, SQL injection, buffer overflow, missing authentication, missing authorization, unrestricted upload of dangerous file types, reliance on untrusted inputs in a security decision, cross-site scripting and forgery, download of codes without integrity checks, broken algorithms, URL redirection, path traversal, software bugs, weak passwords, and previously infected software. The scanning system 101 may be able to access and identify software vulnerabilities for mitigation, rectification, correction, and fortification.

[00155] In one embodiment, a cybersecurity system may also perform scanning according to scanning system 101 by performing a lookup of the identified installed software version 507 in the data structure to identify known vulnerabilities 509. This function can be performed by the scanning system 101 according to Fig. 1 performing a query of the installed software 505 for unique version number or designator and comparing to, amongst many things, a set of likely or potential vulnerabilities to that software version for potential deficiencies or cybersecurity threats known or suspected to similar software types and versions. This query may be performed according to a predetermined set of values, to include previously identified unique version numbers or designators that may contain the known list of previously identified vulnerabilities.

[00156] Among many embodiments, embodiments of the disclosed scanning system 101 may query the cloud provider API 501 to determine network accessibility information 511 related to the workload. In order to accomplish this query of the cloud provider API 301, the scanning system 101 may involve index of search results and display of said search results, followed by processing and grouping search results. Network accessibility information 511 may include connection quality, alternative paths between nodes in a network, and the ability to avoid blockage in said networks. The workloads associated with this query may include applications, services, capabilities, and specific processes such as virtual machines, databases, containers, or Hadoop nodes, among others.

[00157] If the system detects a vulnerable application 513, one embodiment may identify one or more port on which said vulnerable application is accessible. In one of several embodiments, the scanning system 101 may detect a vulnerable application in one or more computation process. In another embodiment, the cybersecurity system may perform a network accessibility query in a separate process. Further, a disclosed embodiment may perform these separate functions in subsequent and sequential steps of the same process. A person having skill would understand that an authorized user or an authorized scanning system 101 can perform these functions concurrently and subsequently by an authorized user or an authorized cybersecurity system while performing the same function as the disclosed embodiment.

[00158] Upon gathering network accessibility information and the identified port to identify one or more vulnerabilities susceptible to attack from outside the workload, a disclosed embodiment would have the functionality to perform processes to gather, display, and mitigate a discovered vulnerability in order to minimize the likelihood and effectiveness of a cyber threat outside of and attempting to access a workload through a known or previously encountered type of cyber threat. This functionality may include collecting and organizing the vulnerabilities according to type or category of vulnerability, displaying the gathered data for an end user or maintainer, and implementing security features automatically or manually by a user or maintainer such as security patches, password or passcode changes and suggestions for users to do the same, and malicious code eradication.

[00159] As one of several possible embodiments, the scanning system 101 may also, upon identification of one or more vulnerabilities, implement remedial actions via one or more processors. Remedial actions may include, among other things, notification to an end user of an identified threat, compensation through a revised security code to mitigate the potential threat, publication of the identified threat and vulnerability in a log or record of detected vulnerabilities, and communication of the sensed vulnerability and threat to a server operator or maintainer to fortify the protections of workloads existing on similar environments.

[00160] In a similar embodiment to above, the remedial measure may include transmission of an alert to a device associated with an administrator. The alert may be, amongst others, written, auditory, and visual for processing and use by an administrator of said scanning system 101. Said administrator may take action based on the received alert, to include eliminating the cyber threat through mitigation measures, change in cybersecurity posture, or removing the workload from the cybersecurity threat environment.

[00161] In another embodiment, a query of the cloud provider API 501 to determine network accessibility information related to the workload may be performed by at least one processor 500 configured to examine data sources associated with the workload. The data examined may include user data, system processing data, accessibility data, clock cycles, storage input/output, or similar data processors. A query of the cloud provider API 501 may be

automated or manually-initiated. Based on the said query, the network accessibility information 511 related to the workload may change based on the data sources associated with said workload.

[00162] In the above processor and similar embodiments, further configuration may include a process to determine network accessibility information 511 based on the examined data sources. The examined data sources may include various cloud-based workloads, internet protocols, transmission communication protocols, or other methods and systems of memory and data storage.

[00163] As an exemplary embodiment, network accessibility information 511 includes at least one of: data from an external data source, cloud provider information, or at least one network capture log. These embodiments of an external data source may include data from the operating environment of the cloud-based environment, an external operating system for a computer processing unit 500, or other similar computer readable media. Cloud provider information may further include information that may identify the network accessibility information 511 vertically or horizontally to fully describe the associated workload. Network capture logs may be automated or manually updated to include possible vulnerabilities 513 and threats to the cloud-based storage medium.

[00164] Further, the disclosed embodiments may include an installed software application 505, with the at least one processor 500 configured to extract data from at least one of operating system packages, libraries, or program language libraries. This data may be extracted through a system query, random access algorithm, or similar automated process. Operating system packages may include systems operable on Microsoft, Apple, Linux, and similar operating systems. Libraries may consist of a series of files, folders, and databases of information stored on one of any indexed data repositories. A program language library may contain several of an exemplary program languages including but not limited to Javascript, Swift, Scala, Go, Python, Elm, Ruby, C#, C++ and other similar sources of software code.

[00165] As another exemplary embodiment, a scanning system 101 may also include a processor 500 configured to identify installed software application 505 based on the extracted data. The processor 500 may perform only this function or this function among many to accomplish the layered cybersecurity defense described herein this disclosed cloud-based security environment. The identification of installed software by said processor may include identifying the software by unique version number or designator, recognized source code, metadata associated with the installed software application files, or similar software-identifying information.

[00166] One embodiment may include an additional function wherein the installed software application 505 that has been identified includes one or more scripts. These scripts

may be processed through various computer readable languages to include Javascript, C#, C++ and other forms of computer code.

[00167] One embodiment contemplated by the disclosed scanning system 101 may also include data structure includes aggregated vulnerability data 309. This aggregated vulnerability data 509 may be compiled by an end user or maintainer from within the cloud-based environment of the current data structure as well as compilation from similar data sets and aggregation of common threats to data structures likely to experience similar vulnerabilities. This aggregation of vulnerability data 509 may be contained within the data structure and it may be collectively aggregated to provide for a more robust and layered cybersecurity defense posture.

[00168] In an embodiment of the disclosed scanning system 101, the aggregated vulnerability data 509 may include data from one or more third-party vendors. These vendors may include operators of the cloud-based server environment, providers of networking and internet communication, methods of layered authentication, and other similar providers of services directly related and in communication with the cloud-based cybersecurity system.

[00169] As an additional exemplary embodiment, the aggregated vulnerability data 309 may include data collected by a scanner. This scanner may involve use of continuous or periodic monitoring of the workload. The scanner may perform security screenings of the various workloads vertically or horizontally to identify network identification information, port accessibility, and associated vulnerabilities. Any scan performed may be communicated to the scanning system 101 that may be responsible for performing and logging the results of the scan and may be able to initiate follow-on processes and protocols to protect the data contained in the workload that is the subject of scanning.

[00170] An embodiment of the disclosed scanning system 101 may also include aggregated vulnerability data 509 that may include at least one of an advisory, an exploit, a security announcement, or a known bug. An advisory may include notification to a system maintainer or user of the potential vulnerability, may log notice of the advisory, and may recommend possible user or maintainer actions to potentially address said advisory. An exploit may further consist of an automated system response designed to take advantage of the sensed vulnerability data. The exploit can be further reflected in the aggregated vulnerability data and protocols can be written into the cybersecurity infrastructure to prevent said exploit from gaining access and permissions to unauthorized areas of the workload storage environment.

[00171] A person having ordinary skill in the art would appreciate the above described embodiments are among many potential embodiments, to include a method of operating a scanning system 101 similar to the one described above. A disclosed embodiment contemplates this method to be accomplished through manual user operation, automated

computer processes, or similar manners of operation. These manners of operation and those contemplated similar to them would allow the scanning system 101 described and disclosed to execute its operations as the system above describes.

[00172] One of several embodiments of the disclosed method of operating a scanning system 101 may include a cyber security scanning system for a cloud environment 101 may include a processor 500 to operate said system. This processor may include central processing units and other similar computer-enabling equipment for processing and executing commands based on the information inputted to said system. The processor may be communicatively connected to a computer network or series of networks to accomplish said cyber security function.

[00173] As an example embodiment, a processor unit 500 may be configured to use a cloud-provider API 501 that may communicate with one or more specified computer-readable media across a digital network. This can be accomplished through internet protocols, internet control message protocols, transmission control protocol, or user datagram protocol. Cloud provider API 501 may be one of several forms of middleware, interface, middle layer, or other systems of interfacing applications. A processor may be one or more computer processing units, central processing unit, server, microcomputer, mainframe, and any other manifestation of digital computing.

[00174] Further to one of several possible embodiments, a cloud-provider API 501 may be configured to access a block-storage volume 503 of a workload maintained in a cloud-storage environment. This may be accomplished through a system of computer-readable media communicatively connected. Said block-storage volume 503 may be contained on a Storage Area Network (SAN) or similar cloud-based memory storage environment. The block storage volume 503 may be contained in smaller storage volumes with an associated identifier unique to that portion of said block storage volume 503. In some embodiments, the block-storage volume 503 of a workload may have multiple paths for the storage volume to be reaggregated and retrieved quickly.

[00175] Among several embodiments, a method of operating a scanning system 101 may comprise a system for identifying an installed software application 505 in the accessed block-storage volume 503. This identification of installed software 505 may be accomplished by accessing installed software files through signature verification, root license, or authorized user lists. The installed software 505 may be located and identified within applications such as file storage, database storage, and virtual machine file system volumes. The identification of said installed software application 505 may be processed, analyzed, and communicated to the method of operating a scanning system 101 for processing, cataloging, and protection through encryption and various methods of layered cyber defense.

[00176] Further, a method of operating a scanning system 101 described herein may include functionality to analyze installed software applications 505 to determine the associated software version 507. The software application version 507 may identify the software version based on unique version name, unique version number, and may be based on unique states of the currently installed computer software.

[00177] One of many embodiments disclosed above may include the method of operating a scanning system 101 having the ability to access a data structure of known software vulnerabilities 509 for a plurality of versions of software applications 507. The known software vulnerabilities 509 may include, among others, missing data encryption, OS command injection, SQL injection, buffer overflow, missing authentication, missing authorization, unrestricted upload of dangerous file types, reliance on untrusted inputs in a security decision, cross-site scripting and forgery, download of codes without integrity checks, broken algorithms, URL redirection, path traversal, software bugs, weak passwords, and previously infected software. The cyber security system may be able to access and identify software vulnerabilities 309 for mitigation, rectification, correction, and fortification.

[00178] In one embodiment, a method of operating a scanning system 101 may also perform scanning according to scanning system 101 by performing a lookup of the identified installed software version 507 in the data structure to identify known vulnerabilities 309. This function can be performed by the scanning system 101 according to Fig. 1 performing a query of the installed software 505 for unique version number 507 or designator and comparing to, amongst many things, a set of likely or potential vulnerabilities to that software version for potential deficiencies or cybersecurity threats known or suspected to similar software types and versions. This query may be performed according to a predetermined set of values, to include previously identified unique version numbers or designators that may contain the known list of previously identified vulnerabilities.

[00179] Among many embodiments, embodiments of the disclosed method of operating a scanning system 101 may query the cloud provider API 501 to determine network accessibility information 511 related to the workload. In order to accomplish this query of the cloud provider API 501, the scanning system 101 may involve index of search results and display of said search results, followed by processing and grouping search results. Network accessibility information 511 may include connection quality, alternative paths between nodes in a network, and the ability to avoid blockage in said networks. The workloads associated with this query may include applications, services, capabilities, and specific processes such as virtual machines, databases, containers, or Hadoop nodes, among others.

[00180] If the system detects a vulnerable application, one embodiment may identify one or more port on which said vulnerable application is accessible. In one of several embodiments, the scanning system 101 may detect a vulnerable application in one or more computation

process. In another embodiment, the scanning system 101 may perform a network accessibility query in a separate process. Further, a disclosed embodiment may perform these separate functions in subsequent and sequential steps of the same process. A person having skill would understand that an authorized user or an authorized cybersecurity system may perform these functions concurrently and subsequently while performing the same function as the disclosed embodiment.

[00181] Upon gathering network accessibility information 511 and the identified port 513 to identify one or more vulnerabilities susceptible to attack from outside the workload, a disclosed embodiment would have the functionality to perform processes to gather, display, and mitigate a discovered vulnerability in order to minimize the likelihood and effectiveness of a cyber threat outside of and attempting to access a workload through a known or previously encountered type of cyber threat. This functionality may include collecting and organizing the vulnerabilities according to type or category of vulnerability, displaying the gathered data for an end user or maintainer, and implementing security features automatically or manually by a user or maintainer such as security patches, password or passcode changes and suggestions for users to do the same, and malicious code eradication.

[00182] A person having ordinary skill in the art would similarly understand that the above disclosed system can be disclosed using a suitable non-transitory computer-readable medium performing each of the disclosed functions. The disclosed embodiment may include a non-transitory computer readable medium with a scanning system 101 similar to the one described above on a non-transitory computer-readable medium. A disclosed embodiment contemplates this system to be accomplished through a medium that may contain a central processing unit, virtual machine, or a similar non-transitory medium. These manners of operation and those contemplated similar to them would allow the scanning system 101 described and disclosed to execute its operations as the system above describes.

[00183] One of several embodiments of the disclosed non-transitory computer readable medium with a scanning system 101 may include a cyber security scanning system for a cloud environment 101 may include a processor 500 to operate said system. This processor 500 may include central processing units and other similar computer-enabling equipment for processing and executing commands based on the information inputted to said system. The processor 500 may be communicatively connected to a computer network or series of networks to accomplish said cyber security function.

[00184] As an example embodiment, a processor unit 300 may be configured to use a cloud-provider API 501 that may communicate with one or more specified computer-readable media across a digital network. This can be accomplished through internet protocols, internet control message protocols, transmission control protocol, or user datagram protocol. Cloud provider API 501 may be one of several forms of middleware, interface, middle layer, or other

systems of interfacing applications. A processor 500 may be one or more computer processing units, central processing unit, server, microcomputer, mainframe, and any other manifestation of digital computing.

[00185] Further to one of several possible embodiments, a cloud-provider API 501 may be configured to access a block-storage volume 503 of a workload maintained in a cloud-storage environment. This may be accomplished through a system of computer-readable media communicatively connected. Said block-storage volume 503 may be contained on a Storage Area Network (SAN) or similar cloud-based memory storage environment. The block storage volume 503 may be contained in smaller storage volumes with an associated identifier unique to that portion of said block storage volume 503. In some embodiments, the block-storage volume 503 of a workload may have multiple paths for the storage volume to be reaggregated and retrieved quickly.

[00186] Among several embodiments, a non-transitory computer readable medium with a scanning system 101 may comprise a system for identifying an installed software application 505 in the accessed block-storage volume 503. This identification of installed software 505 may be accomplished by accessing installed software files through signature verification, root license, or authorized user lists. The installed software 505 may be located and identified within applications such as file storage, database storage, and virtual machine file system volumes. The identification of said installed software application 505 may be processed, analyzed, and communicated to the method of operating a scanning system 101 for processing, cataloging, and protection through encryption and various methods of layered cyber defense.

[00187] Further, a non-transitory computer readable medium with a scanning system 101 described herein may include functionality to analyze installed software applications 505 to determine the associated software version 507. The software application version 507 may identify the software version based on unique version name, unique version number, and may be based on unique states of the currently installed computer software.

[00188] One of many embodiments disclosed above may include the non-transitory computer readable medium with a scanning system 101 having the ability to access a data structure of known software vulnerabilities 509 for a plurality of versions of software applications 507. The known software vulnerabilities 509 may include, among others, missing data encryption, OS command injection, SQL injection, buffer overflow, missing authentication, missing authorization, unrestricted upload of dangerous file types, reliance on untrusted inputs in a security decision, cross-site scripting and forgery, download of codes without integrity checks, broken algorithms, URL redirection, path traversal, software bugs, weak passwords, and previously infected software. The scanning system 101 may be able to access and identify software vulnerabilities for mitigation, rectification, correction, and fortification.

[00189] In one embodiment, a non-transitory computer readable medium with a scanning system 101 may also perform scanning according to scanning system 101 by performing a lookup of the identified installed software version in the data structure to identify known vulnerabilities 509. This function can be performed by the scanning system 101 according to Fig. 1 performing a query of the installed software for unique version number 307or designator and comparing to, amongst many things, a set of likely or potential vulnerabilities 309 to that software version 307 for potential deficiencies or cybersecurity threats known or suspected to similar software types and versions. This query may be performed according to a predetermined set of values, to include previously identified unique version numbers or designators that may contain the known list of previously identified vulnerabilities 509.

[00190] Among many embodiments, embodiments of the disclosed non-transitory computer readable medium with a scanning system 101 may query the cloud provider API 501 to determine network accessibility information 511 related to the workload. In order to accomplish this query of the cloud provider API 501, the scanning system 101 may involve index of search results and display of said search results, followed by processing and grouping search results. Network accessibility information 511 may include connection quality, alternative paths between nodes in a network, and the ability to avoid blockage in said networks. The workloads associated with this query may include applications, services, capabilities, and specific processes such as virtual machines, databases, containers, or Hadoop nodes, among others.

[00191] If the system detects a vulnerable application, one embodiment may identify one or more ports 515 on which said vulnerable application is accessible. In one of several embodiments, the scanning system 101 may detect a vulnerable application in one or more computation processes. In another embodiment, the scanning system 101 may perform a network accessibility query in a separate process. Further, a disclosed embodiment may perform these separate functions in subsequent and sequential steps of the same process. A person having skill would understand that an authorized user or an authorized cybersecurity system can perform these functions concurrently and subsequently while performing the same function as the disclosed embodiment.

[00192] Upon gathering network accessibility information 511 and the identified port 515 to identify one or more vulnerabilities susceptible to attack from outside the workload, a disclosed embodiment would have the functionality to perform processes to gather, display, and mitigate a discovered vulnerability in order to minimize the likelihood and effectiveness of a cyber threat outside of and attempting to access a workload through a known or previously encountered type of cyber threat. This functionality may include collecting and organizing the vulnerabilities according to type or category of vulnerability, displaying the gathered data for an end user or maintainer, and implementing security features automatically or manually by a user

or maintainer such as security patches, password or passcode changes and suggestions for users to do the same, and malicious code eradication.

[00193] Another disclosed embodiment may include a method of using a cloud provider API 501, accessing block storage volume 503 of a workload maintained in a cloud storage environment. The method may include manual user operation, automated system operation, systematic and random operating parameters for the system to provide its cybersecurity and similar security functions. The accessed block storage volume may be contained as a collection of block units organized together, may be a series of individual blocks that can be reorganized to form a new storage volume, and may have the ability to be disaggregated and reaggregated as necessary to accomplish its storage and cybersecurity functions.

[00194] The disclosed method may further comprise a system that may analyze the identified installed software application 505 to determine an associated software version 507. In analyzing an installed software application 305, the disclosed embodiment may include a method of querying the installed software application to access a software version 507 that may include a series of letters, numbers, and other identifying characters to differentiate the version of software currently operative and its associated identifying characteristics and capabilities.

[00195] A method of the disclosed may also include accessing a data structure of known software vulnerabilities 509 for a plurality of versions of software applications 507. The disclosed method may determine a data structure of known software vulnerabilities 509 of one among many operative versions of software installed in the monitored storage system as compared to historical data of similar storage systems.

[00196] The disclosed method as described herein may further include performing a lookup of the identified installed software version 507 in the data structure to identify known vulnerabilities 509. The lookup of identified installed software 507 may be previously indexed based on unique version identifier, and the indexed information may include known vulnerability information 509, network accessibility information 511, network protocols, and administrator identifier information. Based on the said indexed known vulnerabilities 509, the scanning system 101 disclosed may take actions to mitigate a cybersecurity threat and reinforce cybersecurity defenses at the type of cybersecurity threat, as previously discussed.

[00197] Further to the above, a disclosed method may also include querying the cloud provider API 501 to determine network accessibility information 511 related to the workload. A query of the cloud provider API 501 to determine network accessibility information 511 related to the workload may be automatically or manually initiated. Network accessibility information 511 similar to the above embodiment may include connection quality, alternative paths between nodes in a network, and the ability to avoid blockage in said networks. The workloads associated with this query may include applications, services, capabilities, and specific processes such as virtual machines, databases, containers, and Hadoop nodes, among others.

[00198] A disclosed method of one embodiment also may provide a method of identifying at least one port 515 on which the vulnerable application is accessible. The identification of one port 515 on which the vulnerable application is accessible may be for the purpose of identifying a cyber weakness, and it may include the ability to change the port currently accessible to mitigate the said cyber weakness.

[00199] Yet another disclosed embodiment of a disclosed method may include using the network accessibility information 511 and the identified at least one port 515 to identify one or more vulnerabilities 513 susceptible to attack from outside the workload. The vulnerability 513 identified as susceptible to attack from outside the workload may include, among others, missing data encryption, OS command injection, SQL injection, buffer overflow, missing authentication, missing authorization, unrestricted upload of dangerous file types, reliance on untrusted inputs in a security decision, cross-site scripting and forgery, download of codes without integrity checks, broken algorithms, URL redirection, path traversal, software bugs, weak passwords, and previously infected software.

[00200] A method of the disclosed embodiment may further include implementing a remedial action in response to the identified one or more vulnerabilities 513. Said remedial action may include, among other things, notification to an end user of an identified threat, compensation through a cybersecurity patch, publish of the identified threat and vulnerability in a log or record of detected vulnerabilities, and communication of the sensed vulnerability and threat to a server operator and maintainer to fortify the protections of workloads existing on similar environments.

[00201] The disclosed method may also include wherein the remedial measure includes transmitting an alert to a device associated with an administrator. Said alert may be, amongst others, written, auditory, and visual for processing and use by an administrator of said cybersecurity system. Said alert may further be logged and catalogued for future identification of known threats and vulnerabilities to similar software application versions.

[00202] A disclosed method of the present embodiment may also provide querying the cloud provider API 501 to determine the network accessibility information 311 related to the workload and examining data sources associated with the workload. A query of the cloud provider API 501 to determine network accessibility information 511 related to the workload may be automatically or manually initiated. Network accessibility information 511 similar to the above embodiment may include connection quality, alternative paths between nodes in a network, and the ability to avoid blockage in said networks. The workloads associated with this query may include applications, services, capabilities, and specific processes such as virtual machines, databases, containers, and Hadoop nodes, among others.

[00203] The method of the above disclosed embodiment may further determine network accessibility information 511 by determining the network accessibility information 511 based on

the examined data sources. The data examined may include user data, system processing data, accessibility data, clock cycles, storage input/output, and similar processes. A query of the cloud provider API 501 may be automated or manually-initiated. Based on the said query, the network accessibility information 511 related to the workload may change based on the data sources associated with said workload.

[00204] In the method of one disclosed embodiment, network accessibility information 511 may include at least one of data from an external data source, cloud provider information, or at least one network capture log. These embodiments may include data from the operating environment of the cloud-based environment, an external operating system for a computer processing unit, or other similar computer readable media. Network capture logs may be automated or manually updated to include possible vulnerabilities and threats to the cloud-based storage medium.

[00205] A method of the disclosed embodiment may also include identifying the installed software application 507 by extracting data from at least one of OS packages, libraries, or program language libraries. This data may be extracted through a system query, random access algorithm, or similar automated process. Operating system packages may include systems operable on Microsoft, Apple, Linux, and similar operating systems. Libraries may consist of a series of files, folders, and databases of information stored on one of any indexed data repositories. A program language library may contain several of an exemplary program languages including but not limited to Javascript, Swift, Scala, Go, Python, Elm, Ruby, C#, C++ and other similar sources of software code.

[00206] Another method of the disclosed embodiment may identify the installed software application 507 based on the extracted data. The method of the processor 500 may perform only this function or this function among many to accomplish the layered cybersecurity defense described herein this disclosed cloud-based security environment.

[00207] A further disclosed method of the present embodiment may include at least one processor 500 further configured to identify a version 507 of the installed software application 505. The method of the processor may perform only this function or this function among many to accomplish the layered cybersecurity defense described herein this disclosed cloud-based security environment.

[00208] **Prioritization Techniques**

[00209] Fig. 6 is a block diagram of method 600 for risk prioritization, consistent with disclosed embodiments.

[00210] Aspects of this disclosure may include a cloud-based cybersecurity system for assessing internet exposure of a cloud-based workload with embodiments configured to access at least one cloud provider API to determine a plurality of entities capable of routing and/or filtering traffic in a virtual cloud environment associated with a target account containing the

workload. Routing traffic, as used herein, may refer to the process of selecting a path for traffic (e.g., cells, blocks, frames, packets, calls, messages, or other units of data) in a network or between or across multiple networks. The cloud-based cybersecurity system, in some embodiments, may comprise at least one processor configured to assess internet exposure as described herein.

[00211] A cloud provider API, as used herein, may refer to any application program interface that allows the end user to interact with a cloud provider's service. Workloads, as used herein, may refer to devices in a network. By way of example, workloads may include systems, devices, resources in a cloud infrastructure. By way of example, Fig. 1 illustrates examples of a workload, such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, or load balancer 115. In some embodiments, scanning system 101 of Fig. 1 may perform a scanning operation on any workload (e.g., systems, devices, resources, etc.) in cloud infrastructure 106.

[00212] In some embodiments, scanning system 101 of Fig. 1 may use an API to detect workloads such as virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115 in the source account. In some embodiments, workloads may exist as virtual machines, while in other embodiments, workloads may exist as discrete, physical devices. For example, in some embodiments, a layer 3 routing system could be implemented as a physical device (e.g., a router or switch) or a virtual device (e.g., a routing system instantiated as a virtual device on a computer).

[00213] Scanning system 101 may query devices and systems capable of routing and filtering traffic (e.g., load balancer 115, routers, switches, firewalls, security groups, API Gateways and proxies) using an API provided through a cloud service provider's system to determine network configurations, and may evaluate them against known problematic configurations or other configurations. These devices can be cloud native and / or unmanaged (e.g. NGINX proxy)

[00214] A cloud environment, as used herein, may refer to a platform implemented on, hosted on, and/or accessing servers that are accessed over the Internet. A scanner account, as used herein, may refer to any type of account associated with a scanner. A scanner, as used herein, may refer to a device for examining, reading, or monitoring something. By way of example, Fig. 1 illustrates one example of a scanner, scanning system 101 of Fig. 1. A target account containing the workload, as used herein, may refer to a location on a network server selected as an aim of connection, for example. In some embodiments, scanning system 101 of Fig. 1 may determine entities capable of routing traffic.

[00215] In some embodiments, the plurality of entities includes a virtual network appliance. A virtual network appliance, as used herein, may refer to a machine focused on virtualizing network functionality. In some embodiments, a typical network virtual appliance may

include functionality directed to various layer four to seven. That is, such appliances may implement functionality associated with OSI model layers 4-7 (Transport, Session, Presentation, and Application). Such functionality may include, for example, a firewall, WAN optimizer, application delivery controllers, routers, load balancers, IDS/IPS, proxies, or SD-WAN edge device. In some embodiments, scanning system 101 of Fig. 1 may determine a virtual network appliance capable of routing traffic.

[00216] In some embodiments, the virtual network appliance is at least one of a load balancer, a firewall, a proxy, or a router. A load balancer, as used herein, may refer to any device that acts as a reverse proxy and distributes network or application traffic across a number of servers. In some embodiments, load balancer 115 may comprise one or more systems that balance incoming requests between the different systems and devices of cloud infrastructure 106. For example, load balancer 115 may be configured to determine usage (e.g., processor load, used storage capacity) of systems or devices in cloud infrastructure 106 to assist in determining where to route an incoming request from network 105 to store data, perform processing, or retrieve data.

[00217] Load balancer 115 may be configured to receive an incoming request from user device 102. Upon receipt of the request, load balancer 115 may consult a data store (part of or separate from load balancer 115; not pictured) to determine usage or forecasted usage of various systems or devices in cloud infrastructure 106, and may forward the request to the systems or devices having the lowest usage or forecasted usage. A firewall, as used herein, may refer to a device or system that monitors incoming and outgoing network traffic and permits or blocks data packets based on a set of security rules. A proxy, as used herein, may refer to a device or system that translates traffic between networks or protocols. In some embodiments, a proxy may be implemented as an intermediary a device or system separating end-user devices from destinations devices (e.g., web servers) by routing traffic through them. Proxies (or proxy servers) may be configured to provide varying levels of functionality, security, and privacy depending on use case, needs, or company policy. A router, as used herein, may refer to a device or system that connects a local network to the internet.

[00218] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to query the at least one cloud provider API to determine at least one networking configuration of the entities. Querying, as used herein, may refer to a request for data or information from a resource. A cloud provider API, as used herein, may refer to any application program interface that allows an end user to interact with a cloud provider's service. A cloud provider may include any entity through whom information is accessible over the Internet or other shared network. At least one networking configuration of the entities, as used herein, may refer to a manner in which components are arranged to make up an operating

system, networking system, or a computer system. In some embodiments, scanning system 101 of Fig. 1 may send a request to determine at least one networking configuration of the entities.

[00219] In some embodiments, 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.

[00220] A routing configuration, as used herein, may refer to a manner in which components are arranged to make up a routing system. For example, a routing configuration may provide for dynamic or static "routes" between devices on a network or on networks.

[00221] A proxy configuration, as used herein, may refer to a manner in which components are arranged to make up a proxy system. For example, a proxy configuration may provide for a configuration of how a client can route traffic through a central point (e.g., a proxy server), in order to relay multiple clients' requests from a single point on a network.

[00222] A load balancing configuration, as used herein, may refer to a manner in which components are arranged to make up a load balancing system. For example, a loan balancing configuration may define how resources such as uplinks or distributed applications may be utilized by devices on the same (or another) network, in order to balance usage of multiple redundant uplinks or distributed application nodes.

[00223] A firewall configuration, as used herein, may refer to a manner in which components are arranged to make up a firewall system. For example, a firewall configuration may comprise rules that determine what traffic may be forwarded, which devices are permitted to communicate with one another, or what protocols may be used for communication between devices.

[00224] A VPN (virtual private network) configuration, as used herein, may refer to a manner in which components are arranged to make up a VPN system. A VPN may refer to a type of network that extends a private network across a public network and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. For example, a VPN configuration may provide rules or other information by which data is sent or received over a single, encrypted route, such that intermediate points cannot determine the content, source, or destination of the data.

[00225] In some embodiments, scanning system 101 of Fig. 1 may send a request to determine at least a routing configuration, a proxy configuration, a load balancing configuration, a firewall configuration, or a VPN configuration of the entities.

[00226] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to build a graph connecting the plurality of entities based on the networking configuration. A graph connecting the plurality of entities, as used herein, may refer to a diagram showing the relation between variable quantities (e.g., entities). In some embodiments, the graph may include any data structures that includes the connections from

one entity to another. In some embodiments, scanning system 101 of Fig. 1 may construct a graph or diagram linking the plurality of entities based on the networking configuration. In some embodiments, graph includes a data structure sequentially connecting entities. 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.

[00227] In some embodiments, a graph may include directional vectors indicating directions of dataflow. Directional vectors, as used herein, may refer to a quantity having direction as well as magnitude (indicating, for example, what kinds of traffic are allowed to flow, what ports are open, what protocols are allowed), especially as determining the position of one point in space relative to another. Directions of dataflow, as used herein, may refer to the movement of data through a system comprised of software, hardware or a combination of both along a particular course. In some embodiments, scanning system 101 of Fig. 1 may construct a graph including a data structure connecting entities sequentially or a graph including directional vectors showing the direction of data flow.

[00228] In some embodiments, to build the graph, the at least one processor may be configured to identify individual entities as nodes and connect the nodes. Nodes, as used herein, may refer to a device in a network (e.g., devices in cloud infrastructure 100). Connecting nodes, as used herein, may refer to linking or bonding devices or data points in a larger network. In some embodiments, the graph includes port numbers associated with the workload. Port numbers, as used herein, may refer to a way to 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. In some embodiments, the graph includes a path from the at least one source to the workload. A path from the at least one source to the workload, as used herein, may refer to a string of characters used to uniquely identify a location in a directory structure.

[00229] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to access a data structure identifying services publicly accessible via the Internet and capable of serving as an internet proxy. Services capable of serving as an internet proxy, as used herein, may refer to a system or router that provides a gateway between users and the internet (e.g., network 105). In some embodiments, scanning system 101 of Fig. 1 may access a configuration detecting services over the internet and capable of serving as an internet proxy.

[00230] Aspects of this disclosure may include embodiments configured to integrate the identified services into the graph. Integrating identified services into a graph, as used herein, may refer to the act of bringing together smaller components (the identified services) into a single system that functions as one (the graph). For example, scanning system 101 may

integrate information on services available in each node into the data structure, associated with each node in the data structure.

[00231] Aspects of this disclosure may include embodiments in which a cloud-based cybersecurity system is configured to traverse the graph to identify at least one source originating via the Internet and reaching the workload. Traversing the graph, as used herein, may refer to a device, such as scanning system 101, accessing the systems represented by the nodes in the graph, to determine which systems are accessible and how. For example, scanning system 101 may analyze the graph to detect sources coming from the internet and reaching the workload, by accessing a first system, and traversing between systems in a depth-first, breadth-first, or other manner, until a path to an external network (e.g., the Internet) is reached.

[00232] In some embodiments, the at least one source may be a potential malicious source. In some embodiments, the source may include a code in any part of a software system or script that is intended to cause undesired effects, security breaches or damage to a system. (In some embodiments, the Internet at large may be considered to be a potentially malicious source.)

[00233] Aspects of this disclosure may include embodiments configured to output a risk notification associated with the workload. Outputting a risk notification, as used herein, may refer to sending out data such as a message indicating a possibility of danger. In some embodiments, scanning system 101 of Fig. 1 may produce a notification indicating a risk that is related with the workload. Examples of such risk notifications may include, for example, an electronic mail message, a visual alert, an audio alert, an audio-visual alert,

[00234] In some embodiments, the risk notification includes one or more resolution recommendations. Resolution recommendations, as used herein, may refer to a suggestion or proposal as to the best course of action in view of the identified risk.

[00235] **Fig. 6** illustrates a block diagram of method 600 performed by a processor (e.g., a processor associated with scanning system 101), consistent with disclosed embodiments. In some embodiments, the processor may operate on instructions stored in a non-transitory computer readable medium. In some embodiments, the method may include seven (or more or less) steps:

[00236] Block 602: 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. In some embodiments, scanning system 101 of Fig. 1 may determine entities capable of routing traffic, consistent with the above-disclosed embodiments.

[00237] Block 604: Query the at least one cloud provider API to determine at least one networking configuration of the entities. In some embodiments, scanning system 101 of Fig. 1

may send a request to determine at least one networking configuration of the entities, consistent with the above-disclosed embodiments.

[00238] Block 606: Build a graph connecting the plurality of entities based on the networking configuration. In some embodiments, scanning system 101 of Fig. 1 may construct a graph or diagram linking the plurality of entities based on the networking configuration, consistent with the above-disclosed embodiments.

[00239] Block 608: Access a data structure identifying services publicly accessible via the Internet and capable of serving as an internet proxy. In some embodiments, scanning system 101 of Fig. 1 may access a configuration detecting services over the internet and capable of serving as an internet proxy, consistent with the above-disclosed embodiments.

[00240] Block 610: Integrate the identified services into the graph. In some embodiments, scanning system 101 of Fig. 1 may incorporate the services into the graph, consistent with the above-disclosed embodiments.

[00241] Block 612: Traverse the graph to identify at least one source originating via the Internet and reaching the workload. In some embodiments, scanning system 101 of Fig. 1 may analyze the graph to detect sources coming from the internet and reaching the workload, consistent with the above-disclosed embodiments.

[00242] Block 614: Output a risk notification associated with the workload. In some embodiments, scanning system 101 of Fig. 1 may produce a notification indicating a risk that is related with the workload, consistent with the above-disclosed embodiments.

[00243] **Techniques for Malware Detection Using Secondary System**

[00244] Aspects of this disclosure may provide a technical solution to the challenging technical problem of malware detection on a primary system using a secondary system other than the primary system. In existing technologies, malware detection system may be run on a primary system that itself may be susceptible to malware. The malware may detect existence of the malware detection system and deceive it to avoid being detected. To resolve such a technical problem, a secondary system may be provided to host the malware detection system, which may isolate the malware detection system from potential exposure or influence from the malware that may reside on the primary system. The disclosed technical solutions may increase success rate of malware detection and thus enhance security level of the primary system.

[00245] A cyber security system, as used herein, may refer to a system including any combination of software and hardware for enhancing security of a device, a platform, or another system in a network environment. By way of example, the cyber security may be implemented as a system including scanning system 101 in Fig. 1. As another example, scanning system 101 may include the disclosed cyber security system as a subset. A cloud environment, as used herein, may refer to a computing environment running on a cloud. By way of example, the cloud environment may include or be part of cloud infrastructure 106 in Fig. 1.

[00246] Consistent with disclosed embodiments, at least one processor may be configured to utilize a cloud provider API to access a block storage volume of a workload maintained on a target account in a target system of a cloud storage environment. Utilizing, as used herein, may refer to an operation of using, deploying, enacting, enabling, activating, allocating, invoking, calling, or any operation of putting a thing into use in a computer environment. An API refers to an application programming interface herein. A cloud provider in this disclosure may refer to a cloud service provider (e.g., a service provider of cloud computing, cloud storage, or any services provided to users on demand over a network). A cloud provider API, as used herein, may refer to an API provided, prepared, enabled, activated, programmed, or written by a cloud provider. A block storage volume in this disclosure may refer to a storage volume in a block storage scheme. In some embodiments, a block storage volume may be connected, disconnected, or reconnected to a system (e.g., a physical computer, a virtual machine, a network interface, a cloud, or any combination of hardware or software modules) without interfering (e.g., shutting down, halting, rebooting, or any manner of interrupting) the operation status of the system or other running tasks on the system. For example, a block storage volume may be implemented as a virtual disk. In some embodiments, a block storage volume can be associated with an account of a user of the system, the system that it is connected to, or both.

[00247] Block storage, as used herein, may refer to a data storage scheme or model in which data is saved to storage media in fixed-sized raw data chunks (referred to as “blocks”). The raw data blocks may contain no metadata. Each block may be associated with a unique address as metadata assigned to the block. Storage blocks may be controlled by an operating system (OS) and may be accessed by a protocol (e.g., iSCSI, Fibre Channel, or Fibre Channel over Ethernet).

[00248] A storage volume, as used herein, may refer to an identifiable unit (e.g., a physical unit or a logical, virtual unit) of data storage. The storage volume may be mounted to a device via an operation system and be configured with a specific file system (e.g., New Technology File System) assigned a system-unique name or number that identifies the storage volume. The storage volume may represent a named, logical area of storage that enables users and applications to access data on the underlying device. By way of example, a storage volume may be a logical disk that represents a named, logical area of a physical storage device (e.g., a hard disk drive, solid-state drive, compact disc read-only memory, digital video disk, floppy disk, or any other type of storage device). In some embodiments, a logical storage volume may span multiple physical storage devices (e.g., hard disks) and appear as a single, contiguous storage area that works like a physical volume. Storage volumes may be flexibly configured, such as being expanded, contracted, mirrored, stripped, or adapted to support multiple disks (e.g., redundant array of independent disks).

[00249] 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. By way of example, a workload may be a virtual machine, a database, a container, a Hadoop node, an application, a storage object, a load balancer, or an IAM (Identity and Access Management) configuration. A cloud storage environment in this disclosure may refer to a computing environment of a cloud computing model (referred to as "cloud storage") that stores data on the Internet through a cloud computing provider that manages and operates data storage as a service. A cloud storage environment may be managed and controlled by an operating system. The operation system may be susceptible to malware and security vulnerabilities, and may become a primary system (or referred to as a "target system" herein) that needs malware detection. By way of example, the cloud storage environment may include or be part of cloud infrastructure 106 in Fig. 1.

[00250] A target system of a cloud storage environment in this disclosure may refer to an operating system or a computing system with an installed operating system, which manages and controls the cloud storage environment or a subsystem of the target system. The target system may maintain one or more accounts for users of the cloud storage environment. 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).

[00251] By way of example, to utilize the cloud provider API to access the block storage volume (e.g., existing in storage 111A-111D in Fig. 1) of the workload, the at least one processor (e.g., a processor in scanning system 101 of Fig. 1) may communicate with the cloud storage environment (e.g., cloud infrastructure 106 in Fig. 1) via sending and receiving data (e.g., data packets) over a network (e.g., network 105 in Fig. 1), and enable one or more APIs provided by the cloud provider for accessing the block storage volume. For example, the communication and API enabling may be performed through the process of integration as described in step 201 of Fig. 2. In some embodiments, the at least one processor may create a connection (e.g., by logging in) between a first account on scanning system 101 and a second account on cloud infrastructure 106, then cause the second account to authorize access privileges to the first account via the cloud provider API for accessing storage 111A-111D in cloud infrastructure 106.

[00252] Consistent with disclosed embodiments, the at least one processor may be configured to utilize a scanner at a location of the block storage volume and on a secondary system other than the target system. A scanner, as used herein, may refer to an application, a program, a service, a process, a thread, a function, or any executable codes for performing a scanning process on a computer system to obtain information (e.g., related to structures, vulnerabilities, security issues, or any information of the computer system).

[00253] A location of the block storage volume in this disclosure may refer to a system (e.g., a physical computer, a virtual machine, a network interface, a cloud, a data center, or any combination of hardware or software modules) or an identifier associated with the system that the block storage volume is connected to. In some embodiments, the location of the block storage volume may include at least one of the target account, a secondary system account, a cloud provider account, or a third party account. By way of example, the secondary system account may be an account connected with the target account (e.g., connected by an integration process similar to step 201 in Fig. 2A). The cloud provider account may be an account connected to at least one of the target account or the secondary system account. The third party account may be an account hold by a third party and connected to at least one of the target account, the secondary system account, or the cloud provider account.

[00254] In some embodiments, the at least one processor may determine the location of the block storage volume based on at least one of the target account, a secondary system account, a cloud provider account, or a third party account. By way of example, to determine the location of the block storage volume, the at least one processor may consult a lookup table, a list, or a database that stores a relationship record between an account (e.g., a target account, a secondary system account, a cloud provider account, or a third party account) and a system (e.g., a physical computer, a virtual machine, a network interface, a cloud, or any combination of hardware or software modules) that the block storage volume is connected to. Based on the relationship record, the at least one processor may identify the system as the location using the account as a key.

[00255] A secondary system, as used herein, may refer to a computer system running at an environment or conditions not affected by another system (referred to as a "primary system" or "target system" herein). In some embodiments, the secondary system may have an operating system different from an operating system of the target account. By way of example, the target account may have a WINDOWS® operating system, and the secondary system may have a LINUX® operating system.

[00256] In some embodiments, the secondary system includes at least one of a virtual machine (e.g., any of virtual machines 107A-107D in Fig. 1), a container, or a serverless function. A container, as used herein, may refer to an OS-level virtualization instance assigned with resources by a non-virtual computer operating system (OS). A non-virtual computer operating system (OS) may create multiple isolated user space instances that may function like real computers from the point of view of programs running in them. A computer program running on a non-virtual operating system can access all available resources (e.g., connected devices, files, folders, network shares, CPU power, or any other software or hardware capabilities) of the computer. In contrast, a computer program running inside of a container can only access the resources assigned to the container.

[00257] A serverless function, as used herein, may refer to a computer function hosted in a cloud environment that may allocate resources on demand to perform a function (e.g., scanning) for a user of the cloud environment. The user of the serverless function may be free from concerns of capacity planning, configuration, management, maintenance, fault tolerance, or scaling of containers, virtual machines, or physical servers. When a serverless function is not running, no computing resources may be allocated to it (e.g., using no provisioned server, thus named "serverless"). By way of example, a serverless function may be implemented as any combination of Lambda functions, event sources, and other computing resources. A serverless function may be invoked by the user of a cloud infrastructure (e.g., a cloud environment or a cloud environment).

[00258] By way of example, to utilize the scanner at the location of the block storage volume (e.g., existing in storage 111A-111D in Fig. 1), the at least one processor (e.g., a processor in scanning system 101 of Fig. 1) may communicate with the secondary system (e.g., one or more of virtual machines 107A-107D in Fig. 1) other than the target system (e.g., the main system operating and managing storage 111A-111D) and activate the secondary system for scanning. For example, to activate the secondary system, the at least one processor may transmit a scanner (e.g., in the form of a computer program or executable codes) to the secondary system, and the secondary system may allocate computing resources (e.g., CPU powers, memory space, network ports, or any other software or hardware resources) in preparation of running the scanner. The scanner may be programmed to be stored at the location of the block storage volume or to start the scanning at the location of the block storage volume. For example, the block storage volume may be connected to a particular system in cloud infrastructure 106 in Fig. 1, and the scanner may be stored at that system (e.g., on storage connected to the system). When the scanner begins scanning, the scanner may start the scanning at that system before scanning other systems.

[00259] In some embodiments, to utilize the scanner, the at least one processor may suspend an operation of the scanner after the scan of the block storage volume. Suspending, as used herein, may refer to an operation of pausing, stopping, ceasing, freezing, halting, hanging, terminating, killing, or any manner or temporarily or permanently setting aside an ongoing process or computer program. By way of example, after the scanner completes scanning the block storage volume, the at least one processor may detect the completion by an active manner (e.g., by periodically polling the secondary system for a status of the scanning) or a passive manner (e.g., by receiving a message from the secondary system indicating the completion of the scanning). The at least one process may then send data to the secondary system, the data including a signal or computer-executable codes for suspending the operation of the scanner. After receiving the data, the secondary system may suspend the scanner and release the computing resources previously allocated to the scanner.

[00260] In some embodiments, to utilize the scanner, the at least one processor may modify a pre-utilized scanner at the location of the block storage volume based on information related to the target account to obtain a modified scanner, and then utilize the modified scanner. A pre-utilized scanner in this disclosure may refer to a scanner that has been previously utilized (e.g., on the same secondary system) and not destroyed. In some embodiments, the information related to the target account may include data or parameters indicating a status or a change of the target account, such as, for example, newly created files, deleted files, changed files, or any data indicating a state of the target account. For example, the pre-utilized scanner at the block storage volume may use a configuration file. The configuration file may include data or parameters indicating a previous status (e.g., list of existing files, public network ports, Internet-accessible data channels, or any operation state parameters) of the target account. The information related to the target account may include data indicating a current status of the target system. The current status of the target system may be different from the previous status of the target system (e.g., including changed files, changed public network ports, changed Internet-accessible data channels, or any changed operation state parameters). The at least one processor may modify the pre-utilized scanner by modifying the configuration file to reflect the current status of the target system. Then, the at least one processor may utilize the modified scanner.

[00261] Consistent with disclosed embodiments, the at least one processor may be configured to scan the block storage volume for malicious code, using the secondary system. Scanning, as used herein, may refer to an operation of traversing, checking, inspecting, reading, examining, searching, or any manner of looking up. Malicious code, as used herein, may refer to an application, a computer program, a service, a process, a thread, a function, a script, or any type of executable codes that may be executed on a computer without authorization and with a malicious or harmful intent (e.g., manipulating the computer into dangerous behaviors, writing changes or add-ons to existing computer programs or files, stealing computer files, copying sensitive information without authorization, damaging the computer itself, disabling one or more functions of the computer, or attacking another computer using the computer as a disguise). For example, malicious code may include attack scripts, computer viruses, computer worms, Trojan horses, backdoors, or any other malicious executable content. In some embodiments, the malicious code may include a rootkit. A rootkit in this disclosure may refer to any type of computer software designed to enable access to a computer or an area of its software without authorization.

[00262] By way of example, to scan the block storage volume (e.g., existing in storage 111A-111D in Fig. 1) for malicious code, the secondary system (e.g., one or more of virtual machines 107A-107D in Fig. 1) may run the utilized scanner with allocated computing resources to look up databases (e.g., a blacklist) of known malicious files, look up databases storing

patterns of malicious files and identify the patterns, look up databases of cryptographic hashes that identify malicious files, run potential malicious code in a sandbox (e.g., a confined and isolated environment for running programs without causing changes outside the confined and isolated environment), determine whether there are scanned files that are not in a database (e.g., a whitelist) of known good files, detect changes over time in the scanned environment, or any combination thereof.

[00263] In some embodiments, scanning the block storage volume may include scanning disk-backed memory. A disk-backed memory in this disclosure may refer to a storage space implemented on a disk (e.g., a hard disk) and functioning like a volatile memory from the perspective of a running program. By way of example, scanning the block storage volume may include scanning files stored in the disk-backed memory. In some embodiments, the disk-backed memory may include at least one of a page file or a cache file. A page file, as used herein, may refer to a file containing one or more pages (also referred to as “memory pages” or “virtual pages”), each of the one or more pages being a fixed-length contiguous block of virtual memory and described by a single entry in a page table. A page file may be stored on a disk (e.g., a hard disk) for memory management of a virtual memory used by an operating system. A cache file, as used herein, may refer to a file storing data so that future requests for the stored data can be served faster. The data stored in a cache file may be the result of a computation or a copy of the data stored elsewhere.

[00264] Consistent with disclosed embodiments, the at least one processor may be configured to identify malicious code based on the scan. By way of example, if the secondary system scans the block storage volume by consulting databases of known malicious files and finds a scanned file matches a record in the databases, the at least one processor may identify the matched scanned file as the malicious code. In another example, if the secondary system scans the block storage volume by consulting databases storing patterns of malicious files and identifies a pattern of a scanned file matches one or more of the stored patterns, the at least one processor may identify the scanned file as the malicious code. In yet another example, if the secondary system scan the block storage volume by consulting databases of cryptographic hashes that identify malicious files and finds that a cryptographic hash of a scanned file matches one or more of the cryptographic hashes that identify malicious files, the at least one processor may identify the scanned file as the malicious code.

[00265] In yet another example, if the secondary system scans the block storage volume by running potential malicious code in a sandbox and finds that running a scanned file causes malicious or harmful changes or behavior (e.g., an attempt to modify a system setting, an attempt to access files outside of a set of directories, an attempt to perform a network mapping process), the at least one processor may identify the scanned file as the malicious code. In yet another example, if the secondary system scans the block storage volume to determine whether

there are scanned files that are not in a database of known good files and finds that a scanned file is not in the database of known good files, the at least one processor may identify the scanned file as the malicious code. As yet another example, if the secondary system scans the block storage volume to detect changes over time in the scanned environment and finding that a scanned file causes a malicious or harmful change in the scanned environment over time, the at least one processor may identify the scanned file as the malicious code. It should be noted that the identification of the malicious code based on the scan may be implemented in various manners and is not limited to the examples described herein.

[00266] Consistent with disclosed embodiments, the at least one processor may be configured to output from the secondary system, a notification of a presence of malicious code in the target system. In some embodiments, the notification may be a pop-up window, a pop-pup dialog, a flashing symbol, an email, a text message, a mobile application notice, a phone call, a warning sound, a tactile feedback (e.g., a vibration), or any type of communications. By way of example, the secondary system may trigger a communication channel between the secondary system and the target system and feed one or more parameters to the communication channel, in which the one or more parameters represent the presence of the malicious code. The secondary system may then send data representing the notification via the triggered communication channel including the one or more parameters to the target system. After receiving the data, the target system may parse the data and identify the one or more parameters. The target system may then display the notification (e.g., as a pop-up window on a screen) that present the presence of the malicious code.

[00267] Consistent with disclosed embodiments, the at least one processor may be further configured to identify a source of the malicious code. The notification of the presence of malicious code outputted from the secondary system in the target system may include information related to the identified source of the malicious code. In some embodiments, the source of the malicious code may include identity of a developer or distributor of the malicious code, the landing location (e.g., a network port or a vulnerable access point) where the malicious code is imported to the target system, source codes of the malicious code, or any information or data indicating the origination of the malicious code. The information related to the identified source of the malicious code may include a text, a picture, a symbol, a video, a link, a voice, or any type of data for presenting and describing the identified source of the malicious code.

[00268] By way of example, **Fig. 7** is a block diagram illustrating an exemplary process 300 of cyber security scanning for a cloud environment, consistent with the disclosed embodiments. While the block diagram may be described below in connection with certain implementation embodiments presented in other figures, those implementations are provided for illustrative purposes only, and are not intended to serve as a limitation on the block diagram.

For example, the steps in Fig. 7 may be executed in any order, steps may be duplicated, or steps may be omitted. In some embodiments, the process 300 may be performed by at least one processor (e.g., a CPU) of a computing device or system (e.g., scanning system 101 in Fig. 1) to perform operations or functions described herein, and may be described hereinafter with reference to Figs. 1-2D by way of example. In some embodiments, some aspects of the process 300 may be implemented as software (e.g., program codes or instructions) that are stored in a memory or a non-transitory computer-readable medium. In some embodiments, some aspects of the process 300 may be implemented as hardware (e.g., a specific-purpose circuit). In some embodiments, the process 700 may be implemented as a combination of software and hardware.

[00269] Fig. 7 includes process blocks 702-710. At block 702, at least one processor may utilize a cloud provider API to access a block storage volume of a workload maintained on a target account in a target system of a cloud storage environment.

[00270] At block 704, the at least one processor may utilize a scanner at a location of the block storage volume and on a secondary system other than the target system. In some embodiments, the secondary system may include at least one of a virtual machine, a container, or a serverless function. In some embodiments, the secondary system may have an operating system different from an operating system of the target account.

[00271] In some embodiments, to utilize the scanner, the at least one processor may suspend an operation of the scanner after the scan of the block storage volume. In some embodiments, to utilize the scanner, the at least one processor may modify a pre-utilized scanner at the location of the block storage volume based on information related to the target account to obtain a modified scanner, and the utilize the modified scanner.

[00272] At block 706, the at least one processor may scan the block storage volume for malicious code, using the secondary system. In some embodiments, to scan the block storage volume, the at least one processor may scan disk-backed memory. For example, the disk-backed memory includes at least one of a page file or a cache file. In some embodiments, the malicious code includes a rootkit.

[00273] At block 708, the at least one processor may identify malicious code based on the scan. At block 710, the at least one processor may output from the secondary system, a notification of a presence of malicious code in the target system.

[00274] Consistent with disclosed embodiments, besides blocks 702-710, the at least one processor may further determine the location of the block storage volume based on at least one of the target account, a secondary system account, a cloud provider account, or a third party account.

[00275] Consistent with disclosed embodiments, besides blocks 302-310, the at least one processor may further identify a source of the malicious code. The notification outputted in block 710 may include information related to the identified source of the malicious code.

[00276] **Forward and Rearward Facing Attack Vector Visualization Techniques**

[00277] **Fig. 8** is a schematic block diagram illustrating an exemplary embodiment of a system for performing visualization of forward and backward facing threats.

[00278] In one of many embodiments, the system may include a graphical user interface system for providing forward and backward facing attack vector visualizations. A graphical user interface system may include systems similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows 10 Mobile, PalmOS-Web OS, or Firefox OS. Forward facing attack vectors may be those external entities that may access an analyzed asset. Backward facing attack vectors may include the possible impacts to an analyzed asset and those impacts' possible effect. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of attack vectors.

[00279] In an embodiment, the system may comprise at least one processor configured to identify assets 801 in a cloud environment. The at least one processor may include a processor such as user device 102 of Fig. 1. The at least one processor may include a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service.

[00280] Another embodiment contemplated may consist of at least one processor configured to identify risks 803 associated with each of the identified asset. The identified risks 803 associated with each of the identified assets may be listed sequentially based on probability of risk, severity of risk, or similar method of ordering the risks. The identified risks 303 associated with an identified asset may be scored based on the level of risk to the asset and may be categorized as a high risk, medium risk, or low risk. Once the level of risk can be determined by scanning system 101, the level of risk can be conveyed and displayed to a user device 102 with a representative graphical display.

[00281] An exemplar embodiment may additionally comprise at least one processor configured to identify relationships 805 between at least some of the identified assets, the relationships including at least one of a trust 807A, a network connectivity 807B, or a mechanism of network proxying 807C. A trust 807A may establish a relationship and may

involve cryptography, digital signatures, electronic certificates, or similar method of establishing a trusting relationship. A network connectivity 807B may include wireless or wired connections and may be established once a relationship of trust 807A is established. A mechanism of network proxying 807C can be accommodated once trust 807A and network connectivity 807B are established through any number of connected servers to accomplish proxying.

[00282] A further embodiment may also comprise a processor configured to receive an identification 809 of a specific asset under investigation. The specific asset under investigation may be a physical or virtual machine that has software operating a specific version that is vulnerable to any number of associated risks. The identification 809 of a specific asset under investigation may involve an IP address, universally unique identifier, or similar system to uniquely or semi-uniquely identify the specific asset under investigation.

[00283] A system among several embodiments may include a processor configured to perform a forward analysis 811A of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment. A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. The analysis performed by scanning system 101 may determine the origin of the forward threat, the nature of the threat to the analyzed system, and the seriousness of the threat originating from the Internet. For example, scanning system 101 may identify at least one possible attack vector to the workload currently being analyzed. The network outside of the cloud environment may identify new or existing threats and attack vectors currently threatening the associated system and workload.

[00284] An exemplary embodiment may further comprise a processor configured to perform a backward analysis 811B of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset. A backward analysis 811B of the associated machine may look downstream to related workloads and systems connected to the analyzed systems. A scan downstream may involve, among several embodiments, the downstream exposure risk including an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset. The at least one exposure risk may include one or more entry point to access the exposed asset and any related machines potentially vulnerable to attack from the analyzed and exposed asset.

[00285] Another embodiment may comprise a processor configured to output a signal 813 to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. The display of a forward or

backward path to a specific asset may be one of several visual displays such as linear, graphical, or through computer-generated images. The visualization of a plurality of entry points and lateral movement risks with the plurality of entry points may include visualization of the plurality of paths may sequence the paths based on likelihood of access via that path or the severity of potential threat in a path of access to an asset.

[00286] An example embodiment may comprise a system wherein the network outside the cloud environment includes the Internet. The network outside the cloud environment may include access to the Internet in order for the system to access a plurality of analyzed and potentially vulnerable assets.

[00287] Another embodiment may comprise a system wherein the assets in the cloud environment include at least one of a virtual machine, a network appliance, a storage appliance, a compute instance, or an engine instance. A virtual machine may include an emulated version of a computer - including an operating system, memory, storage, graphics processing - such that it can be indistinguishable from a standard (non-virtual) machine to a running program. A network appliance may include a plurality of servers connected through the Internet in order to establish a system and network of stored assets and workloads. A storage appliance may be comprised by a server designed to store workloads and assets to be analyzed. A compute instance may involve input and processing of data entries that cause the analyzed asset to be accessed and run. An engine instance may include a search algorithm run to analyze and search the analyzed workloads or virtual machines to determine the software and data stored on the said virtual machine or stored instance.

[00288] One among many embodiments may comprise a system wherein identifying the assets in a cloud environment includes identifying the assets based on at least one of an identity and access management policy, an organization policy, or an access policy. An identity and access management policy may comprise a software tool or similar method of establishing identity of users attempting to access a system to verify identities of users and control access to an associated workload or system functionality. An organization policy may include a policy applied to all users by the controlling organization of a workload. An access policy may determine the authorized users of an associated system.

[00289] One possible embodiment may comprise a system wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream asset or a downstream asset. The indication will be via a visualization showing specific alternative paths connecting specific assets or it may visualize a plurality of alternative paths connecting specific assets. Alternative paths connecting between the specific asset and an upstream asset or a downstream asset may involve a direct connection, such as via the Internet, or an indirect connection via one or more series of servers acting as proxy servers between the analyzed asset and the upstream or downstream asset.

[00290] An exemplary embodiment may further comprise a system wherein the visualization includes a presentation of the alternative paths. The alternative paths may be presented visually through a graphical representation, sequential list of alternative paths, or a computer-generated image that represents the plurality of alternative paths.

[00291] Another embodiment may comprise a system wherein the presentation of the forward and backward paths indicates port numbers for each pathway. The port numbers may be indicated by a series of alphanumeric characters (e.g., “HTTP” or “80”) or other identifier.

[00292] One among several exemplary embodiments may further comprise a system wherein the visualization of the entry points indicates at least one entry point of risk. The visualization may indicate one or more points of entry to a given analyzed system. An entry point of risk may involve a port of access to the workload, and may involve a one or more entry points visualized for use by user device 102.

[00293] Another exemplary embodiment may comprise a system wherein the at least one processor is further configured to monitor network activities of the assets in a cloud environment. The processor may monitor continuously, intermittently, or as directed by an authorized user to direct monitoring of the analyzed network activities of the assets. The cloud environment may be, among several embodiments, the cloud infrastructure 106.

[00294] An embodiment contemplated may comprise a system further configured to detect a potential risk associated with the specific asset based on a network activity of the specific asset. The potential risk may be associated to a specific asset based on historical threats to the specific asset or a list of possible threats to the specific asset based on its current network activities.

[00295] Another embodiment may comprise a system further configured to detect a potential risk associated with the specific asset based on a network activity of an upstream asset of the specific asset. The potential risk associated with the specific asset may be identified by scanning system 101 and determine whether a potential risk can be classified as high risk, medium risk, or low risk. The potential risk may be identified based on the specific asset, the network activity performed by the specific asset, or the specific asset type or related software of the upstream asset.

[00296] One of many embodiments may comprise a system further configured to detect a potential risk associated with the specific asset based on a network activity of a downstream asset of the specific asset. The potential risk may be identified based on the specific asset, the network activity performed by the specific asset, or the specific asset type or related software of the upstream asset. For example, scanning system 101 may perform a scan on an asset being analyzed, and based on the one or more entry points of risk, may determine if the downstream asset of the specific asset poses a potential risk to the analyzed system.

[00297] Another exemplary embodiment may comprise a method for generating a graphical user interface for providing forward and backward facing attack vector visualizations. The method may include a method of generating a graphical user interface similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows 10 Mobile, PalmOS-Web OS, or Firefox OS. A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. Backward analysis 811B may include the possible impacts to an analyzed asset and those impacts' possible effect. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of attack vectors.

[00298] One exemplar embodiment may further comprise a method of identifying assets 801 in a cloud environment. A method may include at least one processor that may include a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service.

[00299] One of several embodiments may comprise a method of identifying risks 803 associated with each of the identified assets 801. The method of identifying risks 803 associated with each of the identified assets 801 may be listed sequentially based on probability of risk, severity of risk, or similar method of ordering the risks. The identified risks 803 associated with an identified asset may be scored based on the level of risk to the asset and may be categorized as a high risk, medium risk, or low risk. Once the level of risk can be determined by scanning system 101, the level of risk can be conveyed and displayed to a user device 102 with a representative graphical display.

[00300] Another embodiment contemplated may comprise a method for identifying relationships 305 between at least some of the identified assets 801, the relationships including at least one of a trust 807A, a network connectivity 807B, or a mechanism of network proxying 807C. A trust 807A may establish a relationship and may involve cryptography, digital signatures, electronic certificates, or similar method of establishing a trusting relationship. A network connectivity 807B may include wireless or wired connections and may be established once a relationship of trust 807A is established. A mechanism of network proxying 807C can be accommodated once trust 807A and network connectivity 807B are established through any number of connected servers to accomplish proxying.

[00301] An embodiment may further comprise a method of receiving an identification of a specific asset 809 under investigation. The specific asset under investigation may be a physical

or virtual machine that has software operating a specific version that is vulnerable to any number of associated risks 803. The identification 809 of a specific asset under investigation may involve an IP address, universally unique identifier, or similar system to uniquely or semi-uniquely identify the specific asset under investigation.

[00302] An embodiment may also comprise a method of performing a forward analysis 811A of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment. A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. A forward analysis may involve analysis of the specific asset under identification to identify at least one possible Internet-originating attack vector to the asset. The analysis performed by scanning system 101 may determine the origin of the forward threat, the nature of the threat to the analyzed system, and the seriousness of the threat originating from the Internet. For example, scanning system 101 may identify at least one possible attack vector to the workload currently being analyzed. The network outside of the cloud environment may identify new or existing threats and attack vectors currently threatening the associated system and workload.

[00303] One embodiment may comprise a method of performing a backward analysis 811B of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset. Backward analysis 811B may include the possible impacts to an analyzed asset and those impacts' possible effect. A scan downstream may involve, among several embodiments, the downstream exposure risk including an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset. The at least one exposure risk may include one or more entry point to access the exposed asset and any related machines potentially vulnerable to attack from the analyzed and exposed asset.

[00304] An embodiment may comprise a method of outputting a signal 813 to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. The display of a forward or backward path to a specific asset may be one of several visual displays such as linear, graphical, or through computer-generated images. The visualization of a plurality of entry points and lateral movement risks with the plurality of entry points may include visualization of the plurality of paths may sequence the paths based on likelihood of access via that path or the severity of potential threat in a path of access to an asset.

[00305] Another embodiment may comprise a method wherein the network outside the cloud environment includes the Internet. The network outside the cloud environment may include access to the Internet in order for the system to access a plurality of analyzed and potentially vulnerable assets.

[00306] One of several embodiments may comprise a method wherein the assets in the cloud environment include at least one of a virtual machine, a network appliance, a storage appliance, a compute instance, or an engine instance. A virtual machine may include an emulated version of a computer - including an operating system, memory, storage, graphics processing - such that it can be indistinguishable from a standard (non-virtual) machine to a running program. A network appliance may include a plurality of servers connected through the Internet in order to establish a system and network of stored assets and workloads. A storage appliance may be comprised by a server designed to store workloads and assets to be analyzed. A compute instance may involve input and processing of data entries that cause the analyzed asset to be accessed and run. An engine instance may include a search algorithm run to analyze and search the analyzed workloads or virtual machines to determine the software and data stored on the said virtual machine or stored instance.

[00307] An embodiment may further comprise a method wherein identifying assets in a cloud environment includes identifying the assets based on at least one of an identity and access management policy, an organization policy, or an access policy. An identity and access management policy may comprise a software tool to verify identities of users and control access to an associated workload or system functionality. An organization policy may include a policy applied to all users by the controlling organization of a workload. An access policy may determine the authorized users of an associated system.

[00308] An embodiment may also comprise a method wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream or a downstream asset. The indication may be via a visualization showing specific alternative paths connecting specific assets or it may visualize a plurality of alternative paths connecting specific assets.

[00309] Another exemplar embodiment may comprise a method wherein the visualization includes a presentation of the alternative paths. The alternative paths may be presented visually through a graphical representation, sequential list of alternative paths, or a computer-generated image that represents the plurality of alternative paths.

[00310] Another embodiment may comprise 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.

[00311] One embodiment may comprise a non-transitory computer-readable medium storing instructions that, when executed by at least one processor 800, are configured to

perform operations identifying assets 801 in a cloud environment. the instructions stored on a non-transitory computer-readable medium may include at least one processor 800 that may include a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service.

[00312] An embodiment may further comprise 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 300 to perform operations comprising identifying risks 803 associated with each of the identified asset. The instructions performed of identifying risks 803 associated with each of the identified assets 301 may be listed sequentially based on probability of risk, severity of risk, or similar method of ordering the risks.

[00313] Another embodiment may comprise 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 identifying relationships 805 between at least some of the identified assets 301, the relationships including at least one of a trust 807A, network connectivity 807B, or a mechanism of network proxying 807C.

[00314] An exemplar embodiment may comprise 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 receiving an identification of a specific asset 809 under investigation. The instructions may include receiving information identifying the specific asset 809. The specific asset 809 under investigation may be a physical or virtual machine that has software operating a specific version that is vulnerable to any number of associated risks 803.

[00315] One among many embodiments may comprise 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 performing a forward analysis 811A of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment. . A forward analysis 811A of the specific asset may involve those external entities that may access an analyzed asset. And the scanning system 101 may identify at least one possible attack vector to the workload currently being analyzed. The network outside of the cloud environment may identify new or existing threats and attack vectors currently threatening the associated system and workload.

[00316] An embodiment may further comprise 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 performing a backward

analysis 811B of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset. Backward analysis 811B may include the possible impacts to an analyzed asset and those impacts' possible effect. The at least one exposure risk may include one or more entry point to access the exposed asset and any related machines potentially vulnerable to attack from the analyzed and exposed asset.

[00317] An embodiment may comprise 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 outputting a signal 813 to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. The display of a forward or backward path to a specific asset may be one of several visual displays such as linear, graphical, or through computer-generated images. The visualization of a plurality of entry points and lateral movement risks with the plurality of entry points may include visualization of the plurality of paths may sequence the paths based on likelihood of access via that path or the severity of potential threat in a path of access to an asset.

[00318] **Passive Key Identification Techniques**

[00319] Aspects of this disclosure may provide a technical solution to the challenging technical problem of identifying access keys (e.g., passwords, Secure Shell (SSH) keys, or cloud keys) to compute resources (e.g., machines, containers, storages, or any hardware or software component) in a cloud environment without using the access keys to access the compute resources. In existing technologies, scanning systems may identify access keys to compute resources and may need to verify the identified keys by actually accessing the compute resources using the access keys. In such cases, the actual accesses may generate logs or records (e.g., for successful or failed accesses) that can constitute activity patterns similar to an attacker to the system. Such activity patterns may distract a system administrator (e.g., an individual or a computer program) to misidentify the scanning system as an attacker. Further, actual accesses may create loads on both the scanning system and the compute resources and may reduce the computing resources and power for other tasks. To resolve such a technical problem, a cryptographic analysis may be performed to identify a first set of fingerprints of the identified keys, and trust configurations of the compute resources may be analyzed to identify a second set of fingerprints of the compute resources. The first and second set of fingerprints may be compared to match keys with the compute resources without using the keys to access the compute resources. By doing so, the scanning system may perform its

function without raising false alarms to the system administrator, while reducing computing costs to the cloud environment.

[00320] A cyber security system, as used herein, may refer to a system including any combination of software and hardware for enhancing security of a device, a platform, or another system in a network environment. By way of example, the cyber security may be implemented as a system including scanning system 101 in Fig. 1. As another example, scanning system 101 may include the disclosed cyber security system as a subset. A compute resource, as used herein, may refer to a database, a virtual machine, a storage, a keystore, a scanning system, a load balancer, a server, a computer, a container, or any physical or virtual component integrated in or communicatively connected to a cloud infrastructure. By way of example, the compute resource may be scanning system 101, any of databases 103A-103D, any of virtual machines 107A-107D, any of databases 109A-109D, any of storage 111A-111D, any of keystores 113A-113D, or load balancer 115 in cloud infrastructure 106 in Fig. 1.

[00321] A key to access a compute resource, as used herein, may refer to computerized data that includes a credential for granting permission to an accessor to access the compute resource, or a service or function provided by the compute resource. By way of example, the key may include at least one of a password, a remote access key, a script containing a password, a cloud key (e.g., a credential file for accessing a cloud service, such as an AWS credential file), an Secure Shell (SSH) key, an AWS key, a private component of a private-public key pair, or any type of credential that provide checked or unchecked access to the compute resource. Matching a key to a compute resource, as used herein, may refer to confirming, verifying, validating, or any operation or process of determining that the key is valid for an accessor (e.g., another compute resource) to access the compute resource. For example, to match a key to a compute resource, an accessor may test by using the key to access the compute resource and determines that the key is matched to the compute resource if such access is successful. As another example, to match a key to a compute resource, an accessor may perform methods or algorithms to verify that the key is matched to the compute resource without using the key to access the compute resource.

[00322] Consistent with disclosed embodiments, at least one processor may be configured to analyze a cloud environment to identify a plurality of keys to the compute resources in the cloud environment. In some embodiments, the at least one processor may analyze the cloud environment to identify the plurality of keys to the compute resources in the cloud environment in any of steps 231, 237, or 239 in FIG. 2D. In some embodiments, the at least one processor may analyze the cloud environment to identify the plurality of keys to the compute resources in the cloud environment in any step described in association with FIGS. 2A-2D. In some embodiments, at least one of the plurality of keys may include at least one of a password (e.g., a combination of alphanumeric characters and symbols), a script (e.g., a

PYTHON® script or a shell script) containing a password, a cloud key, or an Secure Shell (SSH) key. For example, a cloud key may be a key for accessing a cloud resource.

[00323] A cloud environment, as used herein, may refer to a computing environment running on a cloud. By way of example, the cloud environment may include or be part of cloud infrastructure 106 in Fig. 1. Analyzing a cloud environment, as used herein, may refer to an operation or a process of separating or dividing the cloud environment into parts or components (e.g., physical components or logic components), then determining nature and relationship of the parts or components based on data associated with the parts or components.

[00324] By way of example, if the cloud environment is cloud infrastructure 106 in Fig. 1, to analyze cloud infrastructure 106, the at least one processor may access cloud infrastructure 106 (e.g., via network 105 or via internal connections in cloud infrastructure 106) and obtain a list of compute resources in cloud infrastructure 106 (e.g., scanning system 101, any of databases 103A-103D, any of virtual machines 107A-107D, any of databases 109A-109D, any of storage 111A-111D, any of keystores 113A-113D, or load balancer 115). For example, the at least one processor may obtain the list of compute resources by reading a configuration file from a database (e.g., one of databases 109A-109D), or visit available compute resources one after another through communicative connections between the compute resources to generate the list of compute resources. After obtaining the list of compute resources in cloud infrastructure 106, the at least one processor may determine the nature of the compute resources, such as their types, numbers, functions, or any other characteristics or features. The at least one processor may further determine the relationship between the compute resources, such as communicative connections, access privileges, data input/output directions, or controls between the compute resources.

[00325] Identifying a key to a compute resource, as used herein, may refer to an operation or a process of locating, recognizing, or any operation or process or analyzing computerized data or information to determine that the computerized data or information is a key to the compute resources. By way of example, the at least one processor may analyze cloud infrastructure 106 in Fig. 1 to recognize a list of compute resources in cloud infrastructure 106 (e.g., scanning system 101, any of databases 103A-103D, any of virtual machines 107A-107D, any of databases 109A-109D, any of storage 111A-111D, any of keystores 113A-113D, or load balancer 115). Then, the at least one processor may read computerized data stored in a compute resource and identify that some of the computerized data are one or more keys to other compute resources in cloud infrastructure 106. For example, to recognize the keys, the at least one processor may read and compare the computerized data with records stored in keystores 113A-113D, and determine that the computerized data is a key if it matches a record in any of keystores 113A-113D. As another example, the at least one processor may read the computerized data and check its syntaxes, text string patterns, file formats, file properties,

encryption manners, library versions, software versions, or any other characteristics or features of the computerized data, and determine that the computerized data is a key if its checked characteristics or features fit a predetermined pattern of a key or fit an entry in a dictionary of keys. For example, the dictionary of keys may be stored in a keystore (e.g., any of keystores 113A-113D).

[00326] In some embodiments, the plurality of keys may be stored in at least one workload. A workload, for example, 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. By way of example, a workload may be a virtual machine, a database server, a container, a Hadoop node, an application, a storage server, a load balancer, or an IAM (Identity and Access Management) configuration. By way of example, with reference to FIG. 1, cloud infrastructure 106 may include workloads such as scanning system 101, databases 103A-103D, virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115.

[00327] Consistent with disclosed embodiments, at least one processor may be configured to perform a cryptographic analysis on the plurality of keys to identify a first set of fingerprints that uniquely identify each of the plurality of keys. The first set of fingerprints may be non-functional. In some embodiments, the at least one processor may perform the cryptographic analysis in step 237 of FIG. 2D. In some embodiments, the at least one processor may perform the cryptographic analysis in any step described in association with FIGS. 2A-2D.

[00328] A cryptographic analysis on data, as used herein, may refer to an operation or process of identifying ciphertext, ciphers and cryptosystems contained in the data to discover hidden aspects of the data that may improve or weaken security of the data or compute resources associated with the data. In some embodiments, to perform the cryptographic analysis on the plurality of keys, the at least one processor may perform an algorithm on the plurality of keys to convert the plurality of keys into another form of data. For example, the at least one processor may perform a hashing process on the plurality of keys (e.g., by inputting values of the keys to a hash function) to obtain corresponding hash values of respective keys.

[00329] A fingerprint of a key, as used herein, may refer to non-functional data (e.g., a numeric value, an alphanumeric string, or any combination of letters, numbers, or symbols) generated based on the key and may uniquely identify the key. By way of example, a fingerprint may be generated by performing a hashing process on a key or a part of the key. The hashing process may include, for example, an MD5 algorithm, an SHA-1 algorithm, an SHA-2 algorithm, an SHA-3 algorithm, a RIPEMD-160 algorithm, a BLAKE2 algorithm, a BLAKE3 algorithm, or any type of cryptographic hash algorithms. In such cases, the fingerprint may be a hash value of the key, a part (e.g., a truncated part) of the hash value of the key, a hash value of a part of the key, or a part (e.g., a truncated part) of the hash value of the part of the key. The hash value

may be unique (e.g., non-overlap with any hash value of any other key) and non-functional (e.g., unable to be used as a key to access a compute resource). To identify the first set of fingerprints, the at least one processor may perform the hashing process on the plurality of keys to obtain a plurality of hash values, and search the cloud environment to check whether there exists any compute resource that stores the plurality of hash values.

[00330] In some embodiments, at least one of the first set of fingerprints may be non-identical to any key of the plurality of keys. For example, if the first set of fingerprints are hash values, they may be non-identical to the plurality of keys.

[00331] Consistent with disclosed embodiments, at least one processor may be configured to analyze trust configurations of the compute resources to identify a second set of fingerprints for each of the compute resources. In some embodiments, the at least one processor may analyze the trust configurations in step 233 of FIG. 2D. In some embodiments, the at least one processor may analyze the trust configurations in any step described in association with FIGS. 2A-2D.

[00332] A trust configuration, as used herein, may refer to data (e.g., a stored file, a database entry, a value stored in a data structure, or any computerized information) that stores a pre-established trust relationship between at least two compute resources. For example, a trust configuration may be a file storing a trust policy (e.g., a public key infrastructure (PKI), a digital signature system, or an AWS trust policy). Analyzing a trust configuration, as used herein, may refer to an operation or a process of separating or dividing compute resources associated with the trust configuration into units or groups, then determining nature and relationship of the units or groups based on relationship data associated with the units or groups. For example, the relationship data may include flags, indicators, or any type of value that indicates a first compute resource has an access privilege to a second compute resource.

[00333] A fingerprint for a compute resource, as used herein, may refer to non-functional data (e.g., a numeric value, an alphanumeric string, or any combination of letters, numbers, or symbols) generated based on a key for accessing the compute resource and may uniquely identify the key. By way of example, a fingerprint for a compute resource may be generated by applying a cryptographic hash function (e.g., an SHA-1 or SHA-2 function) to a key for accessing the compute resource to obtain a sequence of bytes (e.g., a hash value or a modified hash value), and the sequence of bytes may be the fingerprint for the compute resource. The key may be uniquely associated with the compute resource.

[00334] For example, to identify the second set of fingerprints, the at least one processor may analyze a trust configuration file (e.g., a file storing fingerprints and their associated data) by reading the byte stream of the trust configuration file to determine whether any sequence of bytes that fits to a predetermined pattern (e.g., an alphanumeric string pattern). The sequences

of bytes that fit to the predetermined pattern may be determined as the second set of fingerprints.

[00335] Consistent with disclosed embodiments, at least one processor may be configured to compare the first set of fingerprints with the second set of fingerprints to match keys with the compute resources without using the keys to access the compute resources. In some embodiments, the at least one processor may compare the first set of fingerprints with the second set of fingerprints in any step described in association with FIGS. 2A-2D. Comparing a first fingerprint and a second fingerprint, as used herein, may refer to performing a textual, numerical, or semantic comparison of the first fingerprint and the second fingerprint, or performing an algorithmic conversion of at least one of the first fingerprint or the second fingerprint to further perform a textual, numerical, or semantic comparison of the conversion results. By way of example, a first compute resource may store a key (e.g., a public key) for accessing a second compute resource. The at least one processor may analyze the cloud environment to identify the key from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. A third compute resource (e.g., may be the same as or different from the second compute resource) may store in its trust configurations a second fingerprint of the key. The second fingerprint may be generated by a different processor at a different time. The at least one processor may then analyze the trust configurations of the third compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other. If the first fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource. As can be seen, after comparison, if the first set of fingerprints are matched with the second set of fingerprints, keys may be matched with the compute resources, and the at least one processor needs not use the keys to access the compute resources to confirm that the keys are matched to the compute resources.

[00336] Consistent with disclosed embodiments, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the first set of fingerprints. A multi-machine interaction, as used herein, may refer to data input/output, operation control, status polling, scanning, or any type of an interaction between a plurality of machines in a cloud environment. Analyzing a multi-machine interaction, as used herein, may refer to an operation or a process of separating or dividing machines in the cloud environment into units or groups, then determining nature and relationship of the units or groups based on relationship data associated with the units or groups. For example, the relationship data may be the first set of fingerprints.

[00337] By way of example, to analyze the multi-machine interaction, the at least one processor may compare the first set of fingerprints with the second set of fingerprints. If a first fingerprint of the first set of fingerprints matches with a second fingerprint of the second set of fingerprints, the at least one processor may determine that a first compute resource associated with the first fingerprint may have an inter-machine interaction (e.g., having an ability to access the other) with a second compute resource associated with the second fingerprint. By way of example, a first compute resource may store a key (e.g., a private key) for accessing a second compute resource. The at least one processor may analyze the cloud environment to identify the key from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. A third compute resource (e.g., may be the same as or different from the second compute resource) may store in its trust configurations a second fingerprint of the key. The at least one processor may then analyze the trust configurations of the third compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other (e.g., being the same). If the first fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource, and may further determine an inter-machine interaction in which the first compute resource has ability to access the second compute resource.

[00338] Consistent with disclosed embodiments, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the plurality of keys. For example, the at least one processor may use the plurality of keys as the relationship data for determining nature and relationship of compute resources in the cloud environment. By way of example, to analyze the multi-machine interaction, the at least one processor may compare a first key for accessing a first one of the compute resources and a second key for accessing a second one of the compute resources. If the first key matches with (e.g., is identical to) the second key, the at least one processor may determine that the first one of the compute resources may have an interaction (e.g., having an ability to access the other) with the second one of the compute resources.

[00339] By way of example, **Fig. 9** is a block diagram illustrating an exemplary process 900 of matching keys with compute resources in a cloud environment, consistent with the disclosed embodiments. While the block diagram may be described below in connection with certain implementation embodiments presented in other figures, those implementations are provided for illustrative purposes only, and are not intended to serve as a limitation on the block diagram. For example, the steps in Fig. 9 may be executed in any order, steps may be duplicated, or steps may be omitted. In some embodiments, the process 900 may be performed by at least one processor (e.g., a CPU) of a computing device or system (e.g., scanning system

101 in Fig. 1) to perform operations or functions described herein, and may be described hereinafter with reference to Figs. 1-2D by way of example. In some embodiments, some aspects of the process 900 may be implemented as software (e.g., program codes or instructions) that are stored in a memory or a non-transitory computer-readable medium. In some embodiments, some aspects of the process 300 may be implemented as hardware (e.g., a specific-purpose circuit). In some embodiments, the process 300 may be implemented as a combination of software and hardware.

[00340] Fig. 9 includes process blocks 902-908. At block 902, at least one processor may analyze a cloud environment to identify a plurality of keys to the compute resources in the cloud environment. In some embodiments, the plurality of keys may be stored in at least one workload. In some embodiments, at least one of the plurality of keys may include at least one of a password (e.g., a combination of alphanumeric characters and symbols), a script (e.g., a PYTHON® script or a shell script) containing a password, a cloud key, or an Secure Shell (SSH) key.

[00341] At block 904, the at least one processor may perform a cryptographic analysis on the plurality of keys to identify a first set of fingerprints that uniquely identify each of the plurality of keys. The first set of fingerprints may, in some embodiments, be non-functional. In some embodiments, at least one of the first set of fingerprints may be non-identical to the key.

[00342] At block 906, the at least one processor may analyze trust configurations of the compute resources to identify a second set of fingerprints for each of the compute resources. As discussed above, to identify the second set of fingerprints, in some embodiments, the at least one processor may analyze a trust configuration file (e.g., a file storing fingerprints and their associated compute resources) by reading the byte stream of the trust configuration file to determine whether any sequence of bytes that fits to a predetermined pattern (e.g., an alphanumeric string pattern). The sequences of bytes that fit to the predetermined pattern may be determined as the second set of fingerprints.

[00343] At block 908, the at least one processor may compare the first set of fingerprints with the second set of fingerprints to match keys with the compute resources without using the keys to access the compute resources. As discussed above, in some embodiments, a first compute resource may store a key (e.g., a private component of a private-public key pair) to a second compute resource, and the second compute resource may store in its trust configurations a second fingerprint of the key. The at least one processor may analyze the cloud environment to identify the key from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. The at least one processor may then analyze the trust configurations of the second compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other (e.g., being identical). If the first

fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource.

[00344] Consistent with disclosed embodiments, besides blocks 902-908, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the first set of fingerprints. By way of example, to analyze the multi-machine interaction, the at least one processor may compare the first set of fingerprints with the second set of fingerprints. As discussed above, in some embodiments, a first compute resource may store a key (e.g., a private key) for accessing a second compute resource. The at least one processor may analyze the cloud environment to identify the private component from the first compute resource, then may perform the cryptographic analysis described herein to identify a first fingerprint of the key. A third compute resource (e.g., may be the same as or different from the second compute resource) may store in its trust configurations a second fingerprint of the key. The at least one processor may then analyze the trust configurations of the third compute resource to identify the second fingerprint. The at least one processor may then compare the first fingerprint and the second fingerprint to determine whether they match each other (e.g., being the same). If the first fingerprint matches with the second fingerprint, the at least one processor may determine that the key identified from the first compute resource is a key for accessing the second compute resource, and may further determine an inter-machine interaction in which the first compute resource has ability to access the second compute resource.

[00345] Consistent with disclosed embodiments, besides blocks 902-908, the at least one processor may further analyze a multi-machine interaction in the cloud environment using the plurality of keys. By way of example, to analyze the multi-machine interaction, the at least one processor may compare a first key for accessing a first one of the compute resources and a second key for accessing a second one of the compute resources.

[00346] **Hybrid Ephemeral Scanner Techniques**

[00347] Aspects of this disclosure may include accessing a primary account maintained in a cloud environment. A primary account as used herein, may refer to a principal or main identity created for a person in a computer or computing system. A primary account may also be created for machine entities, such as service accounts for running programs, system accounts for storing system files and processes, and root and administrator accounts for system administration. A cloud environment, as used herein, may refer to a platform implemented on, hosted on, and/or accessing servers that are accessed over the Internet. An example of a cloud environment is cloud infrastructure 106 in Fig. 1. In some embodiments, scanning system 101 of Fig. 1 may gain access to an account within cloud infrastructure 106.

[00348] Aspects of this disclosure may include receiving information defining a structure of the primary account, wherein the structure includes a plurality of assets.

Information as used herein, may refer to data received. Structure of the primary account as used herein, may refer to a set up or of the primary account. A plurality of assets as used herein, may refer to one or more data, devices, or components within an organization's system. For example, the plurality of assets may include any of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, load balancer 115, log files or databases, API gateway resources, API gateway REST APIs, Autoscaling groups, CloudTrail logs, CloudFront services, volumes, snapshots, VPCs, subnets, route tables, network ACLs, VPC endpoints, NAT gateways, ELB and ALB, ECR repositories, ECS clusters, services, and tasks, EKS, S3 bucket and Glacier storage, SNS topics, IAM roles, policies, groups, users, KMS keys, and Lambda functions.

[00349] In some embodiments, scanning system 101 of Fig. 1 may receive information relating to the structure of an account. In such an embodiment, the structure of the account may include virtual machines 107A-D and databases 109A-109D, for example. In some embodiments, the information may be acquired from another device, received via a one way communication, received in response to a request, retrieved from a storage device, or generated.

[00350] In some embodiments, the information may exclude raw data of the primary account. Raw data as used herein, may refer to a collection of data as gathered before it has been processed, cleaned, or analyzed. For example, raw data may include usage data, passwords, and cache.

[00351] Aspects of this disclosure may include deploying, inside the primary account, at least one ephemeral scanner configured to scan at least one block storage volume and to output metadata defining the at least one block storage volume. In some embodiments, the output may exclude raw data of the primary account. In some embodiments, the ephemeral scanner may be deployed inside a secondary account for which trust has been established with a primary account. For example, a secondary account on another system may have an existing trust relationship (such as a cloud trust policy) with a first account. The ephemeral scanner may be operated inside of that secondary account to scan block storage associated with the first account. Each account may be hosted on the same system or on different systems.

[00352] Further, in some embodiments, the at least one ephemeral scanner may be configured to periodically scan at least one block storage volume. For example, the scanner may be configured to scan at least one block storage volume once per hour, once per day, once per week, or more or less often.

[00353] Further, in some embodiments, the at least one ephemeral scanner may be periodically deployed to scan at least one block storage volume. For example, the scanner may be deployed for scanning at least one block storage volume once per hour, once per day, once per week, or more or less often.

[00354] Outputting metadata as used herein, may refer to producing a set of data that describes and gives information about other data. In some embodiments, scanning system 101 of Fig. 1 may scan a block storage volume and output data related to it (e.g., vulnerability information, configuration information, malware information, risk analysis information, and sensitive information). Periodically scanning as used herein, may refer to conducting a scan occurring or recurring at regular intervals. In some embodiments, the ephemeral scanner is configured to perform vulnerability scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for security vulnerabilities on the device. In some embodiments, the ephemeral scanner is configured to perform configuration scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for configuration information related to the device. In some embodiments, the ephemeral scanner may scan the block storage volume in order to look for security issues. In some embodiments, the ephemeral scanner is configured to perform malware scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for malware information related to the device. As discussed above with respect to Fig. 2D, in step 235, scanning system 101 may perform a step of malware scanning. In some embodiments, scanning system 101 may perform malware scanning across all filesystems in the snapshot (e.g., gathered from virtual machines 107A-107D or storage 111A-111D). Scanning system 101 may use multiple malware scanning software solutions to perform a malware scan against the filesystems, including one sourced from another vendor, such as bucketAV, Trend Micro Cloud One, Sophos Cloud Optix, Crowdstrike Falcon CWP, or others. In some embodiments, malware scanning in step 235 comprises utilizing signatures, heuristics, or sandboxing capabilities to deduce whether there is an infection on the machine.

[00355] In some embodiments, the ephemeral scanner is configured to perform lateral-movement risk analysis of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for lateral-movement risk information related to the device. For example, in some embodiments, scanning system 101 may perform a “backward” analysis of the specific asset to identify exposure risk to assets downstream of the specific asset, wherein the downstream exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and lateral movement risks associated with the exposed asset.

[00356] Further, as discussed above with respect to Fig. 2D, in step 237, scanning system 101 may perform a step of lateral movement scanning. An attacker who establishes a network foothold usually attempts to move laterally from one resource to another in search of rich targets such as valuable data. Stolen passwords and keys unlock access to servers, files, and privileged accounts. In some embodiments, scanning system 101 may gather keys from

each scanned system or device (e.g., virtual machines 107A-107D or storage 111A-111D). In some embodiments, scanning system 101 searches for passwords, scripts, shell history, repositories, or other data that may contain passwords, cloud access keys, SSH keys, or other key/password/access information that provide unchecked access to important resources. In some embodiments, scanning system 101 searches for such keys/passwords/access information and calculates a "hash" (a mathematical fingerprint) of each string. Scanning system 101 then attempts to match the hashed strings to hashes of strings that that are stored on different systems or devices. This will be used to detect the potential lateral movement between assets.

[00357] In some embodiments, the ephemeral scanner is configured to perform sensitive information scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for sensitive information related to the device. As discussed above with respect to Fig. 2D, in step 241, scanning system 101 may perform a step of sensitive information scanning. In some embodiments, scanning system 101 may search the snapshot for sensitive information, such as personally identifiable information (PII), Social Security numbers, healthcare information, or credit card numbers. In some embodiments, scanning system 101 may search data repository history as well. This is because it is not uncommon for an entire production environment repository to be cloned, with no one remembering the copy contains sensitive information. In some situations, detecting sensitive data not secured is critical in adherence to data privacy regulations.

[00358] In some embodiments, the ephemeral scanner is configured to perform container scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for container information related to the device. As discussed with respect to Fig. 2D, in step 243, scanning system 101 may perform a step of container scanning. In some embodiments, scanning system 101 may apply one or more of the preceding steps of Fig. 2D against containerized environments. In some embodiments, in order to do so, scanning system 101 reconstructs a container runtime layered file system (LFS) before recursively running one or more of steps 231-241 on the reconstructed file system.

[00359] In some embodiments, the ephemeral scanner is configured to perform keys and password scanning of the at least one block storage volume. In such an embodiment the ephemeral scanner may scan the block storage volume to check for keys and password information related to the device. As discussed above with respect to Fig. 2D, in step 239, scanning system 101 may perform a step of key/password scanning. As one example situation, suppose there is a weak or unprotected password stored (in plain text) in storage 111A. For example, if a personal email account has been compromised, the passwords may be known about in advance. Scanning system 101 may search the snapshot for similar usernames or login names, and, either using known dictionaries or the account owner's previously leaked

passwords (stored in, e.g., database 103A), may attempt to login to one or more systems or devices in cloud infrastructure 106, and may record the result thereof.

[00360] In some embodiments, defining the at least one block storage volume includes presenting risk data without sharing consumer data or data that was used to identify the risk data. Risk data as used herein, may refer to information relating to causing harm. Consumer data as used herein, may refer to a user's personal information or any information trail user's leave behind as a result of their computer or Internet use. Data used to identify the risk as used herein, may refer to information for determining possible issues. In some embodiments, scanning system 101 of Fig. 1 may present information without sharing a user's personal information or any information trail user's leave behind as a result of their computer or Internet use.

[00361] In some embodiments, the metadata defining the at least one block storage volume includes at least one of: an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, or a profile configuration. An indication of an installed application as used herein, may refer to a sign or signal of a stored program. A version of an installed application as used herein, may refer to an older or newer form of a stored program. An operating system configuration as used herein, may refer to one or more computer system settings that have been set by default automatically or manually by a given program or the user. An application configuration as used herein, may refer to one or more computer program settings that have been set by default automatically or manually by a given program or the user. A profile configuration as used herein, may refer to one or more computer program file settings that have been set by default automatically or manually by a given program or the user.

[00362] Aspects of this disclosure may include receiving a transmission of the metadata from the at least one ephemeral scanner. In some embodiments, the transmission may exclude raw data of the primary account. A transmission of metadata as used herein, may refer to movement of a set of data that describes and gives information about other data. In some embodiments, scanning system 101 of Fig. 1 may receive data excluding any raw data (e.g., data as gathered before it has been processed, cleaned or analyzed).

[00363] Aspects of this disclosure may include analyzing the received metadata to identify a plurality of cybersecurity vulnerabilities. A plurality of cybersecurity vulnerabilities as used herein, may refer to a weakness in a system. For example, a cybersecurity vulnerability may be exploited by cybercriminals to gain unauthorized access to a computer system. In some embodiments, scanning system 101 of Fig. 1 may analyze the received data in order to identify any weakness in a system (e.g., to avoid cybercriminals gaining unauthorized access to a computer system). In some embodiments, vulnerability information may include information stored in a system.

[00364] Aspects of this disclosure may include correlating each of the identified plurality of cybersecurity vulnerabilities with one of the plurality of assets. In some embodiments, scanning system 101 of Fig. 1 may connect a security risk with an asset in the cloud environment. In such embodiments, the connection or correlation of vulnerability to asset may allow the system to address the vulnerability.

[00365] Aspects of this disclosure may include generating a report correlating the plurality of cybersecurity vulnerabilities with the plurality of assets. A report as used herein, may refer to a document containing information. For example, a report may notify the administrator of a website or application about a problem such as a security issue or vulnerability in the system that should be addressed. In some embodiments, scanning system 101 of Fig. 1 may create a report providing information related to the vulnerability and asset. In such an embodiment, the report may allow for the vulnerability to be addressed and resolved.

[00366] In such embodiments, the system may allow scanners (e.g., ephemeral scanners, scanning system 101) to run inside an account, including an Amazon Web Services, Azure, or GCP account. In this mode, a cloud cybersecurity service may generate ephemeral scanners inside an account that performs the same actions as a SAAS node but are logically hosted inside the account.

[00367] In some embodiments, the at least one processor is further configured to receive a transmission of updated metadata defining the at least one block storage volume in response to at least one change to the at least one block storage volumes. Updated metadata as used herein, may refer to a renewed set of data that describes and gives information about other data.

[00368] **Fig. 10** is a block diagram of a method 1000 for deployment of ephemeral scanners, consistent with disclosed embodiments. In some embodiments, the method may include seven (or more or less) steps:

[00369] Block 1002: Access a primary account maintained in a cloud environment. In some embodiments, scanning system 101 of Fig. 1 may gain access to an account within cloud infrastructure 106 using a password stored in a database, for example.

[00370] Block 1004: Receiving information defining a structure of the primary account, wherein the structure includes a plurality of assets. In some embodiments, the information may exclude raw data of the primary account. In some embodiments, scanning system 101 of Fig. 1 may receive data relating to the structure of an account. In such an embodiment, the structure of the account may include virtual machines 107A-D and databases 109A-109D, for example.

[00371] Block 1006: Deploying, inside the primary account, at least one ephemeral scanner configured to periodically scan at least one block storage volume and to output metadata defining the at least one block storage volume. In some embodiments, the output may exclude raw data of the primary account. In some embodiments, scanning system 101 of Fig. 1

may scan a block storage volume and output data related to it (e.g., vulnerability information, configuration information, malware information, risk analysis information, and sensitive information).

[00372] Block 1008: Receiving a transmission of the metadata from the at least one ephemeral scanner. In some embodiments, the transmission may exclude raw data of the primary account. In some embodiments, scanning system 101 of Fig. 1 may receive data excluding any raw data (e.g., data as gathered before it has been processed, cleaned or analyzed).

[00373] Block 1010: Analyzing the received metadata to identify a plurality of cybersecurity vulnerabilities. For example, a cybersecurity vulnerability may be exploited by cybercriminals to gain unauthorized access to a computer system. In some embodiments, scanning system 101 of Fig. 1 may analyze the received data in order to identify any weakness in a system (e.g., to avoid cybercriminals gaining unauthorized access to a computer system).

[00374] Block 1012: Correlating each of the identified plurality of cybersecurity vulnerabilities with one of the plurality of assets. In some embodiments, scanning system 101 of Fig. 1 may connect a security risk with an asset in the cloud environment. In such embodiments, the connection or correlation of vulnerability to asset may allow the system to address the vulnerability.

[00375] Block 1014: Generating a report correlating the plurality of cybersecurity vulnerabilities with the plurality of assets. For example, a report may notify the administrator of a website or application about a problem such as a security issue or vulnerability in the system that should be addressed. In some embodiments, scanning system 101 of Fig. 1 may create a report providing information related to the vulnerability and asset. In such an embodiment, the report may allow for the vulnerability to be addressed and resolved.

[00376] **Risk Information Aggregation Techniques**

[00377] **Fig. 11** represents a schematic block diagram 1100 illustrating an exemplary embodiment of a method for providing a dashboard aggregating risk information.

[00378] Assets operating in a cloud environment face myriad cybersecurity risks that vary in type and nature of threat with increasing regularity. In order to combine the varied and multiple cybersecurity risks an asset may encounter in a cloud environment into a single view to allow for faster and more flexible processing, prioritization and mitigation, a system and method of aggregating cybersecurity risks into a single dashboard is needed. Aggregation of multiple cybersecurity risks into a single view for an administrator is needed. In one of many embodiments, an exemplary embodiment may include a graphical user interface system for providing comprehensive cloud environment risk inventory visualization. A graphical user interface system may include systems similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows

10 Mobile, PalmOS-Web OS, or Firefox OS. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of risk inventories. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D.

[00379] An exemplary embodiment may also include at least one processor. The at least one processor may be part of a personal computer, tablet, smartphone, or virtual machine for processing threats and vulnerabilities to a cloud-based storage volume. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing logic operations.

[00380] In an embodiment may include a processor configured to cause a display to present a plurality of asset categories (step 1101). A display may be graphical, sequential, or multi-dimensional. A plurality may include at least one asset category to be displayed. Asset categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category.

[00381] An account category may further include listing of assets based on type of account associated with that user. A type of account may include active user account, passive user account, administrator account, maintainer account, a system account, superuser account, or a guest user account.

[00382] A container category may include one or more containers currently listed in the system. A container may involve specific versions of programming language runtimes, libraries required to run software, or another method of packaging applications abstracted from the cloud environment.

[00383] A database category may include one or more databases managed and maintained in relation to the system. Databases may include a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database.

Databases may be implemented using ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances.

[00384] An image category may contain one or more images based on image type, size, and content. Images may include images of one or more virtual machines.

[00385] A managed service category may include one or more managed services listed in the above system, to include at least client or customer-owned systems being managed by a third party entity.

[00386] A messaging service category may include systems/programs communicating with other systems/programs using services such as Google Cloud Pub and Sub communications, AWS SQS, or similar queues found in Information Systems.

[00387] A monitoring category may include centralized or decentralized methods of security monitoring systems.

[00388] A network category may include one or more systems of establishing a form of communicative connectivity between systems, e.g., TCP-IP services.

[00389] A storage category may list one or more methods of data storage, to include Direct Attached Storage, Network Attached Storage, SSD Flash Drive Arrays, Hybrid Flash Arrays, Hybrid Cloud Storage, Backup Software, Backup Appliances, Cloud Storage, or similar. Storage may include data structures, instructions, or any other data to be contained in a storage medium.

[00390] A user category may include one or more users of a given system along with identifying information relating to its unique or semi-unique identifier, level of authorized access and permissions, and storage volumes accessible by said user.

[00391] An access category may include one or more levels of access permission and a listing of users or devices granted access at a given level of access permission to a system.

[00392] A virtual machine category may list one or more virtual machines accessible to a system and the levels and types of access granted to a given virtual machine.

[00393] A serverless category may list one or more storage volumes stored on a serverless computing system such as Kubeless, Fission, Google App Engine, AWS Lambda, Google Cloud Functions, IBM Cloud Function, IBM Cloud Code Engine, Microsoft Azure Functions, Cloudflare Workers, and Compute@Edge.

[00394] In an embodiment, an exemplar may include a processor configured to receive, via an input device, a selection of a particular asset category (step 1103). An input device may include a personal computer, tablet, smartphone, or virtual machine. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing

logic operations. A selection of one or more asset categories may be received and updated as new visualizations are developed based on the current threats to an asset in that category. Asset categories may be selected or deselected by an end user. Once a particular asset category is selected, it may be added to a list of selected asset categories with a list for deselected categories as well. The list of selected asset categories may organized per category or may be compiled to include all current threats in a comprehensive selected asset threat list.

[00395] One exemplary embodiment may further include a processor configured to cause the display to present a list of assets in the selected category that have cyber security risks (step 1105). When an asset in a selected category has a cyber security risk associated with it, the asset may appear in a list of all listed categories having a cyber security risk associated with it, and may be organized by individual category, type of cyber security risk, or some combination or sequential ordering methods based on the information available to the processor and its end user.

[00396] In one of many exemplary embodiments a processor may be configured to retrieve workload component cybersecurity risk information (step 1107A) for each listed asset. When a category of assets is selected, a processor may be configured to order those assets according to available cybersecurity risk information available to said processor, and may organize by type of risk, type of asset, or similar type of sequentialization. Cybersecurity risk information may be determined from historical threats to the workload component, the current active cybersecurity threats to workload components of certain types, or some combination of the same.

[00397] In an embodiment, a processor may be configured to retrieve cloud component cybersecurity risk information (step 1107B) for each listed asset. Cloud component cybersecurity risk information may include identification of the risk based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset. Cybersecurity risk information to a cloud component may be determined from historical threats within the cloud environment 106, current active cybersecurity threats to similar workload components in the cloud environment 106 or the level of risk to assets within the cloud environment 106, as outlined below.

[00398] An exemplary embodiment may include a system configured to cause the display to present a common interface (step 1109) providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information. The common interface may be configured to access component cybersecurity risk information based on the nature of the risk, the likelihood of the risk, or the severity of the risk to the component, including recursive risk – risk that can happen from an attacker getting access to a system and using it as a mean to reach other system. On the same interface, a different interface, or a similar and related interface, a display may present cloud component cybersecurity risk information based

on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset.

[00399] As an exemplary embodiment, the system may be configured to cause the display to present, in the common interface, an interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information (step 1111). With this display presented, an exemplary embodiment may allow for a user to prioritize data from additional sources. The interconnection between workload component cybersecurity risk information and cloud component cybersecurity risk information may generate an interface between them and may be presented as an interconnection between them. The common interface presented may include direct interconnections between the component cybersecurity risk information and the cloud component cybersecurity risk information or links from one set of cybersecurity risk information. For instance, the component cybersecurity risk information may include a link to access the cloud component cybersecurity risk information. Conversely, the cloud component cybersecurity risk information may include a link to access the component cybersecurity risk information.

[00400] In one embodiment, a system may be configured to provide the plurality of asset categories include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. A category may list the currently identified entities of that category type as described below. An account category may include one or more listed accounts of users of the system. An account category may further include listing of assets based on type of account associated with that user. A type of account may include active user account, passive user account, administrator account, maintainer account, a system account, superuser account, or a guest user account.

[00401] A container category may include one or more containers currently listed in the system. A container may involve specific versions of programming language runtimes, libraries required to run software, or another method of packaging applications abstracted from the cloud environment.

[00402] A database category may include one or more databases managed and maintained in relation to the system. Databases may include a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database. Databases may be implemented using ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances.

[00403] An image category may contain one or more images based on image type, size, and content. Images may include images of one or more virtual machines displayed. A

managed service category may include one or more managed services listed in the above system, to include at least client or customer-owned systems being managed by a third party entity.

[00404] A messaging service category may include services of users communicating with at least one other user such as Google Cloud Pub and Sub communications, AWS SQS, or similar queues found in Information Systems.

[00405] A monitoring category may include centralized or decentralized methods of security monitoring systems.

[00406] A network category may include one or more systems of establishing a form of communicative connectivity between systems, e.g., TCP-IP services.

[00407] A storage category may list one or more methods of data storage, to include Direct Attached Storage, Network Attached Storage, SSD Flash Drive Arrays, Hybrid Flash Arrays, Hybrid Cloud Storage, Backup Software, Backup Appliances, Cloud Storage, or similar. Storage may include data structures, instructions, or any other data to be contained in a storage medium.

[00408] A user category may include one or more users of a given system along with identifying information relating to its unique or semi-unique identifier, level of authorized access and permissions, and storage volumes accessible by said user.

[00409] An access category may include one or more levels of access permission and a listing of users or devices granted access at a given level of access permission to a system.

[00410] A virtual machine category may list one or more virtual machines accessible to a system and the levels and types of access granted to a given virtual machine.

[00411] A serverless category may list one or more storage volumes stored on a serverless computing system such as Kubeless, Fission, Google App Engine, AWS Lambda, Google Cloud Functions, IBM Cloud Function, IBM Cloud Code Engine, Microsoft Azure Functions, Cloudflare Workers, and Compute@Edge.

[00412] In one of many exemplary embodiments, a system may further include the common interface configured to display information relating to at least one of an asset type, a risk, a region, or an account. A common interface may include the ability to configure and reconfigure the asset type, risk, region, or account to appear in any order and at multiple formats for displaying the information relating to those categories.

[00413] In another embodiment, a system may provide the common interface configured to display description for each listed asset. The display of a description for each listed asset may be listed vertically or horizontally. The display may be displayed as written text, representative graphical figures, or similar method of displaying the aforementioned types of information for an asset.

[00414] In another embodiment, a system may further provide the common interface is configured to display at least one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, or an authentication. A vulnerability may include vulnerability data from a vulnerability database such as NVD, WPVulnDB, US-CERT, Node.js Security Working Group, OVAL – Red Hat, Oracle Linux, Debian, Ubuntu, SUSE, Ruby Advisory Database, JVN, Safety DB(Python), Alpine secdb, PHP Security Advisories Database, Amazon ALAS, RustSec Advisory Database, Red Hat Security Advisories, Microsoft MSRC, KB, Debian Security Bug Tracker, Kubernetes security announcements, Exploit Database, Drupal security advisories, JPCERT. An insecure configuration may include software flaws or misconfigurations, non-encrypted files, improper file or directory permissions, unpatched security flaws in server software, enabled or accessible administrative and debugging functions, administrative account vulnerabilities, SSL certificates and encryption settings not properly configured, or a similar misconfiguration. These misconfigurations may be discovered by performing a scan, e.g., may query devices and systems capable of routing traffic (e.g., load balancer 115, routers, switches, firewalls, and proxies) using an API provided through a cloud service provider's system to determine network configurations, and may evaluate them against known problematic configurations or other configurations. Malware indicated may include Adware, Botnets, Cryptojacking, Malvertising, Ransomware, Remote Administration Tools (RATs), Rootkits, Spyware, Trojans, Virus Malware, Worm Malware, or similar attack vehicles. A neglected asset may include an asset that has been improperly maintained, patched, or similar cybersecurity security measure. A data at risk may include any packet of data that may be exposed to a cybersecurity threat. A lateral movement may include any pathway between assets where a cybersecurity risk can travel from one affected asset to another affected asset. An authentication may include a username, password, one-time password, two-factor authentication, or any other authentication mechanism to gain access to a cloud service provider's system. A common interface may display more than one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, and an authentication separately or concurrently for an asset. A display of said information may be ordered and listed in any order depending on user preference and input.

[00415] An exemplary embodiment may further include a system wherein the common interface is configured to display one or more possible attack vectors reaching the each listed asset. Generating this common interface display may include generating a map of an asset to include recording information such as a region identifier, site identifier, datacenter identifier, physical address, network address, workload name, or any other identifier which may be acquired via an API provided through a cloud service provider's system and demonstrating via a display the myriad attack vectors currently available to an analyzed asset. Display of the one or

more possible attack vectors reaching the each listed asset may be displayed as written text, representative graphical figures, or similar method of displaying the one or more possible attack vectors.

[00416] Another exemplary embodiment may further include a system wherein the common interface is configured to display a recommended mitigation tactic for the each listed asset. A recommended mitigation tactic may be generated based on historical data of the cybersecurity risk, the vulnerability attributed to the listed asset, or the data contained on the each listed asset. Mitigation tactics may be selected based on user input or may be implemented automatically based on the mitigation tactics likelihood of a successful mitigation of the cybersecurity risk. A mitigation tactic may include increased frequency of scanning, heightened access control measures, firewalls and antivirus software patches, increased patch management scheduling, continuous workload monitoring, or similar tactics.

[00417] In another exemplary embodiment is a system wherein the common interface is configured to display one or more workload metrics associated with the each listed asset. Workload metrics may be a statistical representation of the current performance of the workload, the capacity for additional processing of the workload, and the historical performance of the workload based on statistical trend.

[00418] Another exemplary embodiment may include a system wherein the at least one processor is further configured to provide a cybersecurity report for the each listed asset. A cybersecurity report may be generated by scanning system 101 following a complete scan of an analyzed asset, and may include a snapshot of one or more of the number of total cybersecurity threats to a listed asset, number of cybersecurity threats mitigated by recent performance, methods of cybersecurity risk mitigation in the listed asset, rate of cybersecurity threats detected in the listed asset, and possibility of failed cybersecurity mitigation tactics based on the frequency and nature of the historical cybersecurity threats. Data collected to generate a cybersecurity report may include, among many things, operating system packages, installed software applications, libraries, and program language libraries such as Java archives, Python packages, Go modules, Ruby gems, PHP packages, and Node.js modules, other software applications, library versions, software versions, and other identifying characteristics of software and operating systems, lists of users of each system or device (e.g., virtual machines 107A-D), each system's or device's services, password hashes, and application-specific configurations for software/services such as Apache, Nginx, SSH, and other services, bugs or other configuration risks, malware scan results, passwords, scripts, shell history, repositories, or other data that may contain passwords, cloud access keys, SSH keys, or other key/password/access information that provide unchecked access to important resources.

[00419] In another embodiment, a system may include the at least one processor is configured to identify a risk level distribution among the listed assets. The at least one processor

may be configured to list risk level distribution by highest level of risk, lowest level of risk, or largest number of cybersecurity risks attributed among the listed assets. A risk level may be high if the web service is connected to the Internet (e.g., has at least one public port forwarded through a firewall to the web service, is able to be accessed through a load balancer, is able to be accessed via reverse proxy). A risk level may be medium risk if the web service is only accessible internally (e.g., because of firewall configuration). A risk level may be low risk if access to the web server is blocked by a configuration of cloud infrastructure 106.

[00420] Another exemplary embodiment may include a system wherein the at least one processor is configured to receive a search query for a specific risk. A search query may be input by type of risk, name of a risk, or another semi-unique manner of identifying a specific cybersecurity risk. A search may be initiated by end user operating user device 102, by a maintainer or administrator of an analyzed asset, or by a virtual machine associated with operating scanning system 101, A search may be initiated by inputting a number of inputs, to include keywords, phrases, file type, or similar identifiers unique or semi-unique to a given cybersecurity risk. Upon initiating a search, the results generated by the search may be stored in the analyzed asset, transmitted to the operator of a processor running side scanning system 101, or generated for review by an end user operating user device 102. The search and display of a specific risk may be based on historical trends of cybersecurity risks in the listed asset, severity of cybersecurity risk to a listed asset, or a historical search for past cybersecurity threats to a listed asset.

[00421] Another embodiment may include a system wherein the at least one processor is configured to identify one or more assets vulnerable to the specific risk. For instance, a processor operating scanning system 101 may be configured to perform a scan of a set of one or more assets with potential exposure to a given risk, and upon identifying vulnerable pathways for an attack vector to reach a set of identified assets, may generate a list of assets that are potentially or currently exposed to one or more specific risk. Assets vulnerable to a specific risk may be determined to be vulnerable based on the type of the specific risk, the existence of specific risks in adjacent or similar workloads, and related risks that may also cause an asset to be vulnerable based on the specific risk determined.

[00422] In another embodiment is a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing a display to present a plurality of asset categories (step 1101). The method may include the use of a graphical user interface system and may include systems similar in appearance and functionality to Microsoft Windows, Mac OS, Ubuntu Unity, Gnome Shell, Android, Apple iOS, Blackberry OS, Windows 10 Mobile, PalmOS-Web OS, or Firefox OS. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and

the cloud-based storage volume may be contained in storage 111A-111D. The cloud environment may include service provided by a third party such as Amazon Web Services, Microsoft Azure, Google Cloud, Alibaba Cloud, IBM Cloud, Oracle, Salesforce, SAP, or similar service. Visualizations provided by the above system may include graphical, sequential, or multi-dimensional displays of risk inventories. For example, cloud infrastructure 106 may be the cloud environment consisting of virtual machines 107A-107D, databases 109A-109D, storage 111A-111D, keystores 113A-113D, and load balancer 115, and the cloud-based storage volume may be contained in storage 111A-111D.

[00423] One of several embodiments may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising receiving, via an input device, a selection of a particular asset category. A method may include a personal computer, tablet, smartphone, or virtual machine. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing logic operations. A selection of one or more asset categories may be received and updated as new visualizations are developed based on the current threats to an asset in that category. Asset categories may be selected or deselected by an end user. Once a particular asset category is selected, it may be added to a list of selected asset categories with a list for deselected categories as well. The list of selected asset categories may organized per category or may be compiled to include all current threats in a comprehensive selected asset threat list.

[00424] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing the display to present a list of assets in the selected category that have cybersecurity risks (step 1105). A display may be graphical, sequential, or multi-dimensional. A list of selected categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. When an asset in a selected category has a cyber security risk associated with it, the asset may appear in a list of all listed categories having a cyber security risk associated with it, and may be organized by individual category, type of cyber security risk, or some combination or sequential ordering methods based on the information available to the processor and its end user.

[00425] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising retrieving workload component cybersecurity risk information 307A for each listed asset. When a category

of assets is selected, a processor may be configured to order those assets according to available cybersecurity risk information available to said processor, and may organize by type of risk, type of asset, or similar type of sequentialization. Cybersecurity risk information may be determined from historical threats to the workload component, the current active cybersecurity threats to workload components of certain types, or some combination of the same.

[00426] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising retrieving cloud component cybersecurity risk information 307B. Cloud component cybersecurity risk information may include identification of the risk based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset. Cybersecurity risk information to a cloud component may be determined from historical threats within the cloud environment 106, current active cybersecurity threats to similar workload components in the cloud environment 106 or the level of risk to assets within the cloud environment 106, as outlined above.

[00427] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing the display to present a common interface 309 providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information. The common interface may be configured to access component cybersecurity risk information based on the nature of the risk, the likelihood of the risk, or the severity of the risk to the component. On the same or a similar and related interface, a display may present cloud component cybersecurity risk information based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset.

[00428] Another embodiment may include a method for providing comprehensive cloud environment risk inventory visualization in a graphical user interface comprising causing the display to present, in the common interface, an interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information. The interconnection between workload component cybersecurity risk information 307A and cloud component cybersecurity risk information 307B may generate an interface between them 309 and may be presented as an interconnection between them 311. The common interface presented may include direct interconnections between the component cybersecurity risk information and the cloud component cybersecurity risk information or links from one set of cybersecurity risk information. For instance, the component cybersecurity risk information may include a link to access the cloud component cybersecurity risk information. Conversely, the cloud component cybersecurity risk information may include a link to access the component cybersecurity risk information.

[00429] In another embodiment, a method may include wherein the plurality of asset categories include at least one of an account category, an authentication category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. A category may list the currently identified entities of that category type as described below. An account category may include one or more listed accounts of users of the system. An account category may further include listing of assets based on type of account associated with that user. A type of account may include active user account, passive user account, administrator account, maintainer account, a system account, superuser account, or a guest user account.

[00430] A container category may include one or more containers currently listed in the system. A container may involve specific versions of programming language runtimes, libraries required to run software, or another method of packaging applications abstracted from the cloud environment.

[00431] A database category may include one or more databases managed and maintained in relation to the system. Databases may include a NoSQL database, a relational database, a cloud database, a columnar database, a wide column database, a key-value database, an object-oriented database, a hierarchical database, or any other kind of database. Databases may be implemented using ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS, Aurora, Redshift clusters, Kafka clusters, or EC2 instances.

[00432] An image category may contain one or more images based on image type, size, and content. Images may include images of one or more virtual machines displayed.

[00433] A managed service category may include one or more managed services listed in the above system, to include at least client or customer-owned systems being managed by a third party entity.

[00434] A messaging service category may include services of users communicating with at least one other user such as Google Cloud Pub and Sub communications, AWS SQS, or similar queues found in Information Systems.

[00435] A monitoring category may include centralized or decentralized methods of security monitoring systems.

[00436] A network category may include one or more systems of establishing a form of communicative connectivity between systems, e.g., TCP-IP services.

[00437] A storage category may list one or more methods of data storage, to include Direct Attached Storage, Network Attached Storage, SSD Flash Drive Arrays, Hybrid Flash Arrays, Hybrid Cloud Storage, Backup Software, Backup Appliances, Cloud Storage, or similar.

Storage may include data structures, instructions, or any other data to be contained in a storage medium.

[00438] A user category may include one or more users of a given system along with identifying information relating to its unique or semi-unique identifier, level of authorized access and permissions, and storage volumes accessible by said user.

[00439] An access category may include one or more levels of access permission and a listing of users or devices granted access at a given level of access permission to a system.

[00440] A virtual machine category may list one or more virtual machines accessible to a system and the levels and types of access granted to a given virtual machine.

[00441] Another embodiment may include a method wherein the common interface is configured to display information relating to at least one of an asset type, a risk, a region, or an account. A common interface may include the ability to configure and reconfigure the asset type, risk, region, or account to appear in any order and at multiple formats for displaying the information relating to those categories.

[00442] Another embodiment may include a method wherein the common interface is configured to display description for each listed asset. The display of a description for each listed asset may be listed vertically or horizontally. The display may be displayed as written text, representative graphical figures, or similar method of displaying the aforementioned types of information for an asset.

[00443] Another embodiment may include a method wherein the common interface is configured to display at least one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, or an authentication. A vulnerability may include vulnerability data from a vulnerability database such as NVD, WPVulnDB, US-CERT, Node.js Security Working Group, OVAL – Red Hat, Oracle Linux, Debian, Ubuntu, SUSE, Ruby Advisory Database, JVN, Safety DB(Python), Alpine secdb, PHP Security Advisories Database, Amazon ALAS, RustSec Advisory Database, Red Hat Security Advisories, Microsoft MSRC, KB, Debian Security Bug Tracker, Kubernetes security announcements, Exploit Database, Drupal security advisories, JPCERT. An insecure configuration may include software flaws or misconfigurations, non-encrypted files, improper file or directory permissions, unpatched security flaws in server software, enabled or accessible administrative and debugging functions, administrative account vulnerabilities, SSL certificates and encryption settings not properly configured, or a similar misconfiguration. These misconfigurations may be discovered by performing a scan, e.g., may query devices and systems capable of routing traffic (e.g., load balancer 115, routers, switches, firewalls, and proxies) using an API provided through a cloud service provider's system to determine network configurations, and may evaluate them against known problematic configurations or other configurations. Malware indicated may include Adware, Botnets, Cryptojacking, Malvertising,

Ransomware, Remote Administration Tools (RATs), Rootkits, Spyware, Trojans, Virus Malware, Worm Malware, or similar attack vehicles. A neglected asset may include an asset that has been improperly maintained, patched, or similar cybersecurity security measure. A data at risk may include any packet of data that may be exposed to a cybersecurity threat. A lateral movement may include any pathway between assets where a cybersecurity risk can travel from one affected asset to another affected asset. An authentication may include a username, password, one-time password, two-factor authentication, or any other authentication mechanism to gain access to a cloud service provider's system. A common interface may display more than one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, and an authentication separately or concurrently for an asset. A display of said information may be ordered and listed in any order depending on user preference and input.

[00444] Another embodiment may include a method wherein the common interface is configured to display one or more possible attack vectors reaching the each listed asset. Generating this common interface display may include generating a map of an asset to include recording information such as a region identifier, site identifier, datacenter identifier, physical address, network address, workload name, or any other identifier which may be acquired via an API provided through a cloud service provider's system and demonstrating via a display the myriad attack vectors currently available to an analyzed asset. Display of the one or more possible attack vectors reaching the each listed asset may be displayed as written text, representative graphical figures, or similar metho of displaying the one or more possible attack vectors.

[00445] Another embodiment may include a method wherein the common interface is configured to display a recommended mitigation tactic for the each listed asset. A recommended mitigation tactic may be generated based on historical data of the cybersecurity risk, the vulnerability attributed to the listed asset, or the data contained on the each listed asset. Mitigation tactics may be selected based on user input or may be implemented automatically based on the mitigation tactics likelihood of a successful mitigation of the cybersecurity risk. A mitigation tactic may include increased frequency of scanning, heightened access control measures, firewalls and antivirus software patches, increased patch management scheduling, continuous workload monitoring, or similar tactics.

[00446] Another embodiment may include a method wherein the common interface is configured to display one or more workload metrics associated with the each listed asset. Workload metrics may be a statistical representation of the current performance of the workload, the capacity for additional processing of the workload, and the historical performance of the workload based on statistical trend.

[00447] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing a display to present a plurality of asset categories 301. A plurality may include at least one asset category to be displayed. Asset categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category.

[00448] Another embodiment may include 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 for matching keys with compute resources, the operations comprising receiving, via an input device, a selection of a particular asset category (step 1103). An input device may include a personal computer, tablet, smartphone, or virtual machine. A processor may include one or more integrated circuits (IC), including application-specific integrated circuit (ASIC), microchips, microcontrollers, microprocessors, all or part of a central processing unit (CPU), graphics processing unit (GPU), digital signal processor (DSP), field-programmable gate array (FPGA), server, virtual server, or other circuits suitable for executing instructions or performing logic operations. A selection of one or more asset categories may be received and updated as new visualizations are developed based on the current threats to an asset in that category. Asset categories may be selected or deselected by an end user. Once a particular asset category is selected, it may be added to a list of selected asset categories with a list for deselected categories as well. The list of selected asset categories may organized per category or may be compiled to include all current threats in a comprehensive selected asset threat list.

[00449] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing the display to present a list of assets in the selected category that have cybersecurity risks (step 1105). A display may be graphical, sequential, or multi-dimensional. A list of selected categories may include at least one of an account category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category. When an asset in a selected category has a cyber security risk associated with it, the asset may appear in a list of all listed categories having a cyber security risk associated with it, and may be organized by individual category, type of cyber security risk, or

some combination or sequential ordering methods based on the information available to the processor and its end user.

[00450] Another embodiment may include 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 for matching keys with compute resources, the operations comprising retrieving workload component cybersecurity risk information for each listed asset (step 1107A). When a category of assets is selected, a processor may be configured to order those assets according to available cybersecurity risk information available to said processor, and may organize by type of risk, type of asset, or similar type of sequentialization. Cybersecurity risk information may be determined from historical threats to the workload component, the current active cybersecurity threats to workload components of certain types, or some combination of the same.

[00451] Another embodiment may include 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 for matching keys with compute resources, the operations comprising retrieving cloud component cybersecurity risk information for each listed asset (step 1107B). Cloud component cybersecurity risk information may include identification of the risk based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset. Cybersecurity risk information to a cloud component may be determined from historical threats within the cloud environment 106, current active cybersecurity threats to similar workload components in the cloud environment 106 or the level of risk to assets within the cloud environment 106, as outlined above.

[00452] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing the display to present a common interface providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information for each listed asset (step 1109). The common interface may be configured to access component cybersecurity risk information based on the nature of the risk, the likelihood of the risk, or the severity of the risk to the component. On the same or a similar and related interface, a display may present cloud component cybersecurity risk information based on a semi-unique identifier, the nature of the risk, the likelihood of the risk, or the severity of the risk to each listed asset.

[00453] Another embodiment may include 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 for matching keys with compute resources, the operations comprising causing the display to present, in the common interface, an

interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information (step 1111). The interconnection between workload component cybersecurity risk information (step 1107A) and cloud component cybersecurity risk information (step 1107B) may generate an interface between them (step 1109) and may be presented as an interconnection between them (step 1111). The common interface presented may include direct interconnections between the component cybersecurity risk information and the cloud component cybersecurity risk information or links from one set of cybersecurity risk information. For instance, the component cybersecurity risk information may include a link to access the cloud component cybersecurity risk information. Conversely, the cloud component cybersecurity risk information may include a link to access the component cybersecurity risk information.

[00454] In some embodiments, a processor may cause a display of one or more risks aggregated in the same visualization. **Fig. 12** is a schematic block diagram illustrating an exemplary embodiment of a visual representation of displaying aggregated cybersecurity risk information, consistent with disclosed embodiments. A visualization may be generated from one or more sources based on scanning system 101. For example, a visual display may include a cloud provider visualization 1201, displaying where the multiple sources of data may be derived from. A cloud service provider may include services such as Cloudflare, Amazon Web Services, Google Cloud, IBM Cloud, Oracle Cloud, Microsoft Azure, or similar. This is further displayed via a communicative connection through an access port 1203 to a data server 1205.

[00455] Data server 1205 may include servers such as 109A-D. Data server 1205 may then display risk information that may be communicatively passed through web server 1207, demonstrating one or more possible sources of the one or more cybersecurity risk information. Data server 1205 may be communicatively connected to a web server 1207 (through access port 1203, which may be the same or a different port from port 1203 between cloud service provider 1201 and data server 1205), which can then be communicatively connected to one or more users via user access points 1209. This may include, among other things, user access that may be verified or validated via a credential 1211. Each of these connections may demonstrate one or more risk paths that may be exploited in a single view to allow for prioritization of threats and analysis of possible users who may exploit those risks and vulnerabilities, as discussed above. This may allow an administrator to prioritize risks to a given storage volume as well as assess lateral storage volumes that may be similarly risked based on the assessed threats in a single image.

[00456] **Fig. 13** is a schematic block diagram illustrating an exemplary embodiment of a visual representation of a flow path for aggregated risk information, consistent with disclosed embodiments. In another embodiment, a processor may be configured to display a flow path of aggregated risk information. In one view, a displayed visualization may include user access

1301, which displays risk information such as service vulnerability 1303 and an insecure private key 1305. These vulnerabilities or risks may be displayed to an administrator via developer server 1307 to an administrator through administrator access 1309. As previously discussed, this will allow an administrator to visualize myriad risks to a storage volume in a single view that will allow an administrator to prioritize risks and analyze possible exploiters of this risk information on lateral systems and storage volumes.

[00457] In another embodiment, risk information may be shown in one or more views. This may involve a risk path similar to the exemplary embodiment of Fig. 13. In an exemplary embodiment, in a single visualization data from multiple sources may be displayed. This may involve identifying risk information such as the origin of the risk and potential exploiters of a risk. In some embodiments, this visualization may be done for a specific risk. In another embodiment, this visualization may be done for a combination of risks.

[00458] Disclosed embodiments may include any one of the following bullet-pointed features alone or in combination with one or more other bullet-pointed features, whether implemented as a method, by at least one processor, and/or stored as executable instructions on non transitory computer readable media:

- establishing a trusted relationship between a source account in a cloud environment and a scanner account;
- using the established trust relationship, utilize at least one cloud provider API to identify workloads in the source account;
- using the at least one cloud provider API to query a geographical location of at least one of the identified workloads;
- receiving an identification of the geographic location;
- using the cloud provider APIs to access block storage volumes of the at least one workload;
- determining a file-system of the at least one workload;
- mounting the block storage volumes on a scanner based on the determined file-system;
- activating a scanner at the geographic location;
- reconstructing from the block storage volumes a state of the workload; and
- assessing the reconstructed state of the workload to extract insights.
- wherein the geographic location includes an identifier of a physical site.
- wherein mounting includes selecting a driver corresponding to the determined file system.
- wherein the at least one processor is further configured to deploy a scanner at the geographical location.

- wherein the identification of the geographic location comprises an identification of a data center, at least one of a data center name, Internet Protocol (IP) address, name of the cloud provider, or a unique identity.
- wherein the reconstructed state of the workload includes at least two of an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, a profile configuration, a log, or a database content;
- wherein the at least one processor is further configured to update the reconstructed state of the workload based on at least one change to the block storage volumes.
- wherein the insights comprise at least one of a vulnerability associated with the workload or a composition of installed applications associated with the workload.
- wherein to mount the block storage volumes on the scanner, the at least one processor is configured to create a snapshot of the block storage volumes;
- wherein to mount the block storage volumes on the scanner, the at least one processor is configured to mount the snapshot of the block storage volumes on the scanner
- wherein the at least one processor is further configured to encrypt the snapshot of the block storage volumes;
- wherein the at least one processor is further configured to mount the encrypted snapshot of the block storage volumes on the scanner;
- using a cloud provider API, access a block storage volume of a workload maintained in a cloud storage environment;
- identifying an installed software application in the accessed block storage volume;
- analyzing the identified installed software application to determine an associated software version;
- accessing a data structure of known software vulnerabilities for a plurality of versions of software applications;
- performing a lookup of the identified installed software version in the data structure to identify known vulnerabilities; and
- performing at least one of query the cloud provider API to determine network accessibility information related to the workload, identify at least one port on which the vulnerable application is accessible, use network accessibility information and at least one port to identify one or more vulnerabilities susceptible to attack from outside the workload.
- implementing a remedial action in response to the identified one or more vulnerabilities.
- wherein the remedial measure includes transmitting an alert to a device associated with an administrator.

- wherein querying the cloud provider API to determine network accessibility information related to the workload further comprises examining data sources associated with the workload
- wherein querying the cloud provider API to determine network accessibility information related to the workload further comprises determining the network accessibility information based on the examined data sources.
- wherein querying the cloud provider API to determine network accessibility information related to the workload further comprises: wherein the network accessibility information includes at least one of: data from an external data source, cloud provider information, or at least one network capture log.
- identifying the installed software application comprises extracting data from at least one of operating system packages, libraries, or program language libraries;
- identifying the installed software application comprises identifying the installed software application based on the extracted data;
- identifying a version of the installed software application;
- wherein the identified installed software application includes one or more scripts;
- wherein the data structure includes aggregated vulnerability data;
- wherein the aggregated vulnerability data includes data from one or more third-party vendors;
- wherein the aggregated vulnerability data includes data collected by a scanner;
- wherein the aggregated vulnerability data includes at least one of an advisory, an exploit, a security announcement, or a known bug.
- querying the cloud provider API to determine network accessibility information related to the workload further comprises: wherein the network accessibility information includes at least one of: data from an external data source, cloud provider information, or at least one network capture log.
- 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;
- querying the at least one cloud provider API to determine at least one networking configuration of the entities;
- building a graph connecting the plurality of entities based on the networking configuration;
- 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;

- traversing the graph to identify at least one source originating via the internet and reaching the workload; and
- outputting a risk notification associated with the workload.
- wherein the plurality of entities includes a virtual network appliance.
- wherein the virtual network appliance is at least one of a load balancer, a firewall, a proxy, or a router.
- 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.
- wherein the graph includes a data structure sequentially connecting entities.
- wherein the graph includes directional vectors indicating directions of dataflow.
- wherein building the graph comprises identifying individual entities as nodes
- wherein building the graph comprises connecting the nodes.
- wherein the graph includes port numbers associated with the workload.
- wherein the graph includes a path from the at least one source to the workload.
- utilizing a cloud provider API to access a block storage volume of a workload maintained on a target account in a target system of a cloud storage environment;
- utilizing a scanner at a location of the block storage volume and on a secondary system other than the target system;
- scanning the block storage volume for malicious code, using the secondary system;
- identifying malicious code based on the scan; and
- outputting from the secondary system, a notification of a presence of malicious code in the target system.
- wherein the location of the block storage volume includes at least one of: the target account, a secondary system account, a cloud provider account, or a third party account.
- wherein scanning the block storage volume includes scanning disk-backed memory.
- wherein the disk-backed memory includes at least one of a page file or a cache file.
- wherein the secondary system includes at least one of a virtual machine, a container, or a serverless function.
- wherein the secondary system has an operating system different from an operating system of the target account.
- wherein the malicious code includes a rootkit.
- wherein utilizing a scanner includes suspending an operation of the scanner after the scan of the block storage volume.

- wherein utilizing a scanner includes modifying a pre-utilized scanner at the location of the block storage volume based on information related to the target account to obtain a modified scanner;
- wherein utilizing a scanner includes utilizing the modified scanner.
- identifying assets in a cloud environment;
- identifying risks associated with each of the identified asset;
- identifying relationships between at least some of the identified assets, the relationships including at least one of a trust, a network connectivity, or a mechanism of network proxying;
- receiving an identification of a specific asset under investigation;
- performing a forward analysis of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment;
- performing a backward analysis of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset; and
- outputting a signal to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points.
- wherein the network outside the cloud environment includes the Internet.
- wherein the assets in the cloud environment include at least one of: a virtual machine, a network appliance, a storage appliance, a compute instances, or an engine instance.
- wherein identifying the assets in a cloud environment includes identifying the assets based on at least one of: an identity and access management policy, an organization policy, or an access policy.
- wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream asset or a downstream asset.
- wherein the visualization includes a presentation of the alternative paths.
- wherein the presentation of the forward and backward paths indicates port numbers for each pathway.
- wherein the visualization of the entry points indicates at least one entry point at risk.
- monitoring network activities of the assets in a cloud environment.
- detecting detect a potential risk associated with the specific asset based on the monitored network activities.

- detecting a potential risk associated with the specific asset based on a network activity of the specific asset.
- detecting a potential risk associated with the specific asset based on a network activity of an upstream asset of the specific asset.
- detecting a potential risk associated with the specific asset based on a network activity of a downstream asset of the specific asset.
- analyzing a cloud environment to identify a plurality of keys to the compute resources in the cloud environment;
- performing a cryptographic analysis on the plurality of keys to identify a first set of fingerprints that uniquely identify each of the plurality of keys, the first set of fingerprints being non-functional;
- analyzing trust configurations of the compute resources to identify a second set of fingerprints for each of the compute resources; and
- comparing the first set of fingerprints with the second set of fingerprints to match keys with the compute resources without using the keys to access the compute resources.
- wherein the plurality of keys are stored in at least one workload.
- wherein at least one of the first set of fingerprints is not identical to any key of the plurality of keys.
- wherein at least one of the plurality of keys includes at least one of a password, a script containing a password, a private component of a private-public key pair, a cloud key, or an Secure Shell (SSH) key.
- testing validity of at least one of the plurality of keys.
- analyzing a multi-machine interaction in the cloud environment using the first set of fingerprints.
- analyzing the multi-machine interaction includes comparing the first set of fingerprints with the second set of fingerprints.
- analyzing a multi-machine interaction in the cloud environment using the plurality of keys.
- accessing a primary account maintained in a cloud environment;
- receiving information defining a structure of the primary account, wherein the structure includes a plurality of assets, and wherein the information excludes raw data of the primary account;
- deploying, inside the primary account or inside a secondary account for which trust has been established with the primary account, at least one ephemeral scanner configured to scan at least one block storage volume and to output metadata defining the at least one block storage volume, the output excluding raw data of the primary account;

- receiving a transmission of the metadata from the at least one ephemeral scanner, wherein the transmission excludes raw data of the primary account;
- analyzing the received metadata to identify a plurality of cybersecurity vulnerabilities;
- correlating each of the identified plurality of cybersecurity vulnerabilities with one of the plurality of assets; and
- generating a report correlating the plurality of cybersecurity vulnerabilities with the plurality of assets.
- wherein defining the at least one block storage volume includes presenting risk data without sharing consumer data or data that was used to identify the risk data.
- wherein the metadata defining the at least one block storage volume includes at least one of: an indication of an installed application, a version of an installed application, an operating system configuration, an application configuration, or a profile configuration.
- receiving a transmission of updated metadata defining the at least one block storage volume in response to at least one change to the at least one block storage volumes.
- wherein the ephemeral scanner is configured to perform vulnerability scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform configuration scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform malware scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform lateral-movement risk analysis of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform sensitive information scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform container scanning of the at least one block storage volume.
- wherein the ephemeral scanner is configured to perform keys and password scanning of the at least one block storage volume.
- causing a display to present a plurality of asset categories;
- receiving, via an input device, a selection of a particular asset category;
- causing the display to present a list of assets in the selected category that have cyber security risks;
- for each listed asset, retrieving workload component cybersecurity risk information, and
- for each listed asset. retrieving cloud component cybersecurity risk information;

- for each listed asset, causing the display to present a common interface providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information; and
- for each listed asset, causing the display to present, in the common interface, an interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information.
- wherein the plurality of asset categories include at least one of an account category, an authentication category, a container category, a database category, an image category, a container category, a managed service category, a messaging service category, a monitoring category, a network category, a storage category, a user category, a access category, a virtual machine category, or a serverless category.
- wherein the common interface is configured to display information relating to at least one of an asset type, a risk, a region, or an account.
- wherein the common interface is configured to display description for each listed asset.
- wherein the common interface is configured to display at least one of a vulnerability, an insecure configuration, an indication of a presence of malware, a neglected asset, a data at risk, a lateral movement, or an authentication.
- wherein the common interface is configured to display one or more possible attack vectors reaching the each listed asset.
- wherein the common interface is configured to display a recommended mitigation tactic for the each listed asset.
- wherein the common interface is configured to display one or more workload metrics associated with the each listed asset.
- wherein the at least one processor is further configured to provide a cybersecurity report for the each listed asset.
- wherein the at least one processor is configured to identify a risk level distribution among the listed assets.
- receiving a search query for a specific risk; or
- identifying one or more assets vulnerable to the specific risk.

[00459] Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the disclosed embodiments disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the disclosed embodiments being indicated by the following claims.

[00460] Computer programs based on the written description and methods of this specification are within the skill of a software developer. The various programs or program modules can be created using a variety of programming techniques. One or more of such

software sections or modules can be integrated into a computer system, non-transitory computer readable media, or existing software.

[00461] Moreover, while illustrative embodiments have been described herein, the scope includes any and all embodiments having equivalent elements, modifications, omissions, combinations (e.g., of aspects across various embodiments), adaptations or alterations based on the present disclosure. The elements in the claims are to be interpreted broadly based on the language employed in the claims and not limited to examples described in the present specification or during the prosecution of the application. These examples are to be construed as non-exclusive. Further, the steps of the disclosed methods can be modified in any manner, including by reordering steps or inserting or deleting steps. It is intended, therefore, that the specification and examples be considered as exemplary only, with a true scope and spirit being indicated by the following claims and their full scope of equivalents.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT *(Use as many sheets as necessary)*				***Complete if Known***	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	2431
				Examiner Name	2397
Sheet	1	of	1	*Attorney Docket Number*	**15635.0008-00000**

U.S. PATENTS					
Examiner Initials*	Cite No.	Document Number Number-Kind Code *(if known)*	Issue Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear

U.S. PUBLISHED PATENT APPLICATIONS					
Examiner Initials*	Cite No.	Document Number Number-Kind Code *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-2019/0116199-A1	04-18-2019	Stopel et al.	

Note: Submission of copies of U.S. Patents and published U.S. Patent Applications is not required.

FOREIGN PATENT DOCUMENTS						
Examiner Initials*	Cite No.	Foreign Patent Document Country Code Number Kind Code *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	Translation

NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	Translation
		Advisory Action No. 17/657,972 dated October 31, 2022, in the United States Patent and Trademark Office. (15635.0005-00000)	N/A

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

UNITED STATES
PATENT AND TRADEMARK OFFICE

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

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/657,981	11/09/2022 04:08:03 PM ET	15635.0008-00000

Title of Invention

Forward and Rearward Facing Attack Vector Visualization

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2397	FILED BY	Michelle Daniels
PATENT CENTER #	61161882	AUTHORIZED BY	Christopher Johns
CUSTOMER #	22852	FILING DATE	04/05/2022
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Avi SHUA

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1007	E2022A9G08384642	Michelle Daniels

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1806	SUBMISSION OF AN INFORMATION DISCLOSURE STATEMENT	260.00	1	260.00
			TOTAL AMOUNT:	$260.00

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

New Applications Under 35 U.S.C. 111

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

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

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

New International Application Filed with the USPTO as a Receiving Office

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

PATENT
Attorney Docket No. 15635.0008-00000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Avi SHUA	)	Group Art Unit: 2431
	)	
Application No.: 17/657,981	)	Examiner: SU, SARAH
	)	
Filed: April 5, 2022	)	Confirmation No.: 2397
	)	
For: FORWARD AND REARWARD FACING ATTACK VECTOR VISUALIZATION	)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. § 1.97(d)

Pursuant to 37 C.F.R. §§ 1.56 and 1.97(d), Applicant brings to the attention of the Examiner the documents on the attached listing. This Information Disclosure Statement is being filed after a Ex Parte Quayle Office Action and is accompanied by a fee of $260.00 as specified under § 1.17(p) and a statement as specified under § 1.97(e).

Statement under 37 C.F.R. § 1.97(e)

No item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in this Information Disclosure

Statement was known to any individual designated in § 1.56(c) more than three months prior to the filing of this Information Disclosure Statement.

A copy of the non-patent literature document is attached.

Applicant respectfully requests that the Examiner consider the listed documents on the attached Form PTO/SB/08 and indicate that they were considered by making appropriate notations on the attached form.

This submission does not represent that a search has been made or that no better art exists and does not constitute an admission that each or all of the listed documents are material or constitute "prior art." If the Examiner applies any of the documents as prior art against any claim in the application and Applicant determines that the cited documents do not constitute "prior art" under United States law, Applicant reserves the right to present to the U.S. Patent and Trademark Office the relevant facts and law regarding the appropriate status of such documents.

Applicant further reserves the right to take appropriate action to establish the patentability of the disclosed invention over the listed documents, should one or more of the documents be applied against the claims of the present application.

If there is any fee due in connection with the filing of this Statement, please charge the fee to deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: November 9, 2022

By: /Christopher C. Johns/
Christopher C. Johns
Reg. No. 68,664
(202) 408-4000

UNITED STATES
PATENT AND TRADEMARK OFFICE

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

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/657,981	**11/09/2022 04:08:03 PM ET**	**15635.0008-00000**

Title of Invention

Forward and Rearward Facing Attack Vector Visualization

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2397	FILED BY	Michelle Daniels
PATENT CENTER #	61161882	FILING DATE	04/05/2022
CUSTOMER #	22852	FIRST NAMED INVENTOR	Avi SHUA
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Christopher Johns

Documents

TOTAL DOCUMENTS: 3

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
15635.0008-IDS_After_Quayle_OA.pdf	3	Transmittal Letter	18 KB
sb08.pdf	1	Information Disclosure Statement (IDS) Form (SB08)	31 KB
NPL-Advisory Action -10-31-2022-15635.0005-00000.pdf	4	Non Patent Literature	236 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)

15635.0008-IDS_After_Quayle_OA.pdf	20A0C60CC2E8944FB999832110584EC3C8C1B4D035B99D21FD074554FDF7858F40E1017C1A8AF35F240733E9380E061DE8D8C4F8E4F24AFABC0593CEEEF89FE9
sb08.pdf	ED2DA21D11294E15CE81DE23ABDAED1F4BF6F8EEFF141A14B76D1FB7CD7158B37DCEBC487F3FC8EA20DF6366546EA36804F00884106D392ABB407F74A02142BE
NPL-Advisory Action -10-31-2022-15635.0005-00000.pdf	9260CE86990C66B1858BCDCE8BEC93B75BA2EC42A4318C217ED44953E1DD625CACF56AE76C34CDC870BDE73BBC8FD9939F9698ED542B5D8401760FC8D29BF726

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

New Applications Under 35 U.S.C. 111

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

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

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

New International Application Filed with the USPTO as a Receiving Office

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

UNITED STATES PATENT AND TRADEMARK OFFICE
COMMISSIONER FOR PATENTS
P.O.BOX 1450
ALEXANDRIA VA 22313-1451

PRESORTED
FIRST-CLASS MAIL
U.S. POSTAGE PAID
POSTEDIGITAL
NNNNN

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

Courtesy Reminder for
Application Serial No: 17/657,981

Attorney Docket No: 15635.0008-00000
Customer Number: 22852
Date of Electronic Notification: 10/31/2022

This is a courtesy reminder that new correspondence is available for this application. If you have not done so already, please review the correspondence. The official date of notification of the outgoing correspondence will be indicated on the form PTOL-90 accompanying the correspondence.

An email notification regarding the correspondence was sent to the following email address(es) associated with your customer number:
regional-desk@finnegan.com

To view your correspondence online or update your email addresses, please visit us anytime at **https://ppair-my.uspto.gov/pair/PrivatePair.**
If you have any questions, please email the Electronic Business Center (EBC) at EBC@uspto.gov or call 1-866-217-9197.

UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
17/657,981	04/05/2022	Avi SHUA	15635.0008-00000	2397

22852 7590 10/31/2022

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER
SU, SARAH

ART UNIT	PAPER NUMBER
2431	

NOTIFICATION DATE	DELIVERY MODE
10/31/2022	ELECTRONIC

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

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

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

regional-desk@finnegan.com

Office Action Summary	Application No. 17/657,981	Applicant(s) SHUA, Avi	
	Examiner SARAH SU	Art Unit 2431	AIA (FITF) Status Yes

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

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 2 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☑ Responsive to communication(s) filed on 5 April 2022.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL.** 2b) ☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☑ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☑ Claim(s) 81-100 is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6) ☑ Claim(s) 81-100 is/are allowed.

7) ☐ Claim(s) _____ is/are rejected.

8) ☐ Claim(s) _____ is/are objected to.

9) ☐ Claim(s) _____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov.**

Application Papers

10)☑ The specification is objected to by the Examiner.

11)☑ The drawing(s) filed on 4/5/22 is/are: a)☐ accepted or b)☑ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☑ Notice of References Cited (PTO-892)

2) ☑ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date _____.

4) ☐ Other: _____.

DETAILED ACTION

Notice of Pre-AIA or AIA Status

1. The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

2. Amendment A, received on 5 April 2022, has been entered into record. In this amendment, claims 1-80 have been canceled, and claims 81-100 have been added.

3. Claims 81-100 are presented for examination.

Priority

4. The claim for priority from PCT/IB2022/052670 filed on 23 March 2022, which claims priority from US Provisional 63/180,048, filed on 26 April 2021, is duly noted.

5. This application is in condition for allowance except for the following formal matters:

Specification

6. The disclosure is objected to because of the following informalities:

a. In [00114], line 3: "unique identity.." should read –unique identity.–

Appropriate correction is required.

7. The disclosure is objected to because it contains an embedded hyperlink and/or other form of browser-executable code. Applicant is required to delete the embedded hyperlink and/or other form of browser-executable code; references to websites should be limited to the top-level domain name

without any prefix such as http:// or other browser-executable code. See MPEP § 608.01. See [008], line 4.

Drawings

8. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: 411b, 411c (Figure 4); 1305 (Figure 13).

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Prosecution on the merits is closed in accordance with the practice under *Ex parte Quayle*, 25 USPQ 74, 453 O.G. 213, (Comm'r Pat. 1935).

A shortened statutory period for reply to this action is set to expire **TWO (2) MONTHS** from the mailing date of this letter. Extensions of time may be granted under 37 CFR 1.136 but in no case can any extension carry the date for reply to this Office action beyond the maximum period of SIX MONTHS set by statute (35 U.S.C. 133).

Allowable Subject Matter

9. Claims 81-100 are allowed.

10. The following is an examiner's statement of reasons for allowance:

Although the prior art of record (such as Wang et al. (CN 110708315 A1)) teaches *at least one processor configured to: identify assets in a cloud environment* (Abstract, lines 2-3)*; identify risks associated with each of the identified asset* (Abstract, lines 3-5)*; identify relationships between at least some of the identified assets, the relationships including at least one of a trust, a network connectivity, or a mechanism of network proxying* (page 2, lines 17-25)*; receive an identification of a specific asset under investigation* (page 3, lines 16-19), none of the prior art of record alone or in combination teaches *perform a forward analysis of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment; perform a backward analysis of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset; and output a signal to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points.*

The closest prior art made of record are:

a. Wang discloses a system and method for identification of asset vulnerability.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Prior Art Made of Record

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

a. Perlman (US Patent 8,006,285 B1) discloses a system and method for dynamic defense of network attacks.

b. Sekar et al. (US 2020/0059481 A1) discloses a system and method for expedient detection and reconstruction of cyber events in a compact scenario representation using provenance tags and customizable policy.

c. Shen (WO 2021/072877 A1) discloses a system and method for secure starting for cloud host.

d. Shua (US 2020/0244692 A1) discloses a system and method for techniques for securing virtual cloud assets at rest against cyber threats.

e. Zimmermann et al. (US 2020/0137097 A1) discloses a system and method for securing an enterprise computing environment.

Conclusion

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to SARAH SU whose telephone number is (571)270-3835. The examiner can normally be reached 7:30 AM - 4:00 PM.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lynn Feild can be reached on 571-272-2092. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: https://patentcenter.uspto.gov. Visit https://www.uspto.gov/patents/apply/patent-center for more information about Patent Center and https://www.uspto.gov/patents/docx for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/SARAH SU/
Primary Examiner, Art Unit 2431

Notice of References Cited	Application/Control No. 17/657,981	Applicant(s)/Patent Under Reexamination SHUA, Avi	
	Examiner SARAH SU	Art Unit 2431	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20200244692-A1	07-2020	SHUA; Avi	H04L63/1425	1/1
*	B	US-20200137097-A1	04-2020	Zimmermann; Gil	G06F21/6218	1/1
*	C	US-20200059481-A1	02-2020	Sekar; Ramasubramanian	H04L63/1425	1/1
*	D	US-8006285-B1	08-2011	Perlman; Radia J.	H04L63/1458	709/224
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	CPC Classification
	N	WO-2021072877-A1	04-2021	WO	SHEN Y	G06F21/53
	O	CN-110708315-A	01-2020	CN	FAN Y	H04L63/1433
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/657,981	SHUA, Avi
	Examiner	Art Unit
	SARAH SU	2431

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS										
☐ Claims renumbered in the same order as presented by applicant						☐ CPA		☐ T.D.		☐ R.1.47
CLAIM		DATE								
Final	Original	10/25/2022								
	1	-								
	2	-								
	3	-								
	4	-								
	5	-								
	6	-								
	7	-								
	8	-								
	9	-								
	10	-								
	11	-								
	12	-								
	13	-								
	14	-								
	15	-								
	16	-								
	17	-								
	18	-								
	19	-								
	20	-								
	21	-								
	22	-								
	23	-								
	24	-								
	25	-								
	26	-								
	27	-								
	28	-								
	29	-								
	30	-								
	31	-								
	32	-								
	33	-								
	34	-								
	35	-								
	36	-								
	37	-								
	38	-								
	39	-								
	40	-								
	41	-								
	42	-								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/657,981	SHUA, Avi
	Examiner	**Art Unit**
	SARAH SU	2431

CLAIM		DATE								
Final	Original	10/25/2022								
	43	-								
	44	-								
	45	-								
	46	-								
	47	-								
	48	-								
	49	-								
	50	-								
	51	-								
	52	-								
	53	-								
	54	-								
	55	-								
	56	-								
	57	-								
	58	-								
	59	-								
	60	-								
	61	-								
	62	-								
	63	-								
	64	-								
	65	-								
	66	-								
	67	-								
	68	-								
	69	-								
	70	-								
	71	-								
	72	-								
	73	-								
	74	-								
	75	-								
	76	-								
	77	-								
	78	-								
	79	-								
	80	-								
	81	=								
	82	=								
	83	=								
	84	=								
	85	=								
	86	=								
	87	=								
	88	=								
	89	=								
	90	=								
	91	=								
	92	=								
	93	=								
	94	=								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/657,981	SHUA, Avi
	Examiner	**Art Unit**
	SARAH SU	2431

CLAIM		DATE								
Final	Original	10/25/2022								
	95	=								
	96	=								
	97	=								
	98	=								
	99	=								
	100	=								

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/657,981	SHUA, Avi
	Examiner	Art Unit
	SARAH SU	2431

CPC - Searched*		
Symbol	**Date**	**Examiner**
H04L 63/1433, 14$2	10/25/2022	SS
H04L 9/45558, 5072	10/25/2022	SS
G06F 2009/4557, 45595	10/25/2022	SS

CPC Combination Sets - Searched*		
Symbol	**Date**	**Examiner**

US Classification - Searched*			
Class	**Subclass**	**Date**	**Examiner**

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	**Date**	**Examiner**
Inventor Search	10/25/2022	SS
IP.com Search	10/25/2022	SS
PE2E Search (see attached)	10/25/2022	SS
H04L 63/1433, 14$2 (see search history notes to determine scope)	10/25/2022	SS
H04L 9/45558, 5072 (see search history notes to determine scope)	10/25/2022	SS
G06F 2009/4557, 45595 (see search history notes to determine scope)	10/25/2022	SS

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	17/657,981	SHUA, Avi
	Examiner	Art Unit
	SARAH SU	2431

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner

SECURE STARTING METHOD AND APPARATUS FOR CLOUD HOST, AND COMPUTER DEVICE AND STORAGE MEDIUM

DOCUMENT ID
WO-2021072877-A1

DATE PUBLISHED
2021-04-22

INVENTOR INFORMATION

NAME	CITY	STATE	ZIP CODE	COUNTRY
SHEN, YONG	N/A	N/A	518000	CN

DATE FILED
2019-11-14

Abstract

Disclosed are a secure starting method and apparatus for a cloud host, and a computer device and a storage medium. The method comprises: when a starting instruction, triggered by a user, for a specified cloud host is received, acquiring, according to a preset rule, asset information corresponding to the specified cloud host; determining whether there are vulnerabilities, corresponding to the asset information of the specified cloud host, stored in a preset vulnerability database; if so, acquiring the last shutdown time of the specified cloud host and release times respectively corresponding to the vulnerabilities; according to the last shutdown time and the release times, determining whether there is a specified vulnerability, of which the release time is within a specified time period, in all the vulnerabilities; and if so, starting the specified cloud host in a pre-created secure isolation area. According to the preset application, the specified cloud host can be effectively prevented from being invaded or attacked by vulnerabilities during use, and the security of the specified cloud host after starting is guaranteed.

Description

INVENTION-TITLE
This application claims the priority of a Chinese patent application filed with the Chinese Patent Office on October 15, 2019, the application number is 2019109790501, and the invention title is "the secure boot method, device, computer equipment and storage medium of the cloud host", and its entire content Incorporated in this application by reference.
TECHNICAL-FIELD
Technical field
This application relates to the technical field of cloud computing, and in particular to a secure startup method, device, computer equipment, and storage medium of a cloud host.
BACKGROUND-ART
Background technique
Cloud computing is currently an important field and direction in the development of IT technology, and it is gradually infiltrating all areas of society. Cloud host is an important part of cloud computing in infrastructure applications. It is located at the bottom of the cloud computing industry chain and its products are derived from cloud computing platforms. The platform integrates the three core elements of Internet applications: computing, storage, and network, and provides users with public Internet infrastructure services. Cloud

hosting is a kind of similar to VPS (Virtual Private Server, virtual private server) the virtualization technology of the host, VPS uses virtual software (such as VZ or VM) to virtualize multiple parts similar to independent hosts on a host, which can realize single-machine multi-user, and each part can be used. As a separate operating system, the management method is the same as that of the host. The cloud host virtualizes multiple parts similar to independent hosts on a group of host machine clusters. Each host machine in the cluster has an image of the cloud host machine, which greatly improves the security and stability of the virtual host machine. In addition, in order to facilitate users to access cloud computing anytime and anywhere, cloud hosts can usually be accessed directly from the public network (Internet), and even managed.

In the existing various cloud computing systems, each cloud host will face the security risk of the protection gap. Specifically, when the cloud host is in the shutdown state, since the cloud host will repair all currently known security vulnerabilities when it is shut down, it can be considered that the cloud host in the shutdown state is also in a safe state. However, when the cloud host is restarted and exposed to the public network, the external environment at this time may change compared to before the shutdown. If there are new security vulnerabilities related to the cloud host, the cloud host may be attacked by attackers on the public network at any time, and the security of the cloud host is greatly threatened.

BEST-MODE

The best mode of the present invention

It should be understood that the specific embodiments described here are only used to explain the present application, and are not used to limit the present application.

Referring to Fig. 1, a secure boot method of a cloud host according to an embodiment of the present application includes:

S1: Acquire asset information corresponding to the designated cloud host according to preset rules when receiving a startup instruction for a designated cloud host triggered by a user, where the asset information includes at least operating system information;

S2: Determine whether there are vulnerabilities corresponding to the asset information of the designated cloud host stored in the preset vulnerability database, where the number of the vulnerabilities is one or more;

S3: If yes, obtain the last shutdown time of the designated cloud host and the release time corresponding to each of the vulnerabilities;

S4: According to the last shutdown time and each of the release times, determine whether there are specified vulnerabilities whose release time is within a specified period of time among all the vulnerabilities;

S5: If yes, start the designated cloud host, and when the designated cloud host is started, place the designated cloud host in a preset security isolation area to suspend the network access function of the designated cloud host.

As described in the above steps S1 to S5, the execution subject of the embodiment of the present invention is a secure boot device for a cloud host, specifically a host cluster used to virtual out the cloud host, through which the host cluster can realize the For the secure startup of the host, the above-mentioned designated cloud host is any cloud host among all the cloud hosts virtualized by the host cluster. Specifically, it is first judged whether a user-triggered start instruction for the designated cloud host is received; when the start instruction is received, the asset information corresponding to the designated cloud host is obtained according to preset rules, wherein the asset information includes at least the operating system Information, for example, the operating system is windowsserver2012, and the RDP service version 11.0 is used to provide a remote desktop. In addition, the asset information may also include cloud host service information and key application information. In addition, the foregoing preset rules may include: obtaining asset information corresponding to the specified cloud host by querying the cloud platform corresponding to the specified cloud host, or by invoking the specified scanning tool. Then, according to the above asset information, it is determined whether there are vulnerabilities corresponding to the asset information of the above designated cloud host stored in the preset vulnerability database, wherein the number of the above vulnerabilities is one or more, and the above vulnerability database stores passing The crawler tool crawls the vulnerability information related to various operating systems corresponding to the cloud host and the vulnerability information related to the cloud host service from the target website in real time. If the vulnerability database stores the vulnerability corresponding to the asset information of the specified cloud host, the last shutdown time of the specified cloud host and the release time corresponding to each of the

aforementioned vulnerabilities are obtained. Among them, for example, the form of any vulnerability information crawled can be CVE20190004-rdp vulnerability, released on July 02, 2019. The affected system is all versions of the rdp service before version 12.0, and the release time of the corresponding vulnerability is 2019 /07/02. Then, based on the last shutdown time and each of the above release times, it is determined whether there are specified vulnerabilities in all the vulnerabilities whose release time is within the specified time period, where the above specified time period is included between the above shutdown time and the above current time period. If there are specified vulnerabilities whose release time is within the specified time period, the above-mentioned designated cloud host will be started in the pre-created security quarantine area, and the network access function of the designated cloud host will be disabled in the above-mentioned security quarantine area. In this embodiment, when a new vulnerability related to the designated cloud host occurs during the last shutdown period, the designated cloud host will be intelligently placed in the pre-created security isolation zone for startup, thereby starting on the designated cloud host Then temporarily disable the network access function of the designated cloud host, effectively avoiding the intrusion or attack of the designated cloud host by vulnerabilities during use, and ensuring the security of the designated cloud host after it is started. Furthermore, if there are no specified vulnerabilities whose release time is within the specified period of time among all vulnerabilities, indicating that no related vulnerabilities appeared during the last shutdown of the specified cloud host, the startup of the specified cloud host will be completed directly , And there is no need to place the designated cloud host in a security quarantine area. In order to effectively guarantee the user experience of the designated cloud host.

Further, in an embodiment of the present application, the above step S1 includes:

S100: Determine whether an asset management tool is installed in the designated cloud host, where the asset management tool is used to periodically synchronize asset information corresponding to the designated cloud host to the designated cloud platform;

S101: If an asset management tool is installed, search for the asset information from the designated cloud platform;

S102: If an asset management tool is not installed, call a designated scanning tool to perform scanning processing on the designated cloud host to obtain the asset information.

As described in the above steps S100 to S102, the asset information may specifically include the operating system information corresponding to the specified cloud host, as well as the cloud host service information and key application information in the specified cloud host. The cloud platform corresponding to the specified cloud host can be queried. Or by calling the specified scanning tool to obtain the asset information corresponding to the specified cloud host. Specifically, first determine whether an asset management tool is installed in the specified cloud host. The asset management tool has a synchronization function for asset information of the specified cloud host. When the asset management tool is installed in the specified cloud host, the specified cloud host The host will periodically synchronize the internal asset information to the corresponding designated cloud platform through the above asset management tool, so that the designated cloud platform will store the asset information of the designated cloud host. If it is determined that an asset management tool is installed in the designated cloud host, the above asset information can be found from the designated cloud platform corresponding to the above designated cloud host. If there is no asset management tool installed in the designated cloud host, the designated scanning tool can be called to scan the cloud platform to obtain the above asset information. The designated scanning tool can be nmap, and the nmap tool can be used to perform the scanning process on the cloud platform. Specify the cloud host to send the operating system detection command \$nmap-O[target IP], so that the designated cloud host will query its own asset information after receiving the system detection command, and then the asset information returned by the designated cloud host can be obtained. After obtaining the asset information of the designated cloud host, it can then intelligently identify whether there are vulnerabilities corresponding to the designated cloud host in the vulnerability database based on the asset information.

Further, in an embodiment of the present application, before the foregoing step S2, the method includes:

S200: Use crawler tools to grab the first vulnerability information related to various operating systems corresponding to the cloud host and the second vulnerability information related to the cloud host service from the target website in real time;

S201: Create a vulnerability database;

S202: Store the first vulnerability information and the second vulnerability information in the vulnerability

database.
As described in the above steps S200 to S202, before the judgment process of judging whether the vulnerability corresponding to the specified cloud host is stored in the preset vulnerability database according to the asset information, the process of generating the vulnerability database is also included. Specifically, first, a crawler tool is used to grab the first vulnerability information related to various operating systems corresponding to the cloud host and the second vulnerability information related to the cloud host service from the target website in real time. Wherein, the above-mentioned first vulnerability information may include vulnerability information corresponding to the first vulnerability, and first patch information corresponding to the first vulnerability. Similarly, the aforementioned second vulnerability information may include vulnerability information corresponding to the second vulnerability, and second patch information corresponding to the second vulnerability. In addition, crawler tools refer to web crawlers (also known as web spiders or web robots), which are programs or scripts that automatically crawl information on the World Wide Web in accordance with certain rules. The crawler tool includes but is not limited to the Python crawler tool. In addition, the aforementioned target website is an authoritative website at home and abroad for publishing vulnerabilities related to cloud hosting. Specifically, the host cluster uses a crawler tool to execute a crawler file to crawl vulnerability information that meets the data crawling conditions set by the crawler file. The crawler file includes, but is not limited to, two data crawling conditions of target URL and search keywords. The target URL is the URL of the target website corresponding to the first vulnerability information that needs to be crawled and the second vulnerability information in the crawler file, and the URL (short for Uniform Resource Locator, that is, uniform resource locator) is correct A concise representation of the location and access method of resources obtained from the Internet is the address of a standard resource on the Internet. Search keywords refer to keywords in the crawler file that are used to limit the common characteristics of the first vulnerability information and the second vulnerability information to be crawled by the crawler file. After obtaining the above-mentioned first vulnerability information and the second vulnerability information, create a vulnerability database, and then store the first vulnerability information and the second vulnerability information in the above-mentioned vulnerability database to form a subsequent use to determine whether there is a possible threat Specify the vulnerability database of the security vulnerabilities of the cloud host.
Further, in an embodiment of the present application, before the above step S4, the method includes:
S400: Obtain the current time;
S401: Determine the time period included between the last shutdown time and the current time as the designated time period.
As described in the above steps S400 to S401, before the step of judging whether there are specified vulnerabilities whose release time is within a specified period of time among all the foregoing vulnerabilities, the step of determining the specified period of time is also included. Specifically, the current time is first obtained, The above-mentioned current time has the same accuracy as the above-mentioned last shutdown time, but the accuracy is not specifically limited. For example, the above-mentioned accuracy can be accurate to year, month and day, for example, the current time can be 2019/07/11, or even It can be accurate to the year, month, day, and so on, so I won't go into too much detail here. After the current time and the aforementioned last shutdown time are obtained, the time period included between the last shutdown time and the aforementioned current time is determined as the aforementioned designated time period. For example, if the current time is 2019/07/11 and the last shutdown time is 2019/07/01, it can be determined that the above specified time period includes: 2019/07/01-2019/07/11. This embodiment determines the specified time period based on the current time and the last shutdown time of the specified cloud host, which is beneficial to subsequently accurately identifying whether there are specified vulnerabilities in the vulnerability database that threaten the security of the specified cloud host based on the specified time period. , And then intelligently select the opening method for the designated cloud host based on the recognition result.
Further, in an embodiment of the present application, after the above step S5, the method includes:
S500: Determine whether the specified patch corresponding to the specified vulnerability is stored in the vulnerability database;
S501: If yes, download the designated patch, and store the designated patch in a designated storage area in the designated cloud host through a designated transmission method;

S502: Send an installation instruction for installing the specified patch to the specified cloud host, so that the specified cloud host installs the specified patch according to the installation instruction, and returns a corresponding patch installation result, where the patch Installation results include successful or failed installation;
S503: Determine whether the specified patch is successfully installed according to the patch installation result returned by the specified cloud host;
S504: If yes, transfer the designated cloud host out of the security isolation area to restore the normal network access function of the designated cloud host.
As mentioned in the above steps S500 to S504, after the designated cloud host is placed in the security quarantine area, the designated cloud host will be removed from the security only after the designated cloud host has successfully installed the designated patch corresponding to the above-mentioned specified vulnerability. Remove from the quarantine area to ensure the safety of the designated cloud host. Specifically, it is first determined whether the vulnerability database stores the designated patch corresponding to the designated vulnerability. If the above specified patch is stored in the vulnerability database, the specified patch will be downloaded, and the specified patch will be stored in the specified storage area in the specified cloud host through the specified transmission method. Among them, the above-mentioned designated transmission method is the local area network transmission method or the Bluetooth transmission method. After the host cluster downloads the above-mentioned designated patch, it transmits the downloaded designated patch to the designated storage area in the designated cloud host by using the designated transmission method internally. , Since the transmission process does not need to use network resources, the designated cloud host will not be attacked by attackers on the public network when receiving the designated patch, which effectively guarantees the security of the designated cloud host in the process of storing the designated patch In addition, the installation process of the designated cloud host to install the designated patch does not require a network, which further increases the security of the designated cloud host. In addition, the above-mentioned designated storage area is not specifically limited, and can be set according to actual needs. For example, it can be a newly created area dedicated to storing the designated patch, or an area with a larger storage space in the designated cloud host. ,and many more. After finishing storing the specified patch in the above specified storage area, an installation instruction to install the specified patch will be sent to the specified cloud host, so that the specified cloud host can download and install the specified patch according to the above installation instruction, and return to the corresponding patch installation As a result, the patch installation result includes installation success or installation failure. Then, according to the patch installation result returned by the designated cloud host, it is further judged whether the above-mentioned designated patch is successfully installed. If the specified patch is successfully installed in the specified cloud host, the specified cloud host will be transferred out of the aforementioned security quarantine area to restore the normal network access function of the specified cloud host, so that the specified cloud host can perform normal operations after the network access function is restored. Work to ensure the user's sense of experience when using the designated cloud host. Further, if the installation of the specified patch fails, the specified cloud host will return the patch installation result that the installation failed. At this time, the specified cloud host will not be transferred out of the security quarantine area, but the reason for the failure of the specified patch installation will be found first; Then, according to the failure reason of the failure of the specified patch installation, control the specified cloud host to re-download and install the above specified patch, and after confirming that the specified patch is successfully installed in the specified cloud host, transfer the specified cloud host out of the above security The quarantine area enables the normal network access function of the designated cloud host to be restored only when the designated patch is successfully installed, effectively ensuring the security of the designated cloud host.
In an embodiment of the present application, after the above step S504, the method includes:
S5040: Acquire first usage data of a first host machine corresponding to the designated cloud host according to a first preset period, where the first usage data includes at least a CPU usage rate, a memory usage rate, and a network interface bandwidth usage rate
S5041: Analyze and process the first usage data, and determine whether the first usage data is greater than a preset standard threshold;
S5042: If yes, find out the second host machine whose second usage data is less than the standard

threshold from all the host machines;
S5043: Migrate the designated cloud host from the first host to the second host.
As described in the above steps S5040 to S5043, after the designated cloud host is transferred out of the security quarantine area, in the normal working process of the designated cloud host, if the first host corresponding to the designated cloud host is in a high load state , The host cluster also has the function of intelligent migration for the specified cloud host. Specifically, after the startup of the specified cloud host is completed, the first usage data of the first host corresponding to the specified cloud host is acquired according to the first preset period, where the first preset period is not specifically limited, for example Can be set to one hour. In addition, the aforementioned usage data includes CPU usage, memory usage, and network interface bandwidth usage. Then analyze and process the first usage data, and determine whether the above-mentioned first usage data is greater than a preset standard threshold. For different types of first usage data, the same number of standard thresholds corresponding to the type are set. . In addition, the specific value of the standard threshold is not specifically limited, and can be set by the host cluster, or can also be set by the user according to requirements. For example, the standard threshold corresponding to CPU usage can be set to 80%, the standard threshold corresponding to memory usage can be set to 75%, and the standard threshold corresponding to network interface bandwidth usage can be set to 85%. If the above-mentioned first usage data is greater than the preset standard threshold value, it indicates that the first host machine corresponding to the designated cloud host is currently in a high-load state, and the first host machine has fewer computable resources. At this time, the second host with the second usage data less than the above standard threshold will be found from all the hosts in the host cluster, and the designated cloud host will be migrated from the above first host to the above second host to achieve Intelligently migrate the designated cloud host from the first host with excessive load to the second host with lower load, so that the computing resources of the designated cloud host are optimized, thereby effectively improving the operating efficiency and response speed of the designated cloud host. It also guarantees the safety of the first host. Further, if the number of second host machines is more than one, the designated host machine with the lowest data value corresponding to the second usage data is selected from all the second host machines, and then the designated cloud host machine is selected from the above-mentioned first host machine. The host machine is migrated to the above specified host machine. By filtering out the designated host with the lowest data value corresponding to the second usage data from all the second hosts, not only the operating efficiency and response speed of the designated cloud host can be improved, but also the load balance for all the hosts can be achieved.
Further, in an embodiment of the present application, after the above step S504, the method includes:
S5044: Acquire the available resource space in the cloud hard disk corresponding to the designated cloud host according to the second preset period;
S5045: Analyze and process the available resource space, and determine whether the available resource space is less than a preset resource space threshold;
S5046: If yes, filter out the junk data in the cloud hard disk;
S5047: Send a garbage removal instruction to the designated cloud host to control the designated cloud host to clear the junk data according to the garbage removal instruction.
As described in the above steps S5044 to S5047, after the designated cloud host is transferred out of the above-mentioned security quarantine area, the available resource space in the designated cloud host in a normal working state will be monitored and processed to ensure that the designated cloud host can have Sufficient available resource space. Specifically, after the specified cloud host is started, the available resource space in the cloud hard disk corresponding to the specified cloud host is acquired according to a second preset period, where the second preset period is not specifically limited, for example, it can be set to One hour. Then it is judged whether the above-mentioned available resource space is less than the preset resource space threshold. The specific value of the above-mentioned resource space threshold is not specifically limited. It can be set by the host cluster, or can also be set by the user according to the needs. For example, the resource space The threshold can be set to 5g. If the available resource space is less than the preset resource space threshold, it indicates that the current available resource space of the designated cloud host is small, and the cloud hard disk corresponding to the designated cloud host is under high pressure. At this time, the junk data in the cloud hard disk will be filtered out. The junk data includes the remaining designated cloud host configuration files, useless temporary files, useless files that have been

deleted/uninstalled, and the hard disk and/or volume have not been formatted. Successfully leftover files, etc. After the junk data is filtered out, the garbage removal instruction is sent to the designated cloud host to control the designated cloud host to clean up the junk data according to the garbage removal instruction. Specifically, the management tool or the designated cloud host can be used to clean up the junk data. Application software to realize the automatic removal of the above-mentioned junk data. This embodiment automatically and intelligently removes junk data in the cloud hard disk corresponding to the specified cloud host when the current available resources of the specified cloud host are low, effectively reducing the pressure and burden of the specified cloud host, and releasing the specified cloud host. The redundant and useless resources on the cloud host can be used to process the junk data in the cloud hard disk corresponding to the designated cloud host in an orderly and timely manner, avoid unnecessary waste of resources, improve the utilization efficiency of the internal resources of the designated cloud host, and finally improve the designated cloud The operating efficiency and response speed of the host.

2, an embodiment of the present application also provides a secure boot device for a cloud host, including:

The first obtaining module 1 is configured to obtain asset information corresponding to the specified cloud host according to a preset rule when a user-triggered start instruction for a specified cloud host is received, wherein the asset information includes at least operating system information;

The first judgment module 2 is used for judging whether there are vulnerabilities corresponding to the asset information of the designated cloud host stored in the preset vulnerability database, wherein the number of the vulnerabilities is one or more;

The second obtaining module 3 is configured to, if yes, obtain the last shutdown time of the designated cloud host and the release time corresponding to each of the vulnerabilities;

The second judgment module 4 is used for judging whether there are specified vulnerabilities whose release time is within a specified period of time among all the vulnerabilities according to the last shutdown time and each of the release times;

The isolation module 5 is configured to, if yes, start the designated cloud host in a pre-created security isolation area, wherein the network access function of the specified cloud host will be disabled in the security isolation area.

In this embodiment, the realization process of the functions and functions of the first acquisition module, the first judgment module, the second acquisition module, the second judgment module, and the isolation module in the secure boot device of the cloud host is detailed in the above cloud host The implementation process corresponding to steps S1 to S5 in the safe boot method will not be repeated here.

Further, in an embodiment of the present application, the above-mentioned first acquisition module includes:

The first determining unit is configured to determine whether an asset management tool is installed in the designated cloud host, wherein the asset management tool is used to periodically synchronize asset information corresponding to the designated cloud host to the designated cloud platform;

The first searching unit is configured to search for the asset information from the designated cloud platform if an asset management tool is installed;

The first calling unit is configured to, if the asset management tool is not installed, call a designated scanning tool to perform scanning processing on the designated cloud host, so as to obtain the asset information.

In this embodiment, the implementation process of the functions and roles of the first determining unit, the first searching unit, and the first calling unit in the secure boot device of the cloud host is detailed in the corresponding steps S100 to S100 to the secure boot method of the cloud host. The implementation process of S102 will not be repeated here.

Further, in an embodiment of the present application, the secure boot device of the cloud host includes:

The crawling module is used to grab the first vulnerability information related to various operating systems corresponding to the cloud host and the second vulnerability information related to the cloud host service from the target website in real time through a crawler tool;

Create module to create a vulnerability database;

The first storage module is used to store the first vulnerability information and the second vulnerability information in the vulnerability database.

In this embodiment, the implementation process of the functions and roles of the grab module, creation module, and first storage module in the secure boot device of the cloud host is detailed in the implementation of corresponding steps S200 to S202 in the secure boot method of the cloud host. The

process will not be repeated here.
Further, in an embodiment of the present application, the secure boot device of the cloud host includes:
The third obtaining module is used to obtain the current time;
The first determining module is configured to determine the time period included between the last shutdown time and the current time as the designated time period.
In this embodiment, the implementation process of the functions and roles of the third acquiring module and the first determining module in the secure startup device of the cloud host is detailed in the implementation process corresponding to steps S400 to S401 in the secure startup method of the cloud host. I won't repeat them here.
Further, in an embodiment of the present application, the secure boot device of the cloud host includes:
The second determining module is used to determine whether the specified patch corresponding to the specified vulnerability is stored in the vulnerability database;
The second storage module is configured to download the specified patch if it is, and store the specified patch in a specified storage area in the specified cloud host through a specified transmission method;
The first sending module is configured to send an installation instruction for installing the specified patch to the specified cloud host, so that the specified cloud host installs the specified patch according to the installation instruction, and returns the corresponding patch installation result, Wherein, the patch installation result includes installation success or installation failure;
The third judgment module is configured to judge whether the specified patch is successfully installed according to the patch installation result returned by the specified cloud host;
The transfer module is used for transferring the designated cloud host out of the security isolation area, so as to restore the normal network access function of the designated cloud host.
In this embodiment, the realization process of the functions and roles of the second determination module, the second storage module, the first sending module, the third judgment module, and the transfer module in the secure boot device of the cloud host is detailed in the above cloud host The implementation process corresponding to steps S500 to S504 in the secure boot method will not be repeated here.
Further, in an embodiment of the present application, the secure boot device of the cloud host includes:
The fourth acquiring module is configured to acquire the first usage data of the first host machine corresponding to the designated cloud host according to the first preset period, where the first usage data includes at least a CPU usage rate, a memory usage rate, and Network interface bandwidth usage rate;
The fourth judgment module is configured to analyze and process the first usage data, and judge whether the first usage data is greater than a preset standard threshold;
The searching module is configured to, if yes, find out the second host machine whose second usage data is less than the standard threshold value from all the host machines;
The migration module is used to migrate the designated cloud host from the first host to the second host.
In this embodiment, the implementation process of the functions and roles of the fourth acquisition module, the fourth judgment module, the search module, and the migration module in the secure boot device of the cloud host is detailed in the corresponding step S5040 in the secure boot method of the cloud host. The implementation process to S5043 will not be repeated here.
Further, in an embodiment of the present application, the secure boot device of the cloud host includes:
A fifth acquiring module, configured to acquire the available resource space in the cloud hard disk corresponding to the designated cloud host according to a second preset period;
The fifth judgment module is configured to analyze and process the available resource space, and judge whether the available resource space is less than a preset resource space threshold;
The screening module is used to screen out junk data in the cloud hard disk if so;
The second sending module is configured to send a garbage removal instruction to the designated cloud host to control the designated cloud host to clear the junk data according to the garbage removal instruction.
In this embodiment, the implementation process of the functions and roles of the fifth acquiring module, the fifth determining module, the screening module, and the second sending module in the secure boot device of the cloud host is detailed in the corresponding method in the secure boot method of the cloud host. The implementation process of steps S5044 to S5047 will not be repeated here.
Referring to FIG. 3, an embodiment of the present application also provides a computer device. The computer device may be a server, and its internal structure may be as shown in FIG. 3. The computer

equipment includes a processor, a memory, a network interface, and a database connected through a system bus. Among them, the processor designed for the computer equipment is used to provide calculation and control capabilities. The memory of the computer device includes a volatile or non-volatile storage medium and internal memory. The volatile or non-volatile storage medium stores an operating system, computer readable instructions, and a database. The internal memory provides an environment for the operation of the operating system and computer-readable instructions in the volatile or non-volatile storage medium. The database of the computer equipment is used to store data such as asset information and vulnerabilities. The network interface of the computer device is used to communicate with an external terminal through a network connection. When the computer-readable instruction is executed by the processor, the method for secure startup of the cloud host shown in any of the above exemplary embodiments is realized. Those skilled in the art can understand that the structure shown in FIG. 3 is only a block diagram of a part of the structure related to the solution of the present application, and does not constitute a limitation on the devices and computer equipment to which the solution of the present application is applied.

An embodiment of the present application also provides a computer-readable storage medium. The computer-readable storage medium may be volatile or non-volatile, and computer-readable instructions are stored thereon. When executed by one or more processors, the steps in the foregoing embodiment of the secure boot method for a cloud host are realized when executed by one or more processors.

In summary, the secure startup method, device, computer equipment, and storage medium of the cloud host provided in the embodiments of the present application, upon receiving a user-triggered startup instruction for a specified cloud host, obtain the Asset information corresponding to the designated cloud host, where the asset information includes at least operating system information; determine whether a predetermined vulnerability database stores vulnerabilities corresponding to the asset information of the designated cloud host, wherein the number of vulnerabilities Is one or more; if so, obtain the last shutdown time of the designated cloud host and the release time corresponding to each of the vulnerabilities; according to the last shutdown time and each release time, it is determined that all the Whether there are specified vulnerabilities in the vulnerabilities whose release time is within a specified period of time; if so, start the designated cloud host in the pre-created security isolation zone, where the network access function of the specified cloud host is in the security isolation zone Will be disabled. This application will intelligently place the designated cloud host in the pre-created security isolation zone for startup when a new vulnerability related to it occurs during the last shutdown of the designated cloud host, so that after the designated cloud host is started Temporarily disable the network access function of the designated cloud host, effectively avoiding the intrusion or attack of the designated cloud host by vulnerabilities during use, and ensuring the security of the designated cloud host after it is started.

A person of ordinary skill in the art can understand that all or part of the processes in the methods of the foregoing embodiments can be implemented by computer-readable instructions to instruct relevant hardware. The computer-readable instructions can be stored in a non-volatile or In a volatile computer-readable storage medium, when the computer-readable instructions are executed, they may include the processes of the above-mentioned method embodiments. Wherein, any reference to memory, storage, database or other media provided in this application and used in the embodiments may include non-volatile and/or volatile memory. Non-volatile memory may include read-only memory (ROM), programmable ROM (PROM), electrically programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), or flash memory. Volatile memory may include random access memory (RAM) or external cache memory. As an illustration and not a limitation, RAM is available in many forms, such as static RAM (SRAM), dynamic RAM (DRAM), synchronous DRAM (SDRAM), dual-rate SDRAM (SSRSDRAM), enhanced SDRAM (ESDRAM), synchronous Link (Synchlink) DRAM (SLDRAM), memory bus (Rambus) direct RAM (RDRAM), direct memory bus dynamic RAM (DRDRAM), and memory bus dynamic RAM (RDRAM), etc.

The above are only the preferred embodiments of this application, and do not limit the scope of this application. Any equivalent structure or equivalent process transformation made using the content of the specification and drawings of this application, or directly or indirectly applied to other related The technical field is equally included in the scope of patent protection of this application.

Claims

A method for securely starting a cloud host, which is characterized in that it includes:
When receiving a user-triggered start instruction for a designated cloud host, acquiring asset information corresponding to the designated cloud host according to a preset rule, where the asset information includes at least operating system information;
Judging whether there are vulnerabilities corresponding to the asset information of the designated cloud host stored in the preset vulnerability database, wherein the number of the vulnerabilities is one or more;
If yes, obtain the last shutdown time of the designated cloud host and the release time corresponding to each of the vulnerabilities;
According to the last shutdown time and each of the release times, determine whether there is a specified vulnerability whose release time is within a specified period of time among all the vulnerabilities;
If yes, start the designated cloud host in a pre-created security isolation zone, wherein the network access function of the designated cloud host will be disabled in the security isolation zone.
The secure startup method of a cloud host according to claim 1, wherein the step of obtaining asset information corresponding to the designated cloud host according to a preset rule comprises:
Determining whether an asset management tool is installed in the designated cloud host, wherein the asset management tool is used to periodically synchronize asset information corresponding to the designated cloud host to the designated cloud platform;
If an asset management tool is installed, search for the asset information from the designated cloud platform;
If the asset management tool is not installed, call the designated scanning tool to scan the designated cloud host to obtain the asset information.
The method for secure startup of a cloud host according to claim 1, wherein before the step of determining whether a vulnerability corresponding to the asset information of the designated cloud host is stored in the preset vulnerability database, the method further include:
Through the crawler tool, the first vulnerability information related to various operating systems corresponding to the cloud host and the second vulnerability information related to the cloud host service are captured from the target website in real time;
Create a vulnerability database;
The first vulnerability information and the second vulnerability information are stored in the vulnerability database.
The method for secure startup of a cloud host according to claim 1, wherein said determining whether there is a release time in all said vulnerabilities is within a specified time period according to said last shutdown time and each said release time Before the step of specifying vulnerabilities, the method also includes:
Get the current time;
The time period included between the last shutdown time and the current time is determined as the designated time period.
The method for safely starting a cloud host according to claim 1, wherein after the step of starting the designated cloud host in a pre-created security isolation zone, the method further comprises:
Determining whether the specified patch corresponding to the specified vulnerability is stored in the vulnerability database;
If yes, download the designated patch, and store the designated patch in a designated storage area in the designated cloud host through a designated transmission method;
Send an installation instruction for installing the specified patch to the specified cloud host, so that the specified cloud host installs the specified patch according to the installation instruction, and returns a corresponding patch installation result, wherein the patch installation result Including successful or failed installation;
Judging whether the specified patch is successfully installed according to the patch installation result returned by the specified cloud host;
If yes, transfer the designated cloud host out of the security isolation area to restore the normal network access function of the designated cloud host.
The secure startup method of a cloud host according to claim 5, wherein after the step of transferring the designated cloud host out of the security isolation area to restore the normal network access function of the

designated cloud host, The method also includes:
Acquiring first usage data of a first host machine corresponding to the designated cloud host according to a first preset period, where the first usage data includes at least a CPU usage rate, a memory usage rate, and a network interface bandwidth usage rate;
Analyzing and processing the first usage data, and determining whether the first usage data is greater than a preset standard threshold;
If yes, find out the second host machine whose second usage data is less than the standard threshold from all the host machines;
Migrating the designated cloud host from the first host to the second host.
The secure startup method of a cloud host according to claim 5, wherein after the step of transferring the designated cloud host out of the security isolation area to restore the normal network access function of the designated cloud host, The method also includes:
Acquiring, according to a second preset period, the available resource space in the cloud hard disk corresponding to the designated cloud host;
Analyzing and processing the available resource space, and determining whether the available resource space is less than a preset resource space threshold;
If yes, filter out the junk data in the cloud hard disk;
Sending a garbage removal instruction to the designated cloud host to control the designated cloud host to clear the garbage data according to the garbage removal instruction.
A secure boot device for a cloud host, which is characterized in that it includes:
The first obtaining module is configured to obtain asset information corresponding to the specified cloud host according to preset rules when a user-triggered start instruction for the specified cloud host is received, wherein the asset information includes at least operating system information;
The first judgment module is used to judge whether there are vulnerabilities corresponding to the asset information of the designated cloud host stored in the preset vulnerability database, wherein the number of the vulnerabilities is one or more;
The second obtaining module is configured to, if yes, obtain the last shutdown time of the designated cloud host and the release time corresponding to each of the vulnerabilities;
The second judgment module is used for judging whether there are specified vulnerabilities whose release time is within a specified period of time among all the vulnerabilities according to the last shutdown time and each of the release times;
The isolation module is configured to, if yes, start the designated cloud host in a pre-created security isolation area, wherein the network access function of the specified cloud host will be disabled in the security isolation area.
The secure boot device for a cloud host according to claim 8, wherein the first obtaining module comprises:
The first determining unit is configured to determine whether an asset management tool is installed in the designated cloud host, wherein the asset management tool is used to periodically synchronize asset information corresponding to the designated cloud host to the designated cloud platform;
The first searching unit is configured to search for the asset information from the designated cloud platform if an asset management tool is installed;
The first calling unit is configured to, if the asset management tool is not installed, call a designated scanning tool to perform scanning processing on the designated cloud host, so as to obtain the asset information.
The secure boot device of a cloud host according to claim 8, wherein the device comprises:
The crawling module is used to grab the first vulnerability information related to various operating systems corresponding to the cloud host and the second vulnerability information related to the cloud host service from the target website in real time through a crawler tool;
Create module to create a vulnerability database;
The first storage module is used to store the first vulnerability information and the second vulnerability information in the vulnerability database.
The secure boot device of a cloud host according to claim 8, wherein the device comprises:
The third obtaining module is used to obtain the current time;
The first determining module is configured to determine the time period included between the last shutdown time and the current time as the designated time period.

The secure boot device of a cloud host according to claim 8, wherein the device comprises:
The second determining module is used to determine whether the specified patch corresponding to the specified vulnerability is stored in the vulnerability database;
The second storage module is configured to download the specified patch if it is, and store the specified patch in a specified storage area in the specified cloud host through a specified transmission method;
The first sending module is configured to send an installation instruction for installing the specified patch to the specified cloud host, so that the specified cloud host installs the specified patch according to the installation instruction, and returns the corresponding patch installation result, Wherein, the patch installation result includes installation success or installation failure;
The third judgment module is configured to judge whether the specified patch is successfully installed according to the patch installation result returned by the specified cloud host;
The transfer module is used for transferring the designated cloud host out of the security isolation area, so as to restore the normal network access function of the designated cloud host.
The secure boot device of a cloud host according to claim 12, wherein the device comprises:
The fourth acquiring module is configured to acquire the first usage data of the first host machine corresponding to the designated cloud host according to the first preset period, where the first usage data includes at least a CPU usage rate, a memory usage rate, and Network interface bandwidth usage rate;
The fourth judgment module is configured to analyze and process the first usage data, and judge whether the first usage data is greater than a preset standard threshold;
The searching module is configured to, if yes, find out the second host machine whose second usage data is less than the standard threshold value from all the host machines;
The migration module is used to migrate the designated cloud host from the first host to the second host.
The secure boot device of a cloud host according to claim 12, wherein the device comprises:
A fifth acquiring module, configured to acquire the available resource space in the cloud hard disk corresponding to the designated cloud host according to a second preset period;
The fifth judgment module is configured to analyze and process the available resource space, and judge whether the available resource space is less than a preset resource space threshold;
The screening module is used to screen out junk data in the cloud hard disk if so;
The second sending module is configured to send a garbage removal instruction to the designated cloud host to control the designated cloud host to clear the junk data according to the garbage removal instruction.
A computer device includes a memory and a processor, the memory stores computer-readable instructions, and is characterized in that, when the processor executes the computer-readable instructions, a method for safely starting a cloud host is realized, and the cloud host is Start method, including:
When receiving a user-triggered start instruction for a designated cloud host, acquiring asset information corresponding to the designated cloud host according to a preset rule, where the asset information includes at least operating system information;
Judging whether there are vulnerabilities corresponding to the asset information of the designated cloud host stored in the preset vulnerability database, wherein the number of the vulnerabilities is one or more;
If yes, obtain the last shutdown time of the designated cloud host and the release time corresponding to each of the vulnerabilities;
According to the last shutdown time and each of the release times, determine whether there is a specified vulnerability whose release time is within a specified period of time among all the vulnerabilities;
If yes, start the designated cloud host in a pre-created security isolation zone, wherein the network access function of the designated cloud host will be disabled in the security isolation zone.
The computer device according to claim 15, wherein the step of obtaining asset information corresponding to the designated cloud host according to a preset rule comprises:
Determining whether an asset management tool is installed in the designated cloud host, wherein the asset management tool is used to periodically synchronize asset information corresponding to the designated cloud host to the designated cloud platform;
If an asset management tool is installed, search for the asset information from the designated cloud platform;
If the asset management tool is not installed, call the designated scanning tool to scan the designated cloud host to obtain the asset information.
The computer device according to claim 15, wherein before the step of determining whether a vulnerability

corresponding to the asset information of the designated cloud host is stored in the preset vulnerability database, the method further comprises:
Through the crawler tool, the first vulnerability information related to various operating systems corresponding to the cloud host and the second vulnerability information related to the cloud host service are captured from the target website in real time;
Create a vulnerability database;
The first vulnerability information and the second vulnerability information are stored in the vulnerability database.
A computer-readable storage medium having computer-readable instructions stored thereon, wherein the computer-readable instructions are executed by a processor to realize a secure startup method of a cloud host, and the secure startup method of the cloud host includes:
When receiving a user-triggered start instruction for a designated cloud host, acquiring asset information corresponding to the designated cloud host according to a preset rule, where the asset information includes at least operating system information;
Judging whether there are vulnerabilities corresponding to the asset information of the designated cloud host stored in the preset vulnerability database, wherein the number of the vulnerabilities is one or more;
If yes, obtain the last shutdown time of the designated cloud host and the release time corresponding to each of the vulnerabilities;
According to the last shutdown time and each of the release times, determine whether there is a specified vulnerability whose release time is within a specified period of time among all the vulnerabilities;
If yes, start the designated cloud host in a pre-created security isolation zone, wherein the network access function of the designated cloud host will be disabled in the security isolation zone.
The computer-readable storage medium according to claim 18, wherein the step of obtaining asset information corresponding to the designated cloud host according to a preset rule comprises:
Determining whether an asset management tool is installed in the designated cloud host, wherein the asset management tool is used to periodically synchronize asset information corresponding to the designated cloud host to the designated cloud platform;
If an asset management tool is installed, search for the asset information from the designated cloud platform;
If the asset management tool is not installed, call the designated scanning tool to scan the designated cloud host to obtain the asset information.
The computer-readable storage medium according to claim 18, wherein before the step of determining whether a vulnerability corresponding to the asset information of the designated cloud host is stored in the preset vulnerability database, the method further comprises :
Through the crawler tool, the first vulnerability information related to various operating systems corresponding to the cloud host and the second vulnerability information related to the cloud host service are captured from the target website in real time;
Create a vulnerability database;
The first vulnerability information and the second vulnerability information are stored in the vulnerability database.

Copyright © 2022 Clarivate Analytics. All rights reserved. Republication or redistribution of Clarivate Analytics content, including by framing or similar means, is prohibited without the prior written consent of Clarivate Analytics. Clarivate and its logo are trademarks of the Clarivate Analytics group.

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2021 年 4 月 22 日 (22.04.2021)

WIPO | PCT

(10) 国际公布号
WO 2021/072877 A1

(51) 国际专利分类号:
G06F 9/4401 (2018.01) *G06F 21/53* (2013.01)
G06F 9/455 (2006.01) *G06F 21/57* (2013.01)
G06F 8/61 (2018.01)

(21) 国际申请号: PCT/CN2019/118430

(22) 国际申请日: 2019 年 11 月 14 日 (14.11.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201910979050.1 2019年10月15日 (15.10.2019) CN

(71) 申请人: 平安科技（深圳）有限公司 (PING AN TECHNOLOGY(SHENZHEN)CO.,LTD.) [CN/CN]; 中国广东省深圳市福田区福田街道福安社区益田路5033号平安金融中心23楼, Guangdong 518000 (CN)。

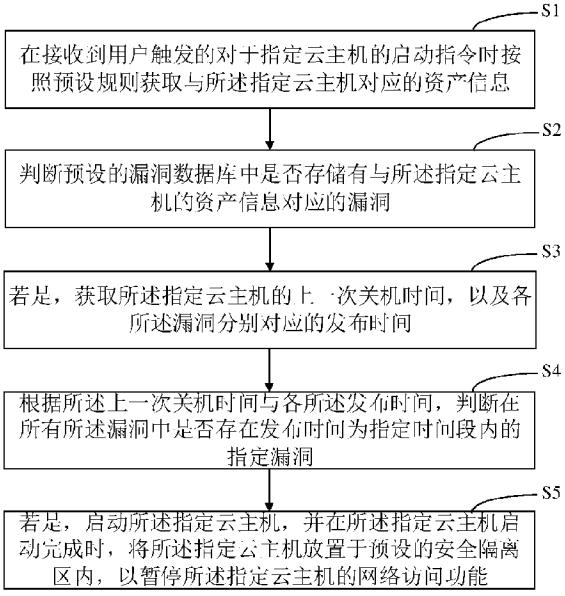
(72) 发明人: 沈勇 (SHEN, Yong); 中国广东省深圳市福田区福田街道福安社区益田路5033号平安金融中心23楼, Guangdong 518000 (CN)。

(74) 代理人: 深圳市明日今典知识产权代理事务所（普通合伙） (SHENZHEN MINGRIJINDIAN INTELLECTUAL PROPERTY AGENCY FIRM(GENERAL)); 中国广东省深圳市南山区南头街道安乐社区关口二路15号智恒产业园30栋4层406室, Guangdong 518000 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB,

(54) **Title:** SECURE STARTING METHOD AND APPARATUS FOR CLOUD HOST, AND COMPUTER DEVICE AND STORAGE MEDIUM

(54) **发明名称:** 云主机的安全启动方法、装置、计算机设备和存储介质



S1 When a starting instruction, triggered by a user, for a specified cloud host is received, acquire, according to a preset rule, asset information corresponding to the specified cloud host
S2 Determine whether there are vulnerabilities, corresponding to the asset information of the specified cloud host, stored in a preset vulnerability database
S3 If so, acquire the last shutdown time of the specified cloud host and release times respectively corresponding to the vulnerabilities
S4 According to the last shutdown time and the release times, determine whether there is a specified vulnerability, of which the release time is within a specified time period, in all vulnerabilities
S5 If so, start the specified cloud host, and place the specified cloud host in a preset secure isolation area when the starting of the specified cloud host is completed, so as to suspend a network access function of the specified cloud host

图 1

(57) **Abstract:** Disclosed are a secure starting method and apparatus for a cloud host, and a computer device and a storage medium. The method comprises: when a starting instruction, triggered by a user, for a specified cloud host is received, acquiring, according to a preset rule, asset information corresponding to the specified cloud host; determining whether there are vulnerabilities, corresponding to the asset information of the specified cloud host, stored in a preset vulnerability database; if so, acquiring the last shutdown time of the specified cloud host and release times respectively corresponding to the vulnerabilities; according to the last shutdown time and the

WO 2021/072877 A1

[见续页]

GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) **指定国**(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布：

— 包括国际检索报告(条约第21条(3))。

release times, determining whether there is a specified vulnerability, of which the release time is within a specified time period, in all the vulnerabilities; and if so, starting the specified cloud host in a pre-created secure isolation area. According to the preset application, the specified cloud host can be effectively prevented from being invaded or attacked by vulnerabilities during use, and the security of the specified cloud host after starting is guaranteed.

(57) 摘要：本申请揭示了一种云主机的安全启动方法、装置、计算机设备和存储介质，其中方法包括：在接收到用户触发的对于指定云主机的启动指令时，按照预设规则获取与指定云主机对应的资产信息；判断预设的漏洞数据库中是否存储有与指定云主机的资产信息对应的漏洞；若是，获取指定云主机的上一次关机时间，以及各漏洞分别对应的发布时间；根据上一次关机时间与各发布时间，判断在所有漏洞中是否存在发布时间为指定时间段内的指定漏洞；若是，在预创建的安全隔离区内启动所述指定云主机。本申请可以有效的避免出现指定云主机在使用过程中受到漏洞的入侵或攻击的情况，保障了指定云主机启动后的安全性。

云主机的安全启动方法、装置、计算机设备和存储介质

[0001] 本申请要求于2019年10月15日提交中国专利局、申请号为2019109790501，发明名称为“云主机的安全启动方法、装置、计算机设备和存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

[0002] 本申请涉及云计算技术领域，具体涉及一种云主机的安全启动方法、装置、计算机设备和存储介质。

背景技术

[0003] 云计算当前是IT技术发展的一个重要领域和方向，并正在逐步渗透到社会的各领域。云主机是云计算在基础设施应用上的重要组成部分，位于云计算产业链金字塔底层，产品源自云计算平台。该平台整合了互联网应用三大核心要素：计算、存储、网络，面向用户提供公用化的互联网基础设施服务。云主机是一种类似VPS(Virtual Private Server，虚拟专用服务器)主机的虚拟化技术，VPS是采用虚拟软件(如VZ或VM)在一台主机上虚拟出多个类似独立主机的部分，能够实现单机多用户，每个部分都可以做单独的操作系统，管理方法同主机相同。而云主机是在一组宿主机集群上虚拟出多个类似独立主机的部分，集群中每个宿主机上都有云主机的一个镜像，从而大大提高了虚拟主机的安全稳定性。另外，为了便于用户随时随地访问云计算，云主机通常能从公网（Internet）直接访问，甚至是管理。

[0004] 在现有的各类云计算系统中，各个云主机均会面临着防护间隙的安全风险。具体地，在云主机处于关机状态时，由于云主机在关机时会对当前所有已知的安全漏洞进行修复处理，因而可认为处于关机状态的云主机也处于一个安全的状态。但是，当云主机被重新启动并暴露在公网时，此时的外界环境相比于关机前可能会发生改变。如果当前出现了与云主机相关的新的安全漏洞，则云主机随时可能会遭受到公网上的攻击者的攻击，云主机的安全性受到极大的威胁。

发明概述

技术问题

[0005] 本申请的主要目的为提供一种云主机的安全启动方法、装置、计算机设备和存储介质，旨在解决在现有的各类云计算系统中，当云主机从关机状态被重新启动并暴露在公网时，随时可能会遭受到公网上的攻击者的攻击，云主机的安全性容易受到极大的威胁的技术问题。

问题的解决方案

技术解决方案

[0006] 为了实现上述申请目的，本申请提出一种云主机的安全启动方法，所述方法包括步骤：

[0007] 在接收到用户触发的对于指定云主机的启动指令时按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

[0008] 若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

[0009] 根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

[0010] 若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。

[0011] 本申请还提供一种云主机的安全启动装置，包括：

[0012] 第一获取模块，用于在接收到用户触发的对于指定云主机的启动指令时按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；

[0013] 第一判断模块，用于判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

[0014] 第二获取模块，用于若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

[0015] 第二判断模块，用于根据所述上一次关机时间与各所述发布时间，判断在所有

所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

[0016] 隔离模块，用于若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。

[0017] 本申请还提供一种计算机设备，包括存储器和处理器，所述存储器中存储有计算机可读指令，所述处理器执行所述计算机可读指令时实现上述方法的步骤。

[0018] 本申请还提供一种计算机可读存储介质，其上存储有计算机可读指令，所述计算机可读指令被处理器执行时实现上述方法的步骤。

发明的有益效果

有益效果

[0019] 本申请中提供的云主机的安全启动方法、装置、计算机设备和存储介质，在接收到用户触发的对于指定云主机的启动指令时，按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。本申请当指定云主机在上一次关机期间时出现了新的与其相关的漏洞的情况下，会智能的将指定云主机放置在预创建的安全隔离区内进行启动，从而在指定云主机启动后暂时禁用指定云主机的网络访问功能，有效的避免出现指定云主机在使用过程中受到漏洞的入侵或攻击的情况，保障了指定云主机启动后的安全性。

对附图的简要说明

附图说明

[0020] 图1是本申请一实施例的云主机的安全启动方法的流程示意图；

[0021] 图2是本申请一实施例的云主机的安全启动装置的结构示意图；

[0022] 图3是本申请一实施例的计算机设备的结构示意图。

实施该发明的最佳实施例

本发明的最佳实施方式

[0023] 应当理解，此处所描述的具体实施例仅仅用于解释本申请，并不用于限定本申请。

[0024] 参照图1，本申请一实施例的云主机的安全启动方法，包括：

[0025] S1：在接收到用户触发的对于指定云主机的启动指令时按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；

[0026] S2：判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

[0027] S3：若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

[0028] S4：根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

[0029] S5：若是，启动所述指定云主机，并在所述指定云主机启动完成时，将所述指定云主机放置于预设的安全隔离区内，以暂停所述指定云主机的网络访问功能。

[0030] 如上述步骤S1至S5所述，本发明实施例的执行主体为一种云主机的安全启动装置，具体可为用于虚拟出云主机的宿主机集群，通过该宿主机集群可实现对于指定云主机的安全启动，上述指定云主机为宿主机集群虚拟出的所有云主机中任意一台云主机。具体地，首先判断是否接收到用户触发的对于指定云主机的启动指令；当接收到该启动指令时，按照预设规则获取与指定云主机对应的资产信息，其中，上述资产信息至少包括操作系统信息，例如可为操作系统为windowsserver2012，使用RDP服务11.0版本提供远程桌面，另外该资产信息还可包括云主机服务信息与关键应用信息。另外，上述预设规则可包括：通过查询与指定云主机对应的云平台的方式，或者通过调用指定扫描工具的方式来获取与指定云主机对应的资产信息。然后，根据上述资产信息，判断预设的漏洞数据库中是否存储有与上述指定云主机的资产信息对应的漏洞，其中，上述漏洞的数量为一个或多个，另外，上述漏洞数据库内存储有通过爬虫工具实时从目标

网站抓取与云主机对应的各类操作系统相关的漏洞信息，以及与云主机服务相关的漏洞信息。如果漏洞数据库中存储有与上述指定云主机的资产信息对应的漏洞，则获取上述指定云主机的上一次关机时间以及各上述漏洞分别对应的发布时间。其中，举例地，爬取的任一漏洞信息的形式可为CVE20190004-rdp漏洞，2019/07/02发布，受影响系统为rdp服务12.0版本之前的所有版本，则对应的漏洞的发布时间为2019/07/02。然后根据上述上一次关机时间与各上述发布时间，判断在所有漏洞中是否存在发布时间为指定时间段内的指定漏洞，其中，上述指定时间段为上述关机时间与上述当前时间之间所包含的时间段。如果存在发布时间为指定时间段内的指定漏洞，则在预创建的安全隔离区内启动上述指定云主机，其中，指定云主机的网络访问功能在上述安全隔离区内会被禁用。本实施例当指定云主机在上一次关机期间时出现了新的与其相关的漏洞的情况下，会智能的将指定云主机放置在预创建的安全隔离区内进行启动，从而在指定云主机启动后暂时禁用指定云主机的网络访问功能，有效的避免出现指定云主机在使用过程中受到漏洞的入侵或攻击的情况，保障了指定云主机启动后的安全性。进一步地，如果在所有漏洞中不存在发布时间为指定时间段内的指定漏洞，表明在指定云主机的上一次关机期间并未出现与其相关的漏洞，则直会接完成对于指定云主机的启动，以及不用将指定云主机放置于安全隔离区。以有效的保证用户对于指定云主机的使用体验。

[0031] 进一步地，本申请一实施例中，上述步骤S1，包括：

[0032] S100：确定所述指定云主机内是否安装有资产管理工具，其中，所述资产管理工具用于定期将所述指定云主机对应的资产信息同步至指定云平台；

[0033] S101：若安装有资产管理工具，从所述指定云平台中查找出所述资产信息；

[0034] S102：若未安装有资产管理工具，调用指定扫描工具对所述指定云主机进行扫描处理，以得到所述资产信息。

[0035] 如上述步骤S100至S102所述，上述资产信息具体可包括指定云主机对应的操作系统信息，以及指定云主机内的云主机服务信息与关键应用信息，可通过查询与指定云主机对应的云平台的方式，或者通过调用指定扫描工具的方式来获取与指定云主机对应的资产信息。具体地，首先确定上述指定云主机内是否安装

有资产管理工具，其中，上述资产管理工具具有对指定云主机的资产信息的同步功能，当指定云主机内安装有资产管理工具时，则指定云主机会定期通过上述资产管理工具来将内部的资产信息同步至对应的上述指定云平台，从而该指定云平台会存储有指定云主机的资产信息。如果确定出指定云主机内安装有资产管理工具，则可从与上述指定云主机对应的指定云平台中查找出上述资产信息。而如果指定云主机内未安装有资产管理工具，则可调用指定扫描工具对上述云平台进行扫描处理，以得到上述资产信息，其中，上述指定扫描工具可为nmap，可通过nmap工具来对该指定云主机发送操作系统探测命令$nmap-O[target IP]，使得指定云主机在接收到该作系统探测命令后会查询自身的资产信息，进而可以得到由指定云主机返回的资产信息。在得到了指定云主机的资产信息后，后续便可智能根据该资产信息来识别出漏洞数据库中是否存在与指定云主机对应的漏洞。

[0036] 进一步地，本申请一实施例中，上述步骤S2之前，包括：

[0037] S200：通过爬虫工具，实时从目标网站抓取出与云主机对应的各类操作系统相关的第一漏洞信息，以及与云主机服务相关的第二漏洞信息；

[0038] S201：创建一个漏洞数据库；

[0039] S202：将所述第一漏洞信息与所述第二漏洞信息存储于所述漏洞数据库内。

[0040] 如上述步骤S200至S202所述，在根据上述资产信息，进行判断预设的漏洞数据库中是否存储有与上述指定云主机对应的漏洞的判断过程之前，还包括生成该漏洞数据库的生成过程。具体地，首先通过爬虫工具，实时从目标网站抓取与云主机对应的各类操作系统相关的第一漏洞信息，以及与云主机服务相关的第二漏洞信息。其中，上述第一漏洞信息可包括与第一漏洞对应的漏洞信息，以及与第一漏洞对应的第一补丁信息。同理，上述第二漏洞信息可包括与第二漏洞对应的漏洞信息，以及与第二漏洞对应的第二补丁信息。另外，爬虫工具是指网络爬虫(又被称为网页蜘蛛或网络机器人)，是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。该爬虫工具包括但不限于Python爬虫工具。另外，上述目标网站为用于发布与云主机相关漏洞的国内外权威网站。具体地，宿主机集群采用爬虫工具执行爬虫文件，以爬取符合爬虫文件设置的数据爬取

条件的漏洞信息。其中，该爬虫文件包括但不限于目标URL和搜索关键词这两个数据爬取条件。该目标URL是该爬虫文件中用于限定所需爬取的第一漏洞信息与第二漏洞信息对应的目标网站的URL，该URL(Uniform Resource Locator的简称，即统一资源定位符)是对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。搜索关键词是指该爬虫文件中用于限定该爬虫文件所需爬取的第一漏洞信息与第二漏洞信息的共同特性的关键词。在得到上述第一漏洞信息与第二漏洞信息后，再创建一个漏洞数据库，然后将该第一漏洞信息与第二漏洞信息存储于上述漏洞数据库内，以形成后续用于确定是否存在可能会威胁指定云主机安全的漏洞的漏洞数据库。

[0041] 进一步地，本申请一实施例中，上述步骤S4之前，包括：

[0042] S400：获取当前时间；

[0043] S401：将所述上一次关机时间与所述当前时间之间所包含的时间段，确定为所述指定时间段。

[0044] 如上述步骤S400至S401所述，上述判断所有上述漏洞中是否存在发布时间为指定时间段内的指定漏洞的步骤之前，还包括确定出该指定时间段的步骤，具体地，首先获取当前时间，其中，上述当前时间与上述上一次关机时间的精确度相同，但对该精确度不做具体限定，例如上述精确度可精确到年月日，例如当前时间可为2019/07/11，或者还可精确到年月日时等，在此不做过多叙述。在得到当前时间以及上述上一次关机时间后，再将该上一次关机时间与上述当前时间之间所包含的时间段确定为上述指定时间段。举例地，如果当前时间为2019/07/11，上一次关机时间为2019/07/01，则可确定出上述指定时间段包括：2019/07/01-2019/07/11。本实施例根据当前时间与指定云主机的上一次关机时间来确定出指定时间段，有利于后续根据该指定时间段来精确的识别出漏洞数据库中是否存有会威胁指定云主机安全的指定漏洞，进而根据识别结果来智能的选择对于指定云主机的开启方式。

[0045] 进一步地，本申请一实施例中，上述步骤S5之后，包括：

[0046] S500：确定所述漏洞数据库中是否存储有与所述指定漏洞对应的指定补丁；

[0047] S501：若是，下载所述指定补丁，并通过指定传输方式将所述指定补丁存储于

所述指定云主机内的指定存储区域中；

[0048] S502：向所述指定云主机发送安装所述指定补丁的安装指令，以便所述指定云主机根据所述安装指令对所述指定补丁进行安装，并返回对应的补丁安装结果，其中，所述补丁安装结果包括安装成功或安装失败；

[0049] S503：根据所述指定云主机返回的所述补丁安装结果，判断所述指定补丁是否安装成功；

[0050] S504：若是，将所述指定云主机转移出所述安全隔离区，以恢复所述指定云主机正常的网络访问功能。

[0051] 如上述步骤S500至S504所述，在将指定云主机放置于安全隔离区后，只有在该指定云主机成功安装成功了与上述指定漏洞对应的指定补丁后，才会将该指定云主机从安全隔离区内移除，以保证指定云主机的使用安全。具体地，首先判断上述漏洞数据库是否存储有与上述指定漏洞对应的指定补丁。如果漏洞数据库内存储有上述指定补丁，则会下载该指定补丁，并通过指定传输方式将该指定补丁存储于指定云主机内的指定存储区域中。其中，上述指定传输方式为局域网传输方式或蓝牙传输方式，宿主机集群在下载了上述指定补丁后，通过在内部利用该指定传输方式将下载好的指定补丁传输至指定云主机内的指定存储区域，由于传输过程并不需要使用到网络资源，使得指定云主机在接收到指定补丁时也不会遭受公网上的攻击者的攻击，有效的保证了指定云主机在存储指定补丁的过程中的安全性，而且指定云主机对指定补丁进行安装的安装过程也不需要网络，进一步增加了指定云主机的使用安全性。另外，对于上述指定存储区域不作具体限定，可以根据实际需求进行设置，例如可以是一个新创建的专门用于存储该指定补丁的区域，也可以是指定云主机中的存储空间较大的一个区域，等等。在完成了将指定补丁存储于上述指定存储区域中后，会向上述指定云主机发送安装指定补丁的安装指令，以便指定云主机根据上述安装指令下载并安装该指定补丁，并返回对应的补丁安装结果，其中，所述补丁安装结果包括安装成功或安装失败。然后根据指定云主机返回的补丁安装结果，进一步判断上述指定补丁是否安装成功。如果指定补丁在指定云主机内安装成功，则会将指定云主机转移出上述安全隔离区，以恢复指定云主机正常的网络访问

功能，从而使得指定云主机在恢复网络访问功能后可以进行正常的工作，保证用户在使用指定云主机时的体验感。进一步地，如果指定补丁安装失败，指定云主机会返回安装失败的补丁安装结果，则此时不会将指定云主机转移出上述安全隔离区，而会先查找出指定补丁安装失败的失败原因；然后根据该指定补丁安装失败的失败原因，控制该指定云主机对上述指定补丁进行重新下载与安装，并在确定出指定补丁在指定云主机内安装成功后，再将指定云主机转移出上述安全隔离区，使得只有在指定补丁安装成功的情况下，才会恢复指定云主机正常的网络访问功能，有效的保障了指定云主机的使用安全。

[0052] 本申请一实施例中，上述步骤S504之后，包括：

[0053] S5040：按照第一预设周期获取与所述指定云主机对应的第一宿主机的第一使用数据，其中，所述第一使用数据至少包括CPU使用率、内存使用率以及网络接口带宽使用率；

[0054] S5041：对所述第一使用数据进行分析处理，判断所述第一使用数据是否大于预设的标准阈值；

[0055] S5042：若是，从所有的宿主机中查找出第二使用数据小于所述标准阈值的第二宿主机；

[0056] S5043：将所述指定云主机从所述第一宿主机迁移至所述第二宿主机。

[0057] 如上述步骤S5040至S5043所述，在将指定云主机转移出上述安全隔离区后，在指定云主机的正常工作过程中，如果出现了指定云主机对应的第一宿主机处于高负荷状态的情况，宿主机集群还具有对于指定云主机的智能迁移的功能。具体地，在上述指定云主机启动完成后，按照第一预设周期获取与上述指定云主机对应的第一宿主机的第一使用数据，其中，对上述第一预设周期不作具体限定，例如可设为一个小时。另外，上述使用数据包括CPU使用率、内存使用率以及网络接口带宽使用率等。然后对该第一使用数据进行分析处理，并判断上述第一使用数据是否大于预设的标准阈值，其中，对于不同类型的第一使用数据，设置有该类型一一对应的相同数量的标准阈值。另外，对标准阈值的具体数值不作具体限定，可由宿主机集群设定，或者也可由用户根据需求进行设置。举例地，与CPU使用率对应的标准阈值可设为80%，与内存使用率对应的标准阈

值可设为75%，与网络接口带宽使用率的标准阈值可设为85%。如果上述第一使用数据大于预设的标准阈值，则表明与指定云主机对应的第一宿主机当前处于高负荷状态，第一宿主机的可计算资源较少。此时会从宿主机集群内的所有宿主机中查找出第二使用数据小于上述标准阈值的第二宿主机，并将指定云主机从上述第一宿主机迁移至上述第二宿主机，以实现将指定云主机从负载过高的第一宿主机智能的迁移到负载较低的第二宿主机，使得指定云主机的计算资源得到优化，从而有效的提升指定云主机的运行效率和响应速度，也保障了第一宿主机的使用安全。进一步地，如果第二宿主机的数量为多个，则从所有的第二宿主机中筛选出与第二使用数据对应的数据值最低的指定宿主机，再将上述指定云主机从上述第一宿主机迁移至上述指定宿主机。通过从所有的二宿主机中筛选出第二使用数据对应的数据值最低的指定宿主机，不仅可以提升指定云主机的运行效率和响应速度，还有利于实现对于所有宿主机的负载均衡。

[0058] 进一步地，本申请一实施例中，上述步骤S504之后，包括：

[0059] S5044：按照第二预设周期获取与所述指定云主机对应的云硬盘内的可用资源空间；

[0060] S5045：对所述可用资源空间进行分析处理，判断所述可用资源空间是否小于预设的资源空间阈值；

[0061] S5046：若是，筛选出所述云硬盘中的垃圾数据；

[0062] S5047：向所述指定云主机发送垃圾清除指令，以控制所述指定云主机根据所述垃圾清除指令，对所述垃圾数据进行清除处理。

[0063] 如上述步骤S5044至S5047所述，在将指定云主机转移出上述安全隔离区后，后续还会对处于正常工作状态的指定云主机中的可用资源空间进行监控处理，以保证指定云主机可具备充足的可用资源空间。具体地，在上述指定云主机启动后，按照第二预设周期获取与上述指定云主机对应的云硬盘内的可用资源空间，其中，对上述第二预设周期不作具体限定，例如可设为一个小时。然后判断上述可用资源空间是否小于预设的资源空间阈值，其中，对上述资源空间阈值的具体数值不作具体限定，可由宿主机集群设定，或者也可由用户根据需求进行设置，举例地，资源空间阈值可设为5g。如果可用资源空间小于预设的资源

空间阈值，表明指定云主机当前的可用资源空间较少，与指定云主机对应的云硬盘的使用压力较大。此时会筛选出上述云硬盘中的垃圾数据，其中，上述垃圾数据包括残余的指定云主机配置文件、无用的临时文件、已删除/卸载后残留的无用文件、硬盘和/或卷格式化未成功残留的文件等。在筛选出该垃圾数据后，再向上述指定云主机发送垃圾清除指令，以控制上述指定云主机根据该垃圾清除指令，对上述垃圾数据进行清除处理，具体可通过指定云主机上的管理工具或应用软件来实现对于上述垃圾数据的自动清除。本实施例在指定云主机当前的可用资源较少时，会自动并智能的对与指定云主机对应的云硬盘内的垃圾数据进行清除，有效的减少了指定云主机的压力和负担，释放指定云主机上多余、无用的资源，从而实现有序并及时地处理指定云主机对应的云硬盘内的垃圾数据，避免不必要的资源浪费，提高指定云主机内部资源的利用效率，最终提升指定云主机的运行效率和响应速度。

[0064] 参照图2，本申请一实施例中还提供了一种云主机的安全启动装置，包括：

[0065] 第一获取模块1，用于在接收到用户触发的对于指定云主机的启动指令时按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；

[0066] 第一判断模块2，用于判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

[0067] 第二获取模块3，用于若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

[0068] 第二判断模块4，用于根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

[0069] 隔离模块5，用于若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。

[0070] 本实施例中，上述云主机的安全启动装置中的第一获取模块、第一判断模块、第二获取模块、第二判断模块与隔离模块的功能和作用的实现过程具体详见上述云主机的安全启动方法中对应步骤S1至S5的实现过程，在此不再赘述。

[0071] 进一步地，本申请一实施例中，上述第一获取模块，包括：

[0072] 第一确定单元，用于确定所述指定云主机内是否安装有资产管理工具，其中，所述资产管理工具用于定期将所述指定云主机对应的资产信息同步至指定云平台；

[0073] 第一查找单元，用于若安装有资产管理工具，从所述指定云平台中查找出所述资产信息；

[0074] 第一调用单元，用于若未安装有资产管理工具，调用指定扫描工具对所述指定云主机进行扫描处理，以得到所述资产信息。

[0075] 本实施例中，上述云主机的安全启动装置中的第一确定单元、第一查找单元与第一调用单元的功能和作用的实现过程具体详见上述云主机的安全启动方法中对应步骤S100至S102的实现过程，在此不再赘述。

[0076] 进一步地，本申请一实施例中，上述云主机的安全启动装置，包括：

[0077] 抓取模块，用于通过爬虫工具，实时从目标网站抓取出与云主机对应的各类操作系统相关的第一漏洞信息，以及与云主机服务相关的第二漏洞信息；

[0078] 创建模块，用于创建一个漏洞数据库；

[0079] 第一存储模块，用于将所述第一漏洞信息与所述第二漏洞信息存储于所述漏洞数据库内。

[0080] 本实施例中，上述云主机的安全启动装置中的抓取模块、创建模块与第一存储模块的功能和作用的实现过程具体详见上述云主机的安全启动方法中对应步骤S200至S202的实现过程，在此不再赘述。

[0081] 进一步地，本申请一实施例中，上述云主机的安全启动装置，包括：

[0082] 第三获取模块，用于获取当前时间；

[0083] 第一确定模块，用于将所述上一次关机时间与所述当前时间之间所包含的时间段，确定为所述指定时间段。

[0084] 本实施例中，上述云主机的安全启动装置中的第三获取模块与第一确定模块的功能和作用的实现过程具体详见上述云主机的安全启动方法中对应步骤S400至S401的实现过程，在此不再赘述。

[0085] 进一步地，本申请一实施例中，上述云主机的安全启动装置，包括：

[0086] 第二确定模块，用于确定所述漏洞数据库中是否存储有与所述指定漏洞对应的

指定补丁；

[0087] 第二存储模块，用于若是，下载所述指定补丁，并通过指定传输方式将所述指定补丁存储于所述指定云主机内的指定存储区域中；

[0088] 第一发送模块，用于向所述指定云主机发送安装所述指定补丁的安装指令，以便所述指定云主机根据所述安装指令对所述指定补丁进行安装，并返回对应的补丁安装结果，其中，所述补丁安装结果包括安装成功或安装失败；

[0089] 第三判断模块，用于根据所述指定云主机返回的所述补丁安装结果，判断所述指定补丁是否安装成功；

[0090] 转移模块，用于若是，将所述指定云主机转移出所述安全隔离区，以恢复所述指定云主机正常的网络访问功能。

[0091] 本实施例中，上述云主机的安全启动装置中的第二确定模块、第二存储模块、第一发送模块、第三判断模块与转移模块的功能和作用的实现过程具体详见上述云主机的安全启动方法中对应步骤S500至S504的实现过程，在此不再赘述。

[0092] 进一步地，本申请一实施例中，上述云主机的安全启动装置，包括：

[0093] 第四获取模块，用于按照第一预设周期获取与所述指定云主机对应的第一宿主机的第一使用数据，其中，所述第一使用数据至少包括CPU使用率、内存使用率以及网络接口带宽使用率；

[0094] 第四判断模块，用于对所述第一使用数据进行分析处理，判断所述第一使用数据是否大于预设的标准阈值；

[0095] 查找模块，用于若是，从所有的宿主机中查找出第二使用数据小于所述标准阈值的第二宿主机；

[0096] 迁移模块，用于将所述指定云主机从所述第一宿主机迁移至所述第二宿主机。

[0097] 本实施例中，上述云主机的安全启动装置中的第四获取模块、第四判断模块、查找模块与迁移模块的功能和作用的实现过程具体详见上述云主机的安全启动方法中对应步骤S5040至S5043的实现过程，在此不再赘述。

[0098] 进一步地，本申请一实施例中，上述云主机的安全启动装置，包括：

[0099] 第五获取模块，用于按照第二预设周期获取与所述指定云主机对应的云硬盘内的可用资源空间；

[0100] 第五判断模块，用于对所述可用资源空间进行分析处理，判断所述可用资源空间是否小于预设的资源空间阈值；

[0101] 筛选模块，用于若是，筛选出所述云硬盘中的垃圾数据；

[0102] 第二发送模块，用于向所述指定云主机发送垃圾清除指令，以控制所述指定云主机根据所述垃圾清除指令，对所述垃圾数据进行清除处理。

[0103] 本实施例中，上述云主机的安全启动装置中的第五获取模块、第五判断模块、筛选模块与第二发送模块的功能和作用的实现过程具体详见上述云主机的安全启动方法中对应步骤S5044至S5047的实现过程，在此不再赘述。

[0104] 参照图3，本申请实施例中还提供一种计算机设备，该计算机设备可以是服务器，其内部结构可以如图3所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中，该计算机设备设计的处理器用于提供计算和控制能力。该计算机设备的存储器包括易失性或非易失性存储介质、内存储器。该易失性或非易失性存储介质存储有操作系统、计算机可读指令和数据库。该内存储器为易失性或非易失性存储介质中的操作系统和计算机可读指令的运行提供环境。该计算机设备的数据库用于存储资产信息、漏洞等数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机可读指令被处理器执行时以实现上述任一个示例性实施例所示出的云主机的安全启动方法。本领域技术人员可以理解，图3中示出的结构，仅仅是与本申请方案相关的部分结构的框图，并不构成对本申请方案所应用于其上的装置、计算机设备的限定。

[0105] 本申请一实施例还提供一种计算机可读存储介质，该计算机可读存储介质可以是易失性的，也可以是非易失性的，其上存储有计算机可读指令，该计算机可读指令被一个或多个处理器执行时，使得一个或多个处理器执行时实现上述云主机的安全启动方法实施例中的步骤。

[0106] 综上所述，本申请实施例中提供的云主机的安全启动方法、装置、计算机设备和存储介质，在接收到用户触发的对于指定云主机的启动指令时，按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息

对应的漏洞，其中，所述漏洞的数量为一个或多个；若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。本申请当指定云主机在上一次关机期间时出现了新的与其相关的漏洞的情况下，会智能的将指定云主机放置在预创建的安全隔离区内进行启动，从而在指定云主机启动后暂时禁用指定云主机的网络访问功能，有效的避免出现指定云主机在使用过程中受到漏洞的入侵或攻击的情况，保障了指定云主机启动后的安全性。

[0107] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机可读指令来指令相关的硬件来完成，所述的计算机可读指令可存储于一非易失性或易失性计算机可读取存储介质中，该计算机可读指令在执行时，可包括如上述各方法的实施例的流程。其中，本申请所提供的和实施例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可以包括只读存储器（ROM）、可编程ROM（PROM）、电可编程ROM（EPROM）、电可擦除可编程ROM（EEPROM）或闪存。易失性存储器可包括随机存取存储器（RAM）或者外部高速缓冲存储器。作为说明而非局限，RAM通过多种形式可得，诸如静态RAM（SRAM）、动态RAM（DRAM）、同步DRAM（SDRAM）、双速据率SDRAM（SSRSDRAM）、增强型SDRAM（ESDRAM）、同步链路（Synchlink）DRAM（SLDRAM）、存储器总线（Rambus）直接RAM（RDRAM）、直接存储器总线动态RAM（DRDRAM）、以及存储器总线动态RAM（RDRAM）等。

[0108] 以上所述仅为本申请的优选实施例，并非因此限制本申请的专利范围，凡是利用本申请说明书及附图内容所作的等效结构或等效流程变换，或直接或间接运用在其他相关的技术领域，均同理包括在本申请的专利保护范围内。

权利要求书

[权利要求 1] 一种云主机的安全启动方法，其特征在于，包括：

在接收到用户触发的对于指定云主机的启动指令时，按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；

判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。

[权利要求 2] 根据权利要求1所述的云主机的安全启动方法，其特征在于，所述按照预设规则获取与所述指定云主机对应的资产信息的步骤，包括：

确定所述指定云主机内是否安装有资产管理工具，其中，所述资产管理工具用于定期将所述指定云主机对应的资产信息同步至指定云平台；

若安装有资产管理工具，从所述指定云平台中查找出所述资产信息；

若未安装有资产管理工具，调用指定扫描工具对所述指定云主机进行扫描处理，以得到所述资产信息。

[权利要求 3] 根据权利要求1所述的云主机的安全启动方法，其特征在于，所述判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞的步骤之前，所述方法还包括：

通过爬虫工具，实时从目标网站抓取出与云主机对应的各类操作系统相关的第一漏洞信息，以及与云主机服务相关的第二漏洞信息；

创建一个漏洞数据库；

将所述第一漏洞信息与所述第二漏洞信息存储于所述漏洞数据库内。

[权利要求 4] 根据权利要求1所述的云主机的安全启动方法，其特征在于，所述根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞的步骤之前，所述方法还包括：

获取当前时间；

将所述上一次关机时间与所述当前时间之间所包含的时间段，确定为所述指定时间段。

[权利要求 5] 根据权利要求1所述的云主机的安全启动方法，其特征在于，所述在预创建的安全隔离区内启动所述指定云主机的步骤之后，所述方法还包括：

确定所述漏洞数据库中是否存储有与所述指定漏洞对应的指定补丁；

若是，下载所述指定补丁，并通过指定传输方式将所述指定补丁存储于所述指定云主机内的指定存储区域中；

向所述指定云主机发送安装所述指定补丁的安装指令，以便所述指定云主机根据所述安装指令对所述指定补丁进行安装，并返回对应的补丁安装结果，其中，所述补丁安装结果包括安装成功或安装失败；

根据所述指定云主机返回的所述补丁安装结果，判断所述指定补丁是否安装成功；

若是，将所述指定云主机转移出所述安全隔离区，以恢复所述指定云主机正常的网络访问功能。

[权利要求 6] 根据权利要求5所述的云主机的安全启动方法，其特征在于，所述将所述指定云主机转移出所述安全隔离区，以恢复所述指定云主机正常的网络访问功能的步骤之后，所述方法还包括：

按照第一预设周期获取与所述指定云主机对应的第一宿主机的第一使用数据，其中，所述第一使用数据至少包括CPU使用率、内存使用率以及网络接口带宽使用率；

对所述第一使用数据进行分析处理，判断所述第一使用数据是否大于预设的标准阈值；

若是，从所有的宿主机中查找出第二使用数据小于所述标准阈值的第二宿主机；

将所述指定云主机从所述第一宿主机迁移至所述第二宿主机。

[权利要求 7] 根据权利要求5所述的云主机的安全启动方法，其特征在于，所述将所述指定云主机转移出所述安全隔离区，以恢复所述指定云主机正常的网络访问功能的步骤之后，所述方法还包括：

按照第二预设周期获取与所述指定云主机对应的云硬盘内的可用资源空间；

对所述可用资源空间进行分析处理，判断所述可用资源空间是否小于预设的资源空间阈值；

若是，筛选出所述云硬盘中的垃圾数据；

向所述指定云主机发送垃圾清除指令，以控制所述指定云主机根据所述垃圾清除指令，对所述垃圾数据进行清除处理。

[权利要求 8] 一种云主机的安全启动装置，其特征在于，包括：

第一获取模块，用于在接收到用户触发的对于指定云主机的启动指令时按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；

第一判断模块，用于判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

第二获取模块，用于若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

第二判断模块，用于根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

隔离模块，用于若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。

[权利要求 9] 根据权利要求8所述的云主机的安全启动装置，其特征在于，所述第

一获取模块，包括：

第一确定单元，用于确定所述指定云主机内是否安装有资产管理工具，其中，所述资产管理工具用于定期将所述指定云主机对应的资产信息同步至指定云平台；

第一查找单元，用于若安装有资产管理工具，从所述指定云平台中查找出所述资产信息；

第一调用单元，用于若未安装有资产管理工具，调用指定扫描工具对所述指定云主机进行扫描处理，以得到所述资产信息。

[权利要求 10] 根据权利要求8所述的云主机的安全启动装置，其特征在于，所述装置包括：

抓取模块，用于通过爬虫工具，实时从目标网站抓取出与云主机对应的各类操作系统相关的第一漏洞信息，以及与云主机服务相关的第二漏洞信息；

创建模块，用于创建一个漏洞数据库；

第一存储模块，用于将所述第一漏洞信息与所述第二漏洞信息存储于所述漏洞数据库内。

[权利要求 11] 根据权利要求8所述的云主机的安全启动装置，其特征在于，所述装置包括：

第三获取模块，用于获取当前时间；

第一确定模块，用于将所述上一次关机时间与所述当前时间之间所包含的时间段，确定为所述指定时间段。

[权利要求 12] 根据权利要求8所述的云主机的安全启动装置，其特征在于，所述装置包括：

第二确定模块，用于确定所述漏洞数据库中是否存储有与所述指定漏洞对应的指定补丁；

第二存储模块，用于若是，下载所述指定补丁，并通过指定传输方式将所述指定补丁存储于所述指定云主机内的指定存储区域中；

第一发送模块，用于向所述指定云主机发送安装所述指定补丁的安装

指令，以便所述指定云主机根据所述安装指令对所述指定补丁进行安装，并返回对应的补丁安装结果，其中，所述补丁安装结果包括安装成功或安装失败；

第三判断模块，用于根据所述指定云主机返回的所述补丁安装结果，判断所述指定补丁是否安装成功；

转移模块，用于若是，将所述指定云主机转移出所述安全隔离区，以恢复所述指定云主机正常的网络访问功能。

[权利要求 13] 根据权利要求12所述的云主机的安全启动装置，其特征在于，所述装置包括：

第四获取模块，用于按照第一预设周期获取与所述指定云主机对应的第一宿主机的第一使用数据，其中，所述第一使用数据至少包括CPU使用率、内存使用率以及网络接口带宽使用率；

第四判断模块，用于对所述第一使用数据进行分析处理，判断所述第一使用数据是否大于预设的标准阈值；

查找模块，用于若是，从所有的宿主机中查找出第二使用数据小于所述标准阈值的第二宿主机；

迁移模块，用于将所述指定云主机从所述第一宿主机迁移至所述第二宿主机。

[权利要求 14] 根据权利要求12所述的云主机的安全启动装置，其特征在于，所述装置包括：

第五获取模块，用于按照第二预设周期获取与所述指定云主机对应的云硬盘内的可用资源空间；

第五判断模块，用于对所述可用资源空间进行分析处理，判断所述可用资源空间是否小于预设的资源空间阈值；

筛选模块，用于若是，筛选出所述云硬盘中的垃圾数据；

第二发送模块，用于向所述指定云主机发送垃圾清除指令，以控制所述指定云主机根据所述垃圾清除指令，对所述垃圾数据进行清除处理。

[权利要求 15] 一种计算机设备，包括存储器和处理器，所述存储器存储有计算机可读指令，其特征在于，所述处理器执行所述计算机可读指令时实现云主机的安全启动方法，该云主机的安全启动方法，包括：

在接收到用户触发的对于指定云主机的启动指令时，按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；

判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。

[权利要求 16] 根据权利要求15所述的计算机设备，其特征在于，所述按照预设规则获取与所述指定云主机对应的资产信息的步骤，包括：

确定所述指定云主机内是否安装有资产管理工具，其中，所述资产管理工具用于定期将所述指定云主机对应的资产信息同步至指定云平台；

若安装有资产管理工具，从所述指定云平台中查找出所述资产信息；

若未安装有资产管理工具，调用指定扫描工具对所述指定云主机进行扫描处理，以得到所述资产信息。

[权利要求 17] 根据权利要求15所述的计算机设备，其特征在于，所述判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞的步骤之前，所述方法还包括：

通过爬虫工具，实时从目标网站抓取出与云主机对应的各类操作系统相关的第一漏洞信息，以及与云主机服务相关的第二漏洞信息；

创建一个漏洞数据库；

将所述第一漏洞信息与所述第二漏洞信息存储于所述漏洞数据库内。

[权利要求 18] 一种计算机可读存储介质，其上存储有计算机可读指令，其特征在于，所述计算机可读指令被处理器执行时实现云主机的安全启动方法，该云主机的安全启动方法，包括：

在接收到用户触发的对于指定云主机的启动指令时，按照预设规则获取与所述指定云主机对应的资产信息，其中，所述资产信息至少包括操作系统信息；

判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞，其中，所述漏洞的数量为一个或多个；

若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间；

根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞；

若是，在预创建的安全隔离区内启动所述指定云主机，其中，所述指定云主机的网络访问功能在所述安全隔离区内会被禁用。

[权利要求 19] 根据权利要求18所述的计算机可读存储介质，其特征在于，所述按照预设规则获取与所述指定云主机对应的资产信息的步骤，包括：

确定所述指定云主机内是否安装有资产管理工具，其中，所述资产管理工具用于定期将所述指定云主机对应的资产信息同步至指定云平台；

若安装有资产管理工具，从所述指定云平台中查找出所述资产信息；

若未安装有资产管理工具，调用指定扫描工具对所述指定云主机进行扫描处理，以得到所述资产信息。

[权利要求 20] 根据权利要求18所述的计算机可读存储介质，其特征在于，所述判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞的步骤之前，所述方法还包括：

通过爬虫工具，实时从目标网站抓取出与云主机对应的各类操作系统相关的第一漏洞信息，以及与云主机服务相关的第二漏洞信息；

创建一个漏洞数据库；

将所述第一漏洞信息与所述第二漏洞信息存储于所述漏洞数据库内。

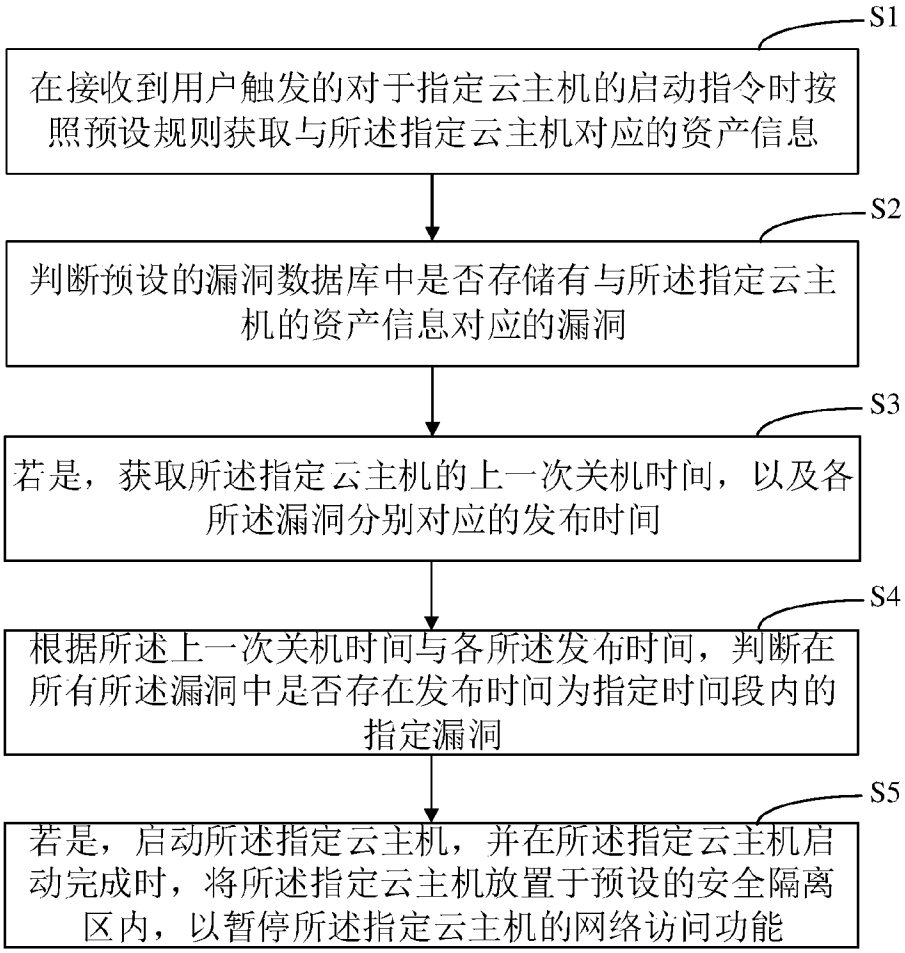

S1
在接收到用户触发的对于指定云主机的启动指令时按照预设规则获取与所述指定云主机对应的资产信息
S2
判断预设的漏洞数据库中是否存储有与所述指定云主机的资产信息对应的漏洞
S3
若是，获取所述指定云主机的上一次关机时间，以及各所述漏洞分别对应的发布时间
S4
根据所述上一次关机时间与各所述发布时间，判断在所有所述漏洞中是否存在发布时间为指定时间段内的指定漏洞
S5
若是，启动所述指定云主机，并在所述指定云主机启动完成时，将所述指定云主机放置于预设的安全隔离区内，以暂停所述指定云主机的网络访问功能

图 1

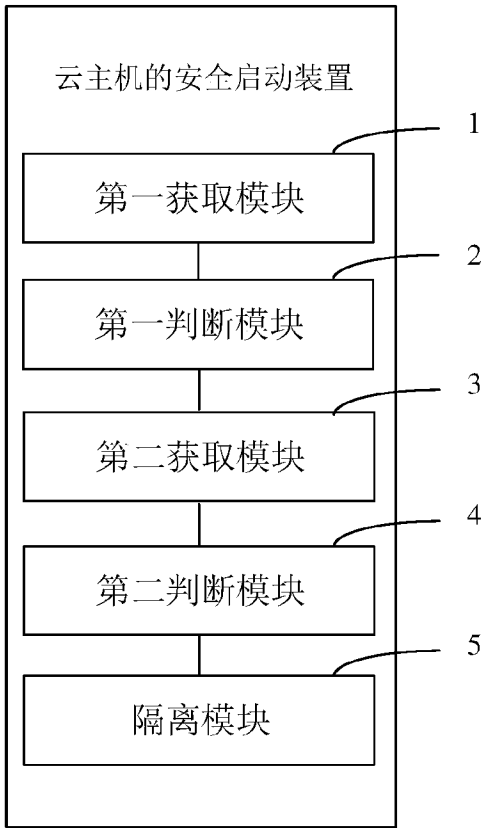

云主机的安全启动装置
1
第一获取模块
2
第一判断模块
3
第二获取模块
4
第二判断模块
5
隔离模块

图 2

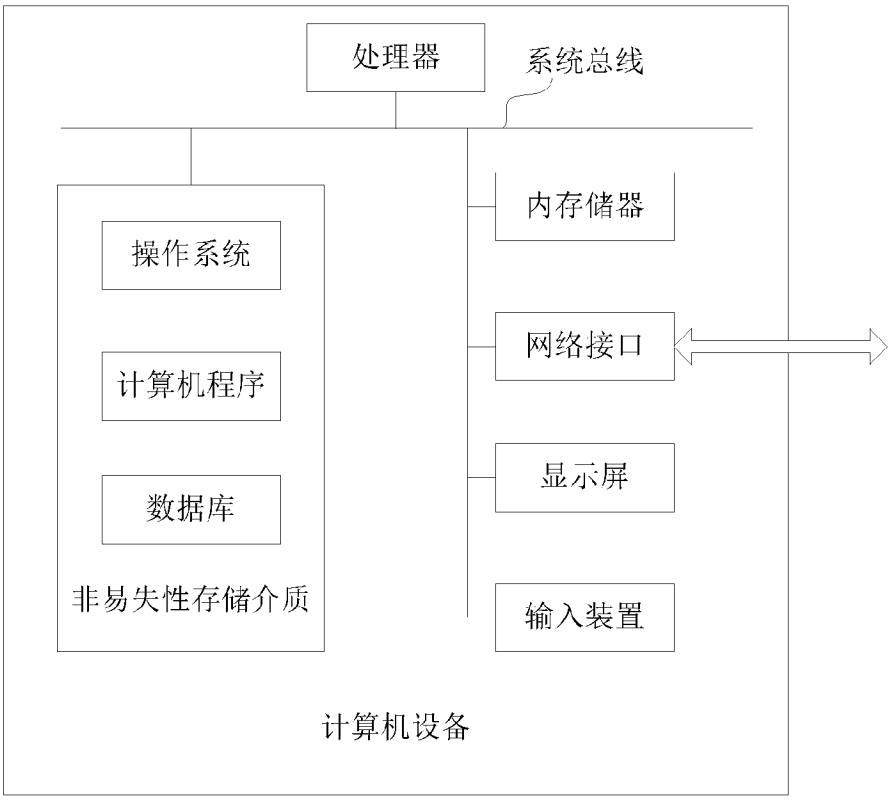

处理器
系统总线
内存储器
操作系统
网络接口
计算机程序
显示屏
数据库
非易失性存储介质
输入装置
计算机设备

图 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2019/118430

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/4401(2018.01)i; G06F 9/455(2006.01)i; G06F 8/61(2018.01)i; G06F 21/53(2013.01)i; G06F 21/57(2013.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS; CNTXT; CNKI; SIPOABS; DWPI; USTXT; WOTXT; EPTXT: 云, 主机, 启动, 漏洞, 虚拟机, 操作系统, 关机, 发布, 时间, 隔离区, 安全环境, 沙箱, 沙盒, cloud, startup, bug, VM, VZ, VPS, OS, shut, down, release, time, security, environment, sandbox, insulate, separate

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 105204902 A (HUAWEI TECHNOLOGIES CO., LTD.) 30 December 2015 (2015-12-30) description, paragraphs [0043]-[0129]	1-20
A	CN 110059007 A (BEIJING QIANXIN TECHNOLOGY CO., LTD.) 26 July 2019 (2019-07-26) entire document	1-20
A	US 2018053001 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 22 February 2018 (2018-02-22) entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 May 2020	**08 July 2020**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088** **China**	
Facsimile No. **(86-10)62019451**	Telephone No.

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2019/118430

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 105204902 A	30 December 2015	CN 105204902 B	07 December 2018
CN 110059007 A	26 July 2019	None	
US 2018053001 A1	22 February 2018	US 10460113 B2	29 October 2019

国际检索报告

国际申请号
PCT/CN2019/118430

A. 主题的分类

G06F 9/4401(2018.01)i; G06F 9/455(2006.01)i; G06F 8/61(2018.01)i; G06F 21/53(2013.01)i; G06F 21/57(2013.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

CNABS;CNTXT;CNKI;SIPOABS;DWPI;USTXT;WOTXT;EPTXT：云，主机，启动，漏洞，虚拟机，操作系统，关机，发布，时间，隔离区，安全环境，沙箱，沙盒，cloud，startup，bug，VM，VZ，VPS，OS，shut，down，release，time，security，environment，sandbox，insulate，separate

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
X	CN 105204902 A (华为技术有限公司) 2015年 12月 30日 (2015-12-30) 说明书第[0043]-[0129]段	1-20
A	CN 110059007 A (北京奇安信科技有限公司) 2019年 7月 26日 (2019-07-26) 全文	1-20
A	US 2018053001 A1 (IBM) 2018年 2月 22日 (2018-02-22) 全文	1-20

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2020年 5月 27日	2020年 7月 8日

ISA/CN的名称和邮寄地址	受权官员
中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088	潘秋羽
传真号 (86-10)62019451	电话号码 (86-512) 88995784

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2019/118430

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	105204902	A	2015年 12月 30日	CN	105204902	B	2018年 12月 7日
CN	110059007	A	2019年 7月 26日		无		
US	2018053001	A1	2018年 2月 22日	US	10460113	B2	2019年 10月 29日

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L1	10	((("SHUA") near3 ("Avi"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	ADJ	ON	ON	2022/10/13 10:09 AM
L2	6	L1 AND (analy$3 OR examin$3 OR examination OR inspect$3) WITH (asset OR resource) WITH (risk OR threat OR vulnerab$5 OR attack)	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	ADJ	ON	ON	2022/10/13 10:11 AM
L3	0	L1 AND (analy$3 OR examin$3 OR examination OR inspect$3) WITH (asset OR resource) WITH (risk OR threat OR vulnerab$5 OR attack) WITH back$1ward	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	ADJ	ON	ON	2022/10/13 10:12 AM
L4	1	((("Orca") near3 ("Security"))).AS,AANM.	(USPAT)	ADJ	ON	ON	2022/10/13 10:13 AM
L5	38	("0002014" OR "10152487" OR "10868825" OR "11036856" OR "11075923" OR "20130262801" OR "20160330237" OR "20180205755" OR "20180367515" OR "20190250857" OR "20190327272" OR "20200097662" OR "20200220887" OR "20200228560" OR "20200244692" OR "20200344256" OR "20200389483" OR "20200396254" OR "20210099478" OR "20210342297" OR "20210409438" OR "20220000021" OR "20220000191" OR "20220001241" OR "20220100869" OR "20220141247" OR "20220182294" OR "20220239681" OR "7013395" OR "7143143" OR "8272061" OR "9294498" OR	(US-PGPUB; USPAT)	ADJ	ON	ON	2022/10/13 10:14 AM

		"9356942" OR "9413774" OR "9569201" OR "9686296" OR "9749349" OR "9794289").pn.					
L6	2	("20190281115" OR "20200137097").pn.	(US-PGPUB; USPAT)	ADJ	ON	ON	2022/10/13 10:22 AM
L7	311485	(G06F9/45558 OR G06F9/5072 OR H04L63/1433 OR G06F2009/4557 OR G06F2009/45595 OR H04L63/14$2).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/13 10:45 AM
L8	1707	L7 AND (asset OR resource) WITH (risk OR threat OR vulnerab$5) WITH cloud	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/13 10:48 AM
L9	254	L7 AND (analy$3 OR examin$3 OR examination OR inspect$3) WITH (asset OR resource) WITH (risk OR threat OR vulnerab$5) WITH cloud	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/13 10:48 AM

L10	45	(Artificial Intelligence) Similar to: 17/657,981 with 0 CPC Selections and 0 Text Selections Text: (CN-106663022-A OR US-11099872-B2 OR US-20190073237-A1 OR US-10120706-B2 OR US-8621030-B2 OR US-20080080514-A1 OR US-9088590-B2 OR US-20140207906-A1 OR US-9342346-B2 OR EP-3177994-A1 OR WO-2016016695-A1 OR US-20160026489-A1 OR CN-104035868-A OR CN-104035868-B OR WO-2015188504-A1 OR CN-106802823-A OR US-20170083478-A1 OR US-10235341-B2 OR CN-106802823-B OR US-20150254095-A1 OR US-20160267051-A1 OR CN-107329801-A OR US-9417903-B2 OR CN-107329801-B OR CN-111538561-A OR US-20160239323-A1 OR EP-4002139-A3 OR CN-111857960-A OR CN-109416661-A OR US-9367370-B2 OR US-20160055041-A1 OR WO-2018087906-A1 OR US-10705879-B2 OR US-20200192654-A1 OR US-10740088-B2 OR CN-109416661-B OR CA-2753228-A1 OR WO-2010111149-A2 OR US-20100250868-A1 OR EP-2411915-A2 OR CA-2753228-C OR US-9529636-B2 OR RU-2011139151-A OR RU-2569805-C2 OR US-20180365138-A1 OR KR-101661783-B1 OR EP-2411915-B1 OR US-20190129700-A1 OR US-10628145-B2 OR US-20180121183-	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	ADJ	ON	ON	2022/10/13 10:49 AM

		A1).did.					
L11	46	(Artificial Intelligence) Similar to: 17/657,981 with 2 CPC Selections and 2 Text Selections Text: (US-20180270130-A1 OR US-20180365138-A1 OR US-9367370-B2 OR US-20160055041-A1 OR US-20190079896-A1 OR US-10657095-B2 OR US-20140207906-A1 OR US-9088590-B2 OR US-10120706-B2 OR US-11099872-B2 OR CN-111538561-A OR US-20150254095-A1 OR US-9495212-B2 OR US-20150339161-A1 OR US-20150026287-A1 OR US-9348649-B2 OR US-8856337-B2 OR WO-2015188504-A1 OR US-20130046892-A1 OR CN-106688208-A OR CN-106688208-B OR JP-2013041569-A OR US-20080080514-A1 OR US-8621030-B2 OR US-20190073237-A1 OR JP-5860320-B2 OR US-20160267051-A1 OR CN-112000606-A OR US-10621138-B2 OR CN-106663022-A OR KR-20100027936-A OR US-20160205215-A1 OR EP-3198806-B1 OR EP-3198806-A1 OR WO-2016045055-A1 OR US-20180225254-A1 OR WO-2010057775-A3 OR US-20180121183-A1 OR US-20190129700-A1 OR US-10628145-B2 OR US-9342346-B2 OR US-20170083478-A1 OR CN-101739113-A OR US-20160239323-A1 OR US-20140237607-A1 OR US-10235341-B2 OR EP-3177994-A1 OR	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN))	ADJ	ON	ON	2022/10/13 10:49 AM

		WO-2016016695-A1 OR US-20160026489-A1 OR US-9043624-B2).did.					
L12	0	17/657981.app.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/24 08:25 AM
L13	315067	(G06F9/45558 OR G06F9/5072 OR H04L63/1433 OR G06F2009/4557 OR G06F2009/45595 OR H04L63/14$2).cpc.	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/24 08:54 AM
L14	1977	L13 AND (risk OR threat OR vulnerab$5) WITH (identify$3 OR identification OR determin$3 OR determination) WITH cloud	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/24 08:55 AM
L15	383	L13 AND (risk OR threat OR vulnerab$5) WITH (identify$3 OR identification OR determin$3 OR	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA,	ADJ	ON	ON	2022/10/24 08:56 AM

		determination) WITH cloud WITH (resource OR asset)	EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L16	163	L13 AND (risk OR threat OR vulnerab$5) WITH (identify$3 OR identification OR determin$3 OR determination) WITH cloud WITH (resource OR asset) WITH network	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/24 08:59 AM
L17	848	L13 AND (forward OR backward) WITH (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:23 PM
L18	100	L13 AND (forward OR backward) near2 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm)	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO;	ADJ	ON	ON	2022/10/25 12:23 PM

			JPO; DERWENT; IBM_TDB)				
L19	0	L13 AND (forward OR backward) near2 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH vector	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:25 PM
L20	31	L13 AND (forward OR backward) WITH (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH vector	(US-PGPUB; USPAT; USOCR; FIT (AU, AP, AT, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, OA, RU, SU, WO, MC, MD, MY, NL, NO, NZ, PH, PL, PT, RO, RS, SE, SG, SI, SK, TH, TN, TR, TW, UA, VN); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:26 PM
L21	5	L13 AND (forward OR backward) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH vector	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:46 PM
L22	5	L13 AND (forward OR backward) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS,	ADJ	ON	ON	2022/10/25 12:47 PM

		OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) SAME vector	IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L23	5	L13 AND (forward OR backward) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) SAME (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH vector	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:47 PM
L24	71	(forward OR backward) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) SAME (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH vector	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:48 PM
L25	0	L13 AND (up$1stream OR down$1stream) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) SAME (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH vector	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:51 PM

L26	116	L13 AND (up$1stream OR down$1stream) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) SAME (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:51 PM
L27	83	L13 AND (up$1stream OR down$1stream) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:51 PM
L28	0	L13 AND (up$1stream) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH vector	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:56 PM
L29	0	L13 AND (down$1stream) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL,	ADJ	ON	ON	2022/10/25 12:58 PM

		OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) WITH lateral	NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L30	0	L13 AND (down$1stream) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) SAME lateral	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:58 PM
L31	0	L13 AND (down$1stream) near3 (analysis OR analyz$3 OR examin$3 OR examination OR evaluat$3 OR test$3) SAME (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) SAME lateral	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 12:58 PM
L32	3	L13 AND (up$1stream OR down$1stream OR forward OR backward) WITH path WITH entry WITH (display$3 OR visual$7)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 01:01 PM
L33	45	L13 AND (up$1stream OR down$1stream OR	(US-PGPUB; USPAT; USOCR; FIT (AP, AT,	ADJ	ON	ON	2022/10/25 01:20 PM

		forward OR backward) WITH path WITH (display$3 OR visual$7)	AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L35	1571	L13 AND (up$1stream OR down$1stream OR forward OR backward) WITH path AND cloud	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 01:24 PM
L36	130	L13 AND (up$1stream OR down$1stream OR forward OR backward) WITH path WITH (risk OR threat OR vulnerab$5 OR attack OR mal$1ware OR spy$1ware OR ad$1ware OR trojan OR virus OR worm) AND cloud	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	ADJ	ON	ON	2022/10/25 01:24 PM

PE2E SEARCH - Search History (Interference)

There are no Interference searches to show.

PE2E SEARCH - Search History (Similarity Search)

Ref #	Application Number	Text Selections	CPC Selections
L10	17/657,981	No selections	No selections

L11	17/657,981	(Specification) This summary is provided for the convenience of the reader to present a basic understanding of a few exemplary embodiments of this disclosure, and does not wholly define the breadth of the disclosure. This summary is not an extensive overview of all contemplated embodiments and is intended to neither identify key or critical elements of all embodiments nor to delineate the scope of any or all aspects. Its sole purpose is to present some features of one or more embodiments in a simplified form as a prelude to the more detailed description presented later. For convenience, the term "embodiments" may be used herein to refer to one or more features or combinations of features described herein. A description associated with an embodiment does not necessarily apply to all embodiments. [0014] Embodiments of the present disclosure may deliver instant-on, workload level visibility across cloud infrastructure assets without running opcodes on the target scanned workloads, helping organizations to: (1) Detect risks such as vulnerabilities, malware, misconfigurations, lateral movement risk, and unsecured sensitive data; (2) See cloud inventory at every layer infrastructure, operating system, applications, and data; (3) Discover and see previously missed assets. [0015] Embodiments of the present disclosure may further use an out-of-band process to reach cloud workloads through the runtime storage layer, combining this with information gathered from cloud provider APIs, including but not limited to K8S APIs. (Claims) A graphical user interface system for providing forward and backward facing attack vector visualizations, the system comprising:	(H04L63/1433) Vulnerability analysis (G06F2009/45595) Network integration; Enabling network access in virtual machine instances

		at least one processor configured to: identify assets in a cloud environment; identify risks associated with each of the identified asset; identify relationships between at least some of the identified assets, the relationships including at least one of a trust, a network connectivity, or a mechanism of network proxying; receive an identification of a specific asset under investigation; perform a forward analysis of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment; perform a backward analysis of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset; and output a signal to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points.	

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17657981
Filing Date	2022-04-05
First Named Inventor	Avi SHUA
Art Unit	
Examiner Name	Unassigned
Attorney Docket Number	15635.0008-00000

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20200137097	A1	2020-04-30	Gil ZIMMERMAN et al.	
	2	20190281115	A1	2019-09-12	Douglas SWALLOW et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17657981
Filing Date	2022-04-05
First Named Inventor	Avi SHUA
Art Unit	
Examiner Name	Unassigned
Attorney Docket Number	15635.0008-00000

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	International Search Report and Written Opinion for PCT International Application No. PCT/IB2022/52670 filed March 23, 2022; pgs. 1-14; mailed July 26, 2022.	

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/SARAH SU/	Date Considered	10/13/2022

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	17657981
Filing Date	2022-04-05
First Named Inventor	Avi SHUA
Art Unit	
Examiner Name	Unassigned
Attorney Docket Number	15635.0008-00000

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

See attached certification statement.

The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Christopher C. Johns/	Date (YYYY-MM-DD)	2022-08-11
Name/Print	Christopher C. Johns	Registration Number	68664

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.S/

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EFS Web 2.1.17

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.S/

Main Concept Text

==========

81. (Original) A graphical user interface system for providing forward and backward facing attack vector visualizations, the system comprising: at least one processor configured to: identify assets in a cloud environment; identify risks associated with each of the identified asset; identify relationships between at least some of the identified assets, the relationships including at least one of a trust, a network connectivity, or a mechanism of network proxying; receive an identification of a specific asset under investigation; perform a forward analysis of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment; perform a backward analysis of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset; and output a signal to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. 82. (Original) The system of claim 81, wherein the network outside the cloud environment includes the Internet. 83. (Original) The system of claim 81, wherein the assets in the cloud environment include at least one of: a virtual machine, a network appliance, a storage appliance, a compute instance, or an engine instance. 84. (Original) The system of claim 81, wherein identifying the assets in a cloud environment includes identifying the assets based on at least one of: an identity and access management policy, an organization policy, or an access policy. 85. (Original) The system of claim 81, wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream asset or a downstream asset. 86. (Original) The system of claim 85, wherein the visualization includes a presentation of the alternative paths. 87. (Original) The system of claim 81, wherein the presentation of the forward and backward paths indicates port numbers for each pathway. 88. (Original) The system of claim 81, wherein the visualization of the entry points indicates at least one entry point at risk. 89. (Original) The system of claim 81, wherein the at least one processor is further configured to monitor network activities of the assets in a cloud environment. 90. (Original) The system of claim 85, wherein the at least one processor is further configured to detect a potential risk associated with the specific asset based on the monitored network activities. 91. (Original) The system of claim 81, the at least one processor is further configured to detect a potential risk associated with the specific asset based on a network activity of the specific asset. 92. (Original) The system of claim 81, the at least one processor is further configured to detect a potential risk associated with the specific asset based on a network activity of an upstream asset of the specific asset. 93. (Original) The system of claim 81, the at least one processor is further configured to detect a potential risk associated with the specific asset based on a network activity of a downstream asset of the specific asset. 94. (Original) A method for generating a graphical user interface for providing forward and backward facing attack vector visualizations, the method comprising: identifying assets in a cloud environment; identifying risks associated

with each of the identified asset; identifying relationships between at least some of the identified assets, the relationships including at least one of a trust, a network connectivity, or a mechanism of network proxying; receiving an identification of a specific asset under investigation; performing a forward analysis of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment; performing a backward analysis of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset; and outputting a signal to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points. 95. (Original) The method of claim 94, wherein the network outside the cloud environment includes the Internet. 96. (Original) The method of claim 94, wherein the assets in the cloud environment include at least one of: a virtual machine, a network appliance, a storage appliance, a compute instances, or an engine instance. 97. (Original) The method of claim 94, wherein identifying the assets in a cloud environment includes identifying the assets based on at least one of: an identity and access management policy, an organization policy, or an access policy. 98. (Original) The method of claim 94, wherein the presentation of the forward and backward paths indicates alternative paths connecting between the specific asset and an upstream asset or a downstream asset. 99. (Original) The method of claim 98, wherein the visualization includes a presentation of the alternative paths. 100. (Original) 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: identifying assets in a cloud environment; identifying risks associated with each of the identified asset; identifying relationships between at least some of the identified assets, the relationships including at least one of a trust, a network connectivity, or a mechanism of network proxying; receiving an identification of a specific asset under investigation; performing a forward analysis of the specific asset under investigation to identify at least one possible attack vector reaching the specific asset via a network outside the cloud environment; performing a backward analysis of the specific asset to identify at least one exposure risk to one or more assets that is in a downstream of the specific asset, wherein the at least one exposure risk includes an identification of an exposed asset, an entry point to the exposed asset, and a lateral movement risk associated with the exposed asset; and outputting a signal to cause on a display to present a presentation of forward and backward paths associated with the specific asset, thereby enabling visualization of a plurality of entry points and lateral movement risks associated with the plurality of entry points.

(No Concept Modifiers)
==========

Filters (3)
==========
Publication Date: To 2021-04-26

Classic Syntax limited to: (attack or intrusion or virus or malware or adware or spyware or trojan) with vector

Classic Syntax limited to: (forward or backward) with (analysis or exam or examination)

Options
=======
De-dup: Simple Family Numbers (PAT only)
Cut-off: None
Sort: Relevance

Content: Patents + NPL (190)
============================
US Patents
US Designs
US Applications
EPO Patents
EPO Applications
China Patents
China Applications
Japan Patents
Japan Applications
Korea Patents
Korea Applications
WIPO Applications
Argentina Patents
Argentina Applications
Brazil Patents
Brazil Applications
Canada Patents
Canada Applications
Chile Patents
Chile Applications
Colombia Applications
Costa Rica Applications
Cuba Patents
Cuba Applications
Dominican Republic Applications
Ecuador Patents

Ecuador Applications
El Salvador Applications
Guatemala Applications
Honduras Applications
Mexico Patents
Mexico Applications
Nicaragua Patents
Panama Applications
Peru Applications
Trinidad & Tobago Patents
Uruguay Applications
Austria Patents
Austria Applications
Belarus Patents
Belgium Patents
Belgium Applications
Bosnia & Herzegovina Patents
Bosnia & Herzegovina Applications
Bulgaria Patents
Bulgaria Applications
Croatia Patents
Croatia Applications
Czech Republic Patents
Czech Republic Applications
Czechoslovakia Patents
Czechoslovakia Applications
Denmark Patents
Denmark Applications
Estonia Patents
Estonia Applications
EUIPO Patents
Finland Patents
Finland Applications
France Patents
France Applications
Germany Patents
Germany Applications
Great Britain Patents
Great Britain Applications
Greece Patents
Greece Applications
Hungary Patents
Hungary Applications
Iceland Patents
Iceland Applications
Ireland Patents
Ireland Applications
Italy Patents

Italy Applications
Latvia Patents
Latvia Applications
Lithuania Patents
Lithuania Applications
Luxembourg Patents
Malta Patents
Monaco Patents
Montenegro Patents
Montenegro Applications
Netherlands Patents
Netherlands Applications
Norway Patents
Norway Applications
Poland Patents
Poland Applications
Portugal Patents
Portugal Applications
Republic of Moldova Patents
Republic of Moldova Applications
Romania Patents
Romania Applications
San Marino Patents
San Marino Applications
Serbia Patents
Serbia Applications
Slovakia Patents
Slovakia Applications
Slovenia Patents
Spain Patents
Spain Applications
Sweden Patents
Sweden Applications
Switzerland Patents
Switzerland Applications
Ukraine Patents
Yugoslavia/Serbia and Montenegro Patents
Yugoslavia/Serbia and Montenegro Applications
Armenia Patents
Australia Patents
Australia Applications
Cyprus Patents
Gulf Cooperation Council Patents
Hong Kong Patents
India Patents
India Applications
Indonesia Patents
Indonesia Applications

Israel Applications
Jordan Patents
Jordan Applications
Kyrgyzstan Patents
Macao Applications
Malaysia Patents
Mongolia Patents
New Zealand Patents
Philippines Patents
Philippines Applications
Saudi Arabia Patents
Saudi Arabia Applications
Singapore Patents
Singapore Applications
Taiwan Patents
Taiwan Applications
Tajikistan Patents
Tajikistan Applications
Thailand Patents
Thailand Applications
Uzbekistan Patents
Vietnam Patents
Algeria Patents
ARIPO Patents
ARIPO Applications
Egypt Patents
Kenya Patents
Malawi Patents
Morocco Patents
Morocco Applications
OAPI Patents
South Africa Patents
Tunisia Applications
Zambia Patents
Zimbabwe Patents
EAPO Patents
EAPO Applications
Georgia Patents
Georgia Applications
Kazakhstan Patents
Kazakhstan Applications
Russia Patents
Russia Applications
Turkey Patents
Turkey Applications
IEEE Xplore Publications: IEEE Periodicals
IEEE Xplore Publications: IEEE Conferences
IEEE Xplore Publications: IEEE Standards

IEEE Xplore Publications: IEEE Early Access
IEEE Xplore Publications: SMPTE Periodicals
IEEE Xplore Publications: SMPTE Conferences
IEEE Xplore Publications: SMPTE Standards
IEEE Xplore Publications: MIT Press eBooks
IEEE Xplore Publications: Wiley-IEEE eBooks
IEEE Xplore Publications: IBM Periodicals
IEEE Xplore Publications: URSI Periodicals
IEEE Xplore Publications: VDE Conferences
IEEE Xplore Publications: Periodicals from China
IP.com Prior Art Database: The IP.com Journal
IP.com Prior Art Database: Internet Society RFC
IP.com Prior Art Database: IBM TDB Archive
IP.com Prior Art Database: Legacy Journals
IP.com Prior Art Database: Software Patent Institute
OnePetro.org: Periodicals at OnePetro.org
OnePetro.org: Conferences at OnePetro.org
Other Literature: IBM Redbooks
Other Literature: PubMed Central
Other Literature: arXiv.org

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use as many sheets as necessary)				Complete if Known	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	OPAP
				Examiner Name	CENTRAL, DOCKET
Sheet	1	of	3	Attorney Docket Number	**15635.0008-00000**

U.S. PATENTS

Examiner Initials*	Cite No.	Document Number Number-Kind Code (if known)	Issue Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-11,036,856B2	06-15-2021	Graun et al.	
		US-9,794,289B1	10-17-2017	Banerjee et al.	
		US-9,749,349B1	08-29-2017	Czarny et al.	
		US-9,569,201B1	02-14-2017	Lotem et al.	
		US-9,294,498B1	03-22-2016	Yampolskiy et al.	
		US-8,272,061B1	09-18-2012	Lotem et al.	
		US-9,356,942B1	05-31-2016	Joffe	
		US-9,413,774B1	08-09-2016	Liu et al.	
		US-11,075,923B1	07-27-2021	Srinivasan et al.	
		US-7,013,395B1	03-14-2006	Swiler et al.	
		US-10,868,825B1	12-15-2020	Dominessy et al.	
		US-10,152,487B1	12-11-2018	Patwardhan et al.	
		US-7,143,143B1	11-28-2006	Thompson	
		US-9,686,296B1	06-20-2017	Murchison et al.	

U.S. PUBLISHED PATENT APPLICATIONS

Examiner Initials*	Cite No.	Document Number Number-Kind Code (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-2020/0244692A1	07-30-2020	SHUA	
		US-2019/0327272A1	10-24-2019	NARAYANASWAMY	
		US-2020/0097662A1	03-26-2020	Hufsmith et al.	
		US-2013/0262801A1	10-03-2013	Sancheti et al.	
		US-2022/0210195A1	06-30-2022	Parekh et al.	
		US-2019/0250857A1	08-15-2019	Reddy et al.	
		US-2021/0409438A1	12-30-2021	Sundaram et al.	
		US-2020/0389483A1	12-10-2020	Dye et al.	
		US-2022/0141247A1	05-05-2022	Shakarian et al.	
		US-2020/0344256A1	10-29-2020	Alabdulhadi	
		US-2022/0191250A1	06-16-2022	Laliberte	
		US-2014/0366118A1	12-11-2014	Yin	
		US-2020/0220887A1	07-09-2020	A. K.	
		US-2018/0367515A1	12-20-2018	Anandam et al.	

Examiner Signature	/SARAH SU/	Date Considered	10/13/2022

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use as many sheets as necessary)				Complete if Known	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	OPAP
				Examiner Name	CENTRAL, DOCKET
Sheet	2	of	3	Attorney Docket Number	**15635.0008-00000**

U.S. PUBLISHED PATENT APPLICATIONS

Examiner Initials*	Cite No.	Document Number Number-Kind Code (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-2022/0182294A1	06-092022	Li et al.	
		US-2022/0100869A1	03-31-2022	BERGER et al.	
		US-2020/0396254A1	12-17-2020	Crabtree et al.	
		US-2016/0330237A1	11-10-2016	Edlabadkar	
		US-2018/0205755A1	07-19-2018	Kavi et al.	
		US-2021/0342297A1	11-04-2021	Gupta et al.	
		US-2022/0239681A1	07-28-2022	Newman et al.	
		US-2022/0124110A1	04-21-2022	CHHABRA et al.	
		US-2021/0099478A1	04-01-2021	Seetharamaiah et al.	
		US-2020/0228560A1	07-16-2020	Murthy et al.	

Note: Submission of copies of U.S. Patents and published U.S. Patent Applications is not required.

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.	Foreign Patent Document Country Code Number Kind Code (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	Translation
		WO 2020/106973A1	05-28-2020	ARAALI NETWORKS INC.		N/A
		WO 2019/051507A1	03-14-2019	CARBON BLACK INC.		N/A
		WO 2017/153983A1	09-14-2017	B. G. NEGEV TECHNOLOGIES AND APPLICATIONS LTD.		N/A

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	Translation
		Non-Final Office Action Application No. 17/716,491 dated August 3, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/657,986 dated August 17, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/658,156 dated July 18, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/716,583 dated August 18, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/657,972 dated June 21, 2022, in the United States Patent and Trademark Office.	N/A
		Final Office Action Application No. 17/657,972 dated September 1, 2022, in the United States Patent and Trademark Office.	N/A
		Ex Parte Quayle Action Application No. 17/716,231 dated June 26, 2022, in the United States Patent and Trademark Office.	N/A

Examiner Signature	/SARAH SU/	Date Considered	10/13/2022

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use as many sheets as necessary)				Complete if Known	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	OPAP
				Examiner Name	CENTRAL, DOCKET
Sheet	3	of	3	Attorney Docket Number	**15635.0008-00000**

Examiner Signature	/SARAH SU/	Date Considered	10/13/2022

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /S.S/

Bibliographic Data

Application No: 17/657,981

Foreign Priority claimed: ○ Yes ◉ No

35 USC 119 (a-d) conditions met: ☐ Yes ☑ No ☐ Met After Allowance

Verified and Acknowledged: /SARAH SU/ Examiner's Signature | Initials

Title: Forward and Rearward Facing Attack Vector Visualization

---

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
04/05/2022	726	2431	15635.0008-00000
RULE			

APPLICANTS

Orca Security, Tel Aviv-Yafo, ISRAEL

INVENTORS

Avi SHUA, Tel Aviv-Yafo, ISRAEL

CONTINUING DATA

This application is a CON of PCT/IB2022/052670 03/23/2022

PCT/IB2022/052670 has PRO of 63180048 04/26/2021

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

04/12/2022

STATE OR COUNTRY

ISRAEL

ADDRESS

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413
UNITED STATES

FILING FEE RECEIVED

$6,160

identification method, device and system of asset vulnerability

DOCUMENT ID
CN-110708315-A

DATE PUBLISHED
2020-01-17

INVENTOR INFORMATION

NAME	CITY	STATE	ZIP CODE	COUNTRY
WANG, YING-XIN	N/A	N/A	N/A	N/A
FAN, Yuan	N/A	N/A	N/A	N/A

DATE FILED
2019-10-09

Abstract

The invention claims a method of asset vulnerability identification method, device and system, referring to the technical field of network security, comprising: obtaining the asset information of the detected asset, the asset information is sent to the cloud data center. to make the cloud data centre based on asset information asset to be detected for detecting vulnerability identification, and generating a vulnerability identification result, wherein the cloud data center stores target threat intelligence data, target threat intelligence data comprises at least one of the following: the security service provider upload of threat intelligence data. threat intelligence data uploaded by the security information organization and the threat information data uploaded by the enterprise detection asset belongs to, obtaining the vulnerability identification result, and generating the pre-warning information based on vulnerability identification result, solves the technical problem of detecting the asset vulnerability of existing asset vulnerability identification method is single.

Description

INVENTION-TITLE
technical field
This invention claims network security technology field, especially refers to a kind of asset vulnerability identification method, device and system.
BACKGROUND
under the current network safety technology condition, the occurrence of a network attack vulnerability basic network threat using information assets exposed by malicious action is carried out. in the network space, network threat is objectively present, to reduce the risk of the network attack, vulnerability reinforcing the information asset is a very effective security technical measures. but before the asset vulnerability reinforcing, needs to know the vulnerability information asset with which, therefore, information asset vulnerability detection is abnormal important for network security protection.
However, the existing vulnerability detection method has the technical problems is single vulnerability detection to the asset.
Aiming at said problem, yet provides effective solution.
summary of the invention
Therefore, it is an object of the present invention is to provide a kind of asset vulnerability identification method, device and system, so as to solve the technical problem of the asset vulnerability detecting existing

of asset vulnerability identification method is single.
In a first aspect, the invention provides a method for identifying asset vulnerability, comprising: obtaining the asset information of the detected asset, said asset information to the cloud data center. to make the cloud data centre for detecting vulnerability identification based on said asset information for the asset to be detected, and generating vulnerability identification result, wherein the cloud data center stores a target threat intelligence data, the target threat intelligence data comprises at least one of the following: the security service provider upload of threat intelligence data. threat intelligence data uploaded by the security information organization and the threat information data uploaded by the enterprise detection asset belongs to, obtaining the vulnerability identification result and generating the pre-warning information based on the vulnerability identification result.
Further, obtaining the asset information of asset detection, comprising: sending the instruction to the local asset information base so that the local asset information database based on the obtaining instruction collecting attribute information and fingerprint information of the asset to be detected, wherein officer name of the asset to be detected belongs to said fingerprint information include, but are not limited to, the following information: the classification information of the to-be-detected asset factory manufacturer name, the to-be-detected asset, the asset to be detected operating system information; obtaining the attribute information and the fingerprint information, the attribute information and the fingerprint information is determined as the asset information, wherein the classification information comprises: first level classification information and second level classification information, first level classification comprises the following types: server, a network device, a security device, a secondary classification comprises the following types: a firewall, a switch, a router and a database.
Further, if the two-stage classification of the asset to be detected is a database, then the fingerprint information further comprises the type and version of the asset to be detected.
Further, the asset information is sent to the cloud data center to make the cloud data centre based on said asset information for the asset to be detected for detecting vulnerability identification, comprising: the cloud service port of said asset information provided by the cloud data centre sending to the cloud data center, so that the cloud data centre based on preset leak detection program and the asset information for the asset to be detected for detecting vulnerability identification.
Further, the cloud data centre for providing infrastructure as a service and platform as a service, the cloud data centre stores the target threat intelligence data based on Hadoop framework.
In a second aspect, the invention provides a kind of asset vulnerability identification device, comprising: obtaining unit, sending unit and a warning unit, wherein the obtaining unit is used for obtaining the asset information of the detected asset; the sending unit is used for sending the asset information to the cloud data center, so that the cloud data centre performing vulnerability recognition detection to the to-be-detected asset based on the asset information, and generating a vulnerability identification result, wherein the cloud data center stores a target threat intelligence data. the target threat intelligence data comprises at least one of the following: the security service provider upload of threat intelligence data, threat information data uploaded by the security information organization and the threat information data uploaded by the enterprise detection asset belongs to; the warning unit for obtaining the vulnerability identification result, and generating the pre-warning information based on the vulnerability identification result.
Further, the obtaining unit is further used for sending the instruction to the local asset information base so that the local asset information database based on the obtaining instruction collecting attribute information and fingerprint information of the asset to be detected, wherein officer name of the asset to be detected belongs to said fingerprint information include, but are not limited to, the following information: the classification information of the to-be-detected asset factory manufacturer name, the to-be-detected asset, the asset to be detected of operating system information, obtaining the attribute information and the fingerprint information; the attribution information and the fingerprint information is determined as the asset information, wherein the classification information comprises: first level classification information and second level classification information, first level classification comprises the following types: server, a network device, a security device, a secondary classification comprises the following types: a firewall, a switch, a router and a database.
Further, if the two-stage classification of the asset to be detected is a database, then the fingerprint information further comprises the type and version of the asset to be detected.

Further, the sending unit is further used for: the cloud service port of said asset information provided by the cloud data center is sent to the cloud data center, so that the cloud data centre based on preset leak detection program and the asset information for the asset to be detected for detecting vulnerability identification.

In a third aspect, the invention claims a kind of asset vulnerability identification system, comprising: a production information collecting device, a cloud data center and leak warning platform, wherein, the local asset information base for collecting the asset information of asset to be detected; the cloud data centre is used for performing vulnerability recognition detection on the to-be-detected asset based on the asset information, and generating the vulnerability identification result, the vulnerability warning platform for generating the pre-warning information based on the vulnerability identification result.

In the embodiment of the invention, firstly, obtaining the to-be-detected asset information of the asset, the asset information is sent to the cloud data center, to make the cloud data centre based on asset information asset to be detected for detecting vulnerability identification, and generating vulnerability identification result, wherein cloud data center stores target threat intelligence data, target threat intelligence data comprises at least one of the following: the security service provider upload of threat intelligence data, security threat information data uploaded by the information organization and the threat information data uploaded by the enterprise detection asset belongs, vulnerability identification result for characterizing the asset to be detected whether there is vulnerability, obtaining vulnerability identification result, and generating the pre-warning information based on vulnerability identification result.

In the invention, the asset information of the asset to be detected to the obtained sent to the cloud data center to make the cloud data centre according to the target the stored threat intelligence data, assets to be detected of the comprehensive vulnerability detection. Because the cloud data centre according to the collected vulnerability multi-party provides the threat intelligence data asset to be detected for leak detection, the asset to be detected for overall vulnerability detection, so as to solve the technical problem of detecting of the asset vulnerability in the existing technology is single. so as to realize the asset to be detected to the technical effect of the comprehensive vulnerability detection.

Other features and advantages of the present invention will be described in the following specification, and partly become apparent from the specification, or by implementing the present invention understood. structure for purpose of this invention and the other advantages particularly pointed out in the specification, claims and drawings realized and obtained.

To make the above object, features and advantages of the present invention can be best understood, lower Went preferred embodiments, and the accompanying drawings, to the following detailed description.

Description

In order to clearly illustrate the technical solution of the invention specifically implementation manner or in the existing technology, will now be simply described drawings required to be used in the specific embodiment or existing technology description, obviously, the drawings in the following description are some embodiments of the present invention, for this field common technician, under the premise of without paying creative labor, it also can be other drawings according to these drawings to obtain.

FIG. 1 an embodiment of the present invention provides a method for identifying asset vulnerability of FIG.

FIG. 2 an embodiment of the present invention to provide an asset vulnerability identification device of the present invention;

FIG. 3 an embodiment of the present invention to provide an asset vulnerability identification system of the present invention;

FIG. 4 is a schematic diagram of a server according to an embodiment of the invention.

Specific implementation methods

In order to make the purpose of the invention, technology scheme advantages and more clearly, Following combine the accompanying drawings to the technical solution of the invention will be described in a clearly and fully, obviously, the embodiment description of the invention embodiment part, but not all of the embodiments. Based on the embodiments of the invention, all other embodiments obtained by ordinary technician in present technical field without any creative work is within the protective scope of the present invention.

about a vulnerability detection technology. the current vulnerability detection technology of the information asset by actively detecting technology mainly relies on special vulnerability scanning device to achieve, in

this way, the vulnerability scanning device using which carries the "vulnerability rule library" asset to actively scan the various kinds of information, such as when to scan for a certain asset, the rule R if the scanning result hits the "vulnerability rule library", it is determined that the asset has vulnerability R vulnerability corresponding to the rule. result in such a case, the vulnerability scanning depends on the quantity and quality of "vulnerability rule library" carried by the vulnerability scanning device. if a certain a certain vulnerability information exposed out of the asset vulnerability in the "rule library", then the leak can be detected by the vulnerability scanning device. if a certain vulnerability not leak the "rule library", then the vulnerability scanning device cannot be detected, the vulnerability exists even if true this vulnerability.
to "vulnerability rule base" of the update. For the discovery of vulnerability model, vulnerability scanning service provider will generally periodically or aperiodically the vulnerability are packaged into software upgrading packet, vulnerability scanning device by the upgrade of the online or offline to finish upgrading of vulnerability rule library ", of vulnerability scanning device side after finishing the upgrading can have the newly discovered vulnerability detection capability.
identification method of the existing asset vulnerability has the following disadvantages:
1 based on the vulnerability scanning technology of active scanning needs to send data to the scanned target by a network packet, due to the complexity of the real environment, which may cause operation of the target host is not stable, causing downtime of the information asset, stopping service, data loss and so on.
2, the hysteresis of the vulnerability detection. For the Oday vulnerability and vulnerability, the vulnerability scanning device are not disclosed because of hysteresis vulnerability rule library " upgrade, may cause some vulnerability cannot be detected, so there is the risk the malicious threat using the vulnerability to attack.
3, single detection capability. Generally, the leak detection device of a manufacturer has its inherent technical and limit of vulnerability rule library, cannot exert a plurality of security vendor and vulnerability detection capability of various security organization, cannot realize the polymerization of vulnerability detection capability.
4, the efficiency of vulnerability detection is low. The traditional vulnerability scanning equipment technology implementation reasons, a plurality of asset for vulnerability detection time, low efficiency, typically complete vulnerability detection of one asset requires several minutes.
Aiming at the above disadvantages, the invention adopts the following method to overcome said disadvantages:
First Embodiment
according to an embodiment of the present invention, there is provided a method for identifying asset vulnerability of embodiment, it should be noted that the illustration of drawings in the step flow of execution in computer system can be such as a group of computer-executable instructions, and although the logical order is shown in the flow chart, but in some cases, steps may be executed in different order shown or described.
FIG. 1 according to an embodiment of the present invention a method for identifying asset vulnerability, as shown in FIG. 1, the method comprises the following steps:
step S102, obtaining the asset information of the detected asset;
step S104, the asset information is sent to the cloud data center to make the cloud data centre performing vulnerability recognition detection to the to-be-detected asset based on the asset information, and generating a vulnerability identification result, wherein the cloud data center stores a target threat intelligence data, the target threat intelligence data comprises at least one of the following: the security service provider upload of threat intelligence data, threat information data uploaded by the security information organization and the threat information data uploaded by the enterprise detection asset belongs to;
step S106, obtaining the vulnerability identification result, and generating the pre-warning information based on the vulnerability identification result.
In the invention, the asset information of the asset to be detected to the obtained sent to the cloud data center to make the cloud data centre according to the target the stored threat intelligence data, assets to be detected of the comprehensive vulnerability detection. Because the cloud data centre according to the collected vulnerability multi-party provides the threat intelligence data asset to be detected for leak detection, the asset to be detected for overall vulnerability detection, so as to solve the technical problem of detecting of the asset vulnerability in the existing technology is single. so as to realize the asset to be detected to the

technical effect of the comprehensive vulnerability detection.
It should be noted that, the capability data by establishing threat intelligence data uploaded by the security service provider, a security threat intelligence data uploaded by the information organization and the cloud data centre threat intelligence data uploaded by the enterprise detection asset belongs, may be incorporated consisting of and abutting each party security threat intelligence data, form a threat intelligence synergy and multi-security information of the synergy of security threat information sharing mechanism construction.
In addition, it should be noted that, the vulnerability warning information can be sent to the worker through short message, mail and so on, so that the worker can timely processing the asset vulnerability to be detected, so as to reduce the risk to be detected asset faces.
In the embodiment of the invention, the step S102 further comprises the following steps:
step S11, sending the instruction to the local asset information base so that the local asset information library collects attribute information and fingerprint information of the asset to be detected based on the acquired instruction, name of responsible person, the asset to be detected belongs, the fingerprint information include, but are not limited to, the following information: the classification information of the to-be-detected asset factory manufacturer name, the to-be-detected asset, the asset to be detected operating system information;
step S11, obtaining the attribute information and the fingerprint information, the attribute information and the fingerprint information is determined as the asset information.
In the embodiment of the invention, after sending the instruction to the local asset information base, local asset information library according to the obtaining instruction collecting the attribute information and fingerprint information of the detected assets.
Then, the obtained local asset information base collected after detecting the attribute information and fingerprint information of the asset as the asset information of the detected asset.
It should be noted that the attribute information include, but are not limited to the following information: name of the asset to be detected, after detecting the asset belongs to responsible person name and other information.
said fingerprint information include, but are not limited to, the following information: operating system information detection asset to be factory manufacturer name, the classification information of the detected asset, asset to be detected, wherein the classification information comprises: first level classification information and second level classification information, first level classification comprises the following types: server, a network device, a security device, a secondary classification comprises the following types: a firewall, a switch, a router and a database.
In addition, it should be noted that operating system to be detected comprises: type of information assets to the operating system, the version of the operating system and other information.
if the asset is the database to be detected, then the version and type of fingerprint information of the detected asset comprises a cash asset.
in the asset information of the asset to be detected comprises fingerprint information, so that each of the fingerprint information to determine one implementation asset to be detected, so as to accurately by asset fingerprint matching technology assets to be detected for the passive vulnerability detection accurately, reduce the vulnerability detection when running effect of assets to be detected.
In addition, local asset information database in the to-be-detected asset information of asset collection can be performed through automatic or manual input, it automatically collects the asset information mainly by asset detection device finishes, manual input, the local asset information library can provide visualization of the input interface for the worker to the asset information manually recording local asset information in the database.
In the embodiment of the invention, the step S104 further comprises the following steps:
step S21, the cloud service port of said asset information provided by the cloud data centre sends to the cloud data center, so that the cloud data centre inlet vulnerability identification detection based on preset leak detection program and the asset information for the asset to be detected.
In the embodiment of the invention, the threat intelligence database provided by the cloud data centre IaaS (infrastructure as a service) and relatedness (platform as a service) as a base layer, using Hadoop framework of building based on the large data processing architecture, the threat of data source information in the database may be from various security service providers, security intelligence organization, enterprise

itself. data in the PaaS (platform as a service) constructing leak detection cloud application program (i.e., the predetermined leak detection program), the program can call the threat intelligence database in order for assets to be detected for detecting vulnerability identification. at the same time, the program provided by via the Internet cloud service interface to realize the service of vulnerability detection interface.
Because the cloud data centre can by real-time update of threat information cloud data as basic data, by means of cloud interface provides leak detection service, it can real time provide fast and in time of the detection result is 0day vulnerability detection and position detection of the asset, solve the vulnerability detection service due to the hysteresis caused by the risk vulnerability rule library, and cloud data center to cloud interface provided by way of can batch the asset vulnerability detection and improves the detection efficiency of the asset vulnerability.
Second Embodiment
The invention further claims an asset vulnerability identification device, the device identification method for performing asset vulnerability of embodiment of the above content of the present invention, specifically claims the following device is identifying asset vulnerability provided by the embodiment of the invention.
As shown in FIG. 2, the asset identification device comprises an obtaining unit 10, a transmitting unit 20 and a warning unit 30.
the obtaining unit 10 for obtaining the asset information of asset to be detected;
the sending unit 20 for sending the asset information to the cloud data center, so that the cloud data centre performing vulnerability recognition detection to the to-be-detected asset based on the asset information, and generating a vulnerability identification result, wherein the cloud data center stores a target threat intelligence data, the target threat intelligence data comprises at least one of the following: the security service provider upload of threat intelligence data, threat information data uploaded by the security information organization and the threat information data uploaded by the enterprise detection asset belongs to;
the warning unit 30 for obtaining the vulnerability identification result, and generating the pre-warning information based on the vulnerability identification result.
In the invention, the asset information of the asset to be detected to the obtained sent to the cloud data center to make the cloud data centre according to the target the stored threat intelligence data, assets to be detected of the comprehensive vulnerability detection. Because the cloud data centre according to the collected vulnerability multi-party provides the threat intelligence data asset to be detected for leak detection, the asset to be detected for overall vulnerability detection, so as to solve the technical problem of detecting of the asset vulnerability in the existing technology is single. so as to realize the asset to be detected to the technical effect of the comprehensive vulnerability detection.
Preferably, the obtaining unit is further used for sending the instruction to the local asset information base so that the local asset information database based on the obtaining instruction collecting attribute information and fingerprint information of the asset to be detected, wherein officer name of the asset to be detected belongs to said fingerprint information include, but are not limited to, the following information: the classification information of the to-be-detected asset factory manufacturer name, the to-be-detected asset, the asset to be detected of operating system information, obtaining the attribute information and the fingerprint information; the attribution information and the fingerprint information is determined as the asset information, wherein the classification information comprises: first level classification information and second level classification information, first level classification comprises the following types: server, a network device, a security device, a secondary classification comprises the following types: a firewall, a switch, a router and a database.
Preferably, if two-stage classification of the asset to be detected is a database, then the fingerprint information further comprises the type and version of the asset to be detected.
Preferably, the sending unit is further used for: the cloud service port of said asset information provided by the cloud data center is sent to the cloud data center, so that the cloud data centre based on preset leak detection program and the asset information for the asset to be detected for detecting vulnerability identification.
Third Embodiment
The invention further claims an asset vulnerability identification system, the device identification system for performing asset vulnerability of embodiment of the above content of the present invention, the following are

specifically described identification system of asset vulnerabilities according to an embodiment of the invention.
As shown in FIG. 3, the asset identification system comprising: a local asset information library 100, the cloud data center 200 and leak warning platform 300.
the local asset information database 100 for collecting and storing the to-be-detected asset information of asset;
the cloud data center 200 for performing vulnerability recognition detection based on said asset information for the asset to be detected, and generating a vulnerability identification result;
asset information the vulnerability warning platform 300 for obtaining the local asset information stored in the asset to be detected, and sending the asset information to the cloud data centre, and generating the pre-warning information based on the vulnerability identification result.
In the invention, the asset information of the asset to be detected to the obtained sent to the cloud data center to make the cloud data centre according to the target the stored threat intelligence data, assets to be detected of the comprehensive vulnerability detection. Because the cloud data centre according to the collected vulnerability multi-party provides the threat intelligence data asset to be detected for leak detection, the asset to be detected for overall vulnerability detection, so as to solve the technical problem of detecting of the asset vulnerability in the existing technology is single. so as to realize the asset to be detected to the technical effect of the comprehensive vulnerability detection.
Referring to FIG. 4, an embodiment of the invention further claims a server 400, including a processor 50, a memory 51, a bus 52, and a communication interface 53, the processor 50, communication interface 53, and memory 51 connected via bus 52, a processor 50 for executing module to execute, such as a computer program stored in the memory 51.
wherein, the memory 51 may include high speed random access memory (RAM, Random Access Memory), and may also include memory (non volatile memory) of non-stable -, such as at least one disk memory. via the at least one communication interface 53 (which may be wired or wireless) to realize the system network and the communication connection between the at least one other network element, it can use the internet, wide area network, local area network, metropolitan area network, and the like.
Bus 52 can be an ISA bus, PCI bus or an EISA bus. the bus can be divided into address bus, data bus, control bus, and the like. For ease of representation, FIG. 4 is only represented by a double arrow, but does not indicate that only one bus or one type of bus.
wherein, a memory 51 for storing a program, the processor 50 after receiving the execution instruction, executes the program, the method performed by the device of any embodiment of the invention embodiment claims a flow process definition of can be used for processor 50, or implemented by the processor 50.
Processor 50 may be an integrated circuit chip with a processing capability of the signal. finishing instruction of the integrated logic circuit in the realizing process, each step of the above method can by processor 50 in hardware or software form. the processor 50 may be a general processor, comprising a central processing unit (Central Processing Unit (CPU), network processor (Network Processor, abbreviated NP), but also can be digital signal processor (Digital SignalProcessing (DSP), an application specific integrated circuit (Application Specific Integrated Circuit, abbreviated as ASIC), programmable gate array (Field-Programmable Gate Array (FPGA) or other programmable logic device, discrete gate or transistor logic device, discrete hardware components. it can realize or perform each method disclosed in an embodiment of the invention, and the steps and logic diagram. a general purpose processor may be a microprocessor or the processor can be any conventional processor and so on. The embodiment of the invention claims a step of combining method may be directly embodied as hardware decoding processor execution is completed, or for combination in the decoding processor hardware and software module execution is completed. software module may be located in random access memory, flash memory, read only memory, programmable read only memory or an electrically erasable programmable memory, registers, and so the mature storage medium. the storage medium is located in memory 51, steps the processor 50 reads information in memory 51, combines its hardware to complete the method.
In addition, in the description of the embodiment of the invention, unless otherwise explicitly specified and defined, the terms "mounted", "connected", "connection" should be understood broadly, for example, may be a fixed connection, also can be detachably connected or integrally connected, can be mechanically connected, also can be electric connected can be directly connected, or may be indirectly connected by an

intermediate medium, can be connected in the two elements. For those of ordinary skill in the art, can understand the specific meaning of the term in the present invention the specific condition.
In the description of the invention, it should be noted that the term "central", "upper", "lower", "left", "right", "vertical", "horizontal", "inner", orientation or positional relationship of "outside") indication is based on orientation or positional relationship shown in the drawings is only to facilitate description of the present invention and to simplify the description, but not indicate or imply the apparatus or element must have a specific orientation in a specific orientation structure and operation, so it can not be understood as limiting the present invention. In addition, the terms "first", "second", "third" used for description purposes only, but are not intended to indicate or imply relative importance.
In the several embodiments provided in the present application, it should be understood that the system, apparatus and method disclosed, can be realized by other ways. apparatus embodiments described above are schematic only, for example, the division of the unit, is only a logical function division, the actual implementation can have additional dividing mode of, for example, multiple units or components can be combined or integrated into another system, or some features may be ignored or not performed. the other point, the displayed or discussed mutual coupling between or direct coupling or communication connection may be via some communication interface, indirect coupling or communication connection of the apparatus or unit can be electrically, mechanically or other form.
as the separation member unit can be or may not be physically separated, as unit display module may be or may not be physical units, that is, may be located in one position, or may be distributed to multiple network units. can be adjusted according to the actual need to select part or all of unit wherein to realize the purpose of the embodiment.
In addition, each function unit in each embodiment of the present invention can be integrated in a processing unit, or may be separate physical presence of each unit, also can be two or more than two unit are integrated in one unit.
at last, it should be noted that the above embodiment is only the specific embodiments of the invention, used to illustrate the technical solution of the invention, and not to its limit, the protection scope of the present invention is not limited to this. Although reference to the foregoing embodiment of the invention proceed the detailed explanation, one of ordinary skill in the art should understand that the technical solution in the technical scope any familiar with the technical field of technician in this invention claims, it still can be the preceding embodiment according to modifying or can easily think of changes. or wherein the part technical feature for equivalent replacement, and these modifications, changes or substitutions do not make the essence of the corresponding technical solutions departing from the spirit and scope of the technical solution of the embodiment of the invention, should be covered in the protection scope of the present invention. Therefore, the protection range of the invention should be compatible with the protection scope.

Claims

1. An asset vulnerability identification method, wherein, applied to vulnerability warning platform, comprising: obtaining the asset information of the detected asset, said asset information to the cloud data center. to make the cloud data centre for detecting vulnerability identification based on said asset information for the asset to be detected, and generating vulnerability identification result, wherein the cloud data center stores a target threat intelligence data, the target threat intelligence data comprises at least one of the following: the security service provider upload of threat intelligence data. threat intelligence data uploaded by the security information organization and the threat information data uploaded by the enterprise detection asset belongs to, obtaining the vulnerability identification result and generating the pre-warning information based on the vulnerability identification result.
2. The method according to claim 1, wherein obtaining the asset information of asset detection, comprising: sending the instruction to the local asset information base, to make the local asset information database based on the obtaining instruction collecting attribute information and fingerprint information of the asset to be detected, wherein the attribute information includes, but is not limited to the following information: name of the asset to be detected, the asset to be detected of the responsible person name belongs to said fingerprint information include, but are not limited to, the following information: factory name of the

manufacturer of the asset to be detected, classification information of the asset to be detected, operation system information of the asset to be detected, obtaining the attribute information and the fingerprint information; the attribution information and the fingerprint information is determined as the asset information, wherein the classification information comprises: first level classification information and second level classification information, first level classification comprises the following types: server, network equipment, safety equipment, the secondary classification comprises the following types: firewall, switch, router, database.

3. The method according to claim 2, wherein, if the two-stage classification of the asset to be detected is a database, then the fingerprint information further comprises the type and version of the asset to be detected.

4. The method according to claim 1, wherein the asset information to the cloud data center to make the cloud data centre based on said asset information for the asset to be detected for detecting vulnerability identification, comprising: the cloud service port of said asset information provided by the cloud data centre sending to the cloud data center, so that the cloud data centre inlet vulnerability identification detection based on preset leak detection program and the asset information for the asset to be detected.

5. The method according to claim 1, wherein the cloud data centre for providing infrastructure as a service and platform as a service, the cloud data centre stores the target threat intelligence data based on Hadoop framework.

6. An asset vulnerability identification device, wherein the device comprises an obtaining unit, sending unit and a warning unit, wherein the obtaining unit is used for obtaining the asset information of the detected asset; the sending unit is used for sending the asset information to the cloud data center, so that the cloud data centre performing vulnerability recognition detection to the to-be-detected asset based on the asset information, and generating a vulnerability identification result, wherein the cloud data center stores a target threat intelligence data. the target threat intelligence data threat data comprises at least one of the following: the security service provider upload of threat intelligence data, threat information data uploaded by the security information organization and the threat information data uploaded by the enterprise detection asset belongs to; the warning unit for obtaining the vulnerability identification result, and generating the pre-warning information based on the vulnerability identification result.

7. The device according to claim 6, wherein the obtaining unit is further used for sending the instruction to the local asset information base. to make the local asset information database collecting attribute information and fingerprint information of the asset to be detected based on the obtaining instruction, name of responsible person, the asset to be detected belongs, said fingerprint information include, but are not limited to, the following information: the classification information of the to-be-detected asset factory manufacturer name, the to-be-detected asset, the asset to be detected operating system information; obtaining the attribute information and the fingerprint information, the attribute information and the fingerprint information is determined as the asset information, wherein the classification information comprises: first level classification information and second level classification information, first level classification comprises the following types: server, a network device, a security device, a secondary classification comprises the following types: a firewall, a switch, a router and a database.

8. The device according to claim 7, wherein, if the two-stage classification of the asset to be detected is a database, then the fingerprint information further comprises the type and version of the asset to be detected.

9. The device according to claim 6, wherein said sending unit is further used for: the cloud service port of said asset information provided by the cloud data center is sent to the cloud data center, so that the cloud data centre based on preset leak detection program and the asset information for the asset to be detected for detecting vulnerability identification.

10. An asset vulnerability identification system, wherein the system comprises a local asset information database, a cloud data center and leak warning platform, wherein, the local asset information database for collecting and storing the to-be-detected asset information of asset; the cloud data centre for performing vulnerability recognition detection on the to-be-detected asset based on the asset information, and generating a vulnerability identification result, the vulnerability warning platform for asset information for obtaining the local asset information stored in the asset to be detected, and sending said asset information to the cloud data centre, and generating the pre-warning information based on the vulnerability identification result.

Copyright © 2022 Clarivate Analytics. All rights reserved. Republication or redistribution of Clarivate Analytics content, including by framing or similar means, is prohibited without the prior written consent of Clarivate Analytics. Clarivate and its logo are trademarks of the Clarivate Analytics group.

(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110708315 A
(43)申请公布日 2020.01.17

(21)申请号 201910956125.4

(22)申请日 2019.10.09

(71)申请人 杭州安恒信息技术股份有限公司
地址 310000 浙江省杭州市滨江区西兴街道联慧街188号

(72)发明人 王英新 范渊

(74)专利代理机构 北京超凡宏宇专利代理事务所(特殊普通合伙) 11463
代理人 徐彦圣

(51)Int.Cl.
H04L 29/06(2006.01)

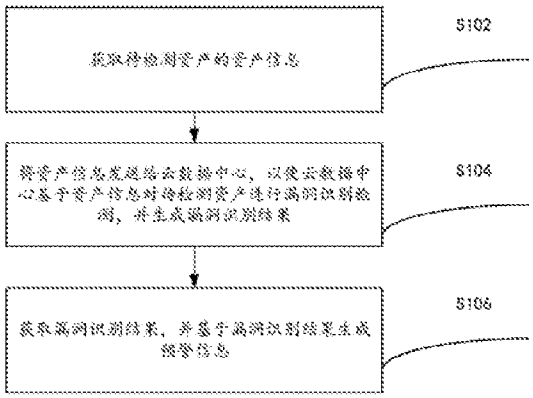
权利要求书2页 说明书8页 附图2页

(54)发明名称

资产漏洞的识别方法、装置和系统

(57)摘要

本发明提供了一种资产漏洞的识别方法、装置和系统，涉及网络安全的技术领域，包括：获取待检测资产的资产信息；将资产信息发送给云数据中心，以使云数据中心基于资产信息对待检测资产进行漏洞识别检测，并生成漏洞识别结果，其中，云数据中心存储有目标威胁情报数据，目标威胁情报数据包括以下至少之一：安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据；获取漏洞识别结果，并基于漏洞识别结果生成预警信息，解决了现有的资产漏洞识别方法对资产漏洞的检测较为单一的技术问题。



CN 110708315 A

1.一种资产漏洞的识别方法，其特征在于，应用于漏洞预警平台，包括：

获取待检测资产的资产信息；

将所述资产信息发送给云数据中心，以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果，其中，所述云数据中心存储有目标威胁情报数据，所述目标威胁情报数据包括以下至少之一：安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据；

获取所述漏洞识别结果，并基于所述漏洞识别结果生成预警信息。

2.根据权利要求1所述的方法，其特征在于，获取待检测资产的资产信息，包括：

向本地资产信息库发送获取指令，以使所述本地资产信息库基于所述获取指令采集所述待检测资产的属性信息和指纹信息，其中，所述属性信息包括但不限于以下信息：所述待检测资产的名称，所述待检测资产的所属责任人名称，所述指纹信息包括但不限于以下信息：所述待检测资产的出厂厂商名称，所述待检测资产的分类信息，所述待检测资产的操作系统信息；

获取所述属性信息和所述指纹信息，将所述属性信息和所述指纹信息确定为所述资产信息；

其中，所述分类信息包括：一级分类信息和二级分类信息，一级分类包括以下种类：服务器，网络设备，安全设备，二级分类包括以下种类：防火墙，交换机，路由器，数据库。

3.根据权利要求2所述的方法，其特征在于，若所述待检测资产的二级分类为数据库，则所述指纹信息还包括所述待检测资产的类型和版本。

4.根据权利要求1所述的方法，其特征在于，将所述资产信息发送给云数据中心，以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测，包括：

将所述资产信息通过所述云数据中心提供的云服务端口发送给所述云数据中心，以使所述云数据中心基于预设漏洞检测程序和所述资产信息对所述待检测资产进漏洞识别检测。

5.根据权利要求1所述的方法，其特征在于，

所述云数据中心用于提供基础设施即服务和平台即服务；

所述云数据中心基于Hadoop框架存储所述目标威胁情报数据。

6.一种资产漏洞的识别装置，其特征在于，所述装置包括：获取单元，发送单元和预警单元，其中，

所述获取单元用于获取待检测资产的资产信息；

所述发送单元用于将所述资产信息发送给云数据中心，以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果，其中，所述云数据中心存储有目标威胁情报数据，所述目标威胁情报数据威胁数据包括以下至少之一：安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据；

所述预警单元用于获取所述漏洞识别结果，并基于所述漏洞识别结果生成预警信息。

7.根据权利要求6所述的装置，其特征在于，所述获取单元还用于：

向本地资产信息库发送获取指令，以使所述本地资产信息库基于所述获取指令采集所述待检测资产的属性信息和指纹信息，其中，所述待检测资产的所属责任人名称，所述指纹

信息包括但不限于以下信息：所述待检测资产的出厂厂商名称，所述待检测资产的分类信息，所述待检测资产的操作系统信息；

获取所述属性信息和所述指纹信息，将所述属性信息和所述指纹信息确定为所述资产信息；

其中，所述分类信息包括：一级分类信息和二级分类信息，一级分类包括以下种类：服务器，网络设备，安全设备，二级分类包括以下种类：防火墙，交换机，路由器，数据库。

8.根据权利要求7所述的装置，其特征在于，若所述待检测资产的二级分类为数据库，则所述指纹信息还包括所述待检测资产的类型和版本。

9.根据权利要求6所述的装置，其特征在于，所述发送单元还用于：

将所述资产信息通过所述云数据中心提供的云服务端口发送给所述云数据中心，以使所述云数据中心基于预设漏洞检测程序和所述资产信息对所述待检测资产进行漏洞识别检测。

10.一种资产漏洞的识别系统，其特征在于，所述系统包括：本地资产信息库，云数据中心和漏洞预警平台，其中，

所述本地资产信息库用于采集并存储待检测资产的资产信息；

所述云数据中心用于基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果；

所述漏洞预警平台用于获取所述本地资产信息库中存储的待检测资产的资产信息，并将所述资产信息发送给所述云数据中心，以及基于所述漏洞识别结果生成预警信息。

资产漏洞的识别方法、装置和系统

技术领域

[0001] 本发明涉及网络安全技术领域，尤其是涉及一种资产漏洞的识别方法、装置和系统。

背景技术

[0002] 在当前网络安全技术条件下，网络攻击的发生基本是网络威胁利用信息资产所暴露的漏洞而实施的恶意行为。在网络空间中，网络威胁是客观存在的，要降低网络攻击的风险，对信息资产进行漏洞加固是非常有效的安全技术措施。但在进行资产漏洞加固之前，需要知道信息资产所面临的漏洞有哪些，因此，信息资产漏洞检测对于网络安全防护来说是异常重要的。

[0003] 但是，现有的漏洞检测方法存在对资产漏洞检测较为单一的技术问题。

[0004] 针对上述问题，还未提出有效的解决方案。

发明内容

[0005] 有鉴于此，本发明的目的在于提供一种资产漏洞的识别方法、装置和系统，以解决了现有的资产漏洞识别方法对资产漏洞的检测较为单一的技术问题。

[0006] 第一方面，本发明实施例提供了一种资产漏洞的识别方法，包括：获取待检测资产的资产信息；将所述资产信息发送给云数据中心，以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果，其中，所述云数据中心存储有目标威胁情报数据，所述目标威胁情报数据包括以下至少之一：安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据；获取所述漏洞识别结果，并基于所述漏洞识别结果生成预警信息。

[0007] 进一步地，获取待检测资产的资产信息，包括：向本地资产信息库发送获取指令，以使所述本地资产信息库基于所述获取指令采集所述待检测资产的属性信息和指纹信息，其中，所述待检测资产的所属责任人名称，所述指纹信息包括但不限于以下信息：所述待检测资产的出厂厂商名称，所述待检测资产的分类信息，所述待检测资产的操作系统信息；获取所述属性信息和所述指纹信息，将所述属性信息和所述指纹信息确定为所述资产信息；其中，所述分类信息包括：一级分类信息和二级分类信息，一级分类包括以下种类：服务器，网络设备，安全设备，二级分类包括以下种类：防火墙，交换机，路由器，数据库。

[0008] 进一步地，若所述待检测资产的二级分类为数据库，则所述指纹信息还包括所述待检测资产的类型和版本。

[0009] 进一步地，将所述资产信息发送给云数据中心，以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测，包括：将所述资产信息通过所述云数据中心提供的云服务端口发送给所述云数据中心，以使所述云数据中心基于预设漏洞检测程序和所述资产信息对所述待检测资产进行漏洞识别检测。

[0010] 进一步地，所述云数据中心用于提供基础设施即服务和平台即服务；所述云数据

中心基于Hadoop框架存储所述目标威胁情报数据。

[0011] 第二方面，本发明实施例提供了一种资产漏洞的识别装置，包括：获取单元，发送单元和预警单元，其中，所述获取单元用于获取待检测资产的资产信息；所述发送单元用于将所述资产信息发送给云数据中心，以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果，其中，所述云数据中心存储有目标威胁情报数据，所述目标威胁情报数据包括以下至少之一：安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据；所述预警单元用于获取所述漏洞识别结果，并基于所述漏洞识别结果生成预警信息。

[0012] 进一步地，所述获取单元还用于：向本地资产信息库发送获取指令，以使所述本地资产信息库基于所述获取指令采集所述待检测资产的属性信息和指纹信息，其中，所述待检测资产的所属责任人名称，所述指纹信息包括但不限于以下信息：所述待检测资产的出厂厂商名称，所述待检测资产的分类信息，所述待检测资产的操作系统信息；获取所述属性信息和所述指纹信息，将所述属性信息和所述指纹信息确定为所述资产信息；其中，所述分类信息包括：一级分类信息和二级分类信息，一级分类包括以下种类：服务器，网络设备，安全设备，二级分类包括以下种类：防火墙，交换机，路由器，数据库。

[0013] 进一步地，若所述待检测资产的二级分类为数据库，则所述指纹信息还包括所述待检测资产的类型和版本。

[0014] 进一步地，所述发送单元还用于：将所述资产信息通过所述云数据中心提供的云服务端口发送给所述云数据中心，以使所述云数据中心基于预设漏洞检测程序和所述资产信息对所述待检测资产进行漏洞识别检测。

[0015] 第三方面，本发明实施例提供了一种资产漏洞的识别系统，包括：产信息采集设备，云数据中心和漏洞预警平台，其中，所述本地资产信息库用于采集待检测资产的资产信息；所述云数据中心用于基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果；所述漏洞预警平台用于基于所述漏洞识别结果生成预警信息。

[0016] 在本发明实施例中，首先，获取待检测资产的资产信息；将资产信息发送给云数据中心，以使云数据中心基于资产信息对待检测资产进行漏洞识别检测，并生成漏洞识别结果，其中，云数据中心存储有目标威胁情报数据，目标威胁情报数据包括以下至少之一：安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据，漏洞识别结果用于表征待检测资产是否存在漏洞；获取漏洞识别结果，并基于漏洞识别结果生成预警信息。

[0017] 本申请中，通过将获取到的待检测资产的资产信息发送给云数据中心，以使云数据中心能够根据其存储的目标威胁情报数据，对待检测资产进行全面的漏洞检测，由于云数据中心能够根据汇集了多方提供的漏洞威胁情报数据对待检测资产进行漏洞检测，达到了对待检测资产进行全面漏洞检测的目的，进而解决了现有技术中资产漏洞的检测较为单一的技术问题，从而实现了对待检测资产进行全面漏洞检测的技术效果。

[0018] 本发明的其他特征和优点将在随后的说明书中阐述，并且，部分地从说明书中变得显而易见，或者通过实施本发明而了解。本发明的目的和其他优点在说明书、权利要求书以及附图中所特别指出的结构来实现和获得。

[0019] 为使本发明的上述目的、特征和优点能更明显易懂，下文特举较佳实施例，并配合

所附附图，作详细说明如下。

附图说明

[0020] 为了更清楚地说明本发明具体实施方式或现有技术中的技术方案，下面将对具体实施方式或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图是本发明的一些实施方式，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

[0021] 图1为本发明实施例提供的一种资产漏洞的识别方法的流程图；

[0022] 图2为本发明实施例提供的一种资产漏洞的识别装置的示意图；

[0023] 图3为本发明实施例提供的一种资产漏洞的识别系统的示意图；

[0024] 图4为本发明实施例提供的一种服务器的示意图。

具体实施方式

[0025] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合附图对本发明的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0026] 关于漏洞检测技术。当前针对信息资产的漏洞检测技术主要依靠专门的漏洞扫描设备通过主动探测技术来实现，在这种方式下，漏洞扫描设备利用其携带的“漏洞规则库”对各类信息资产进行主动扫描，如在对某一个资产进行扫描时，若其扫描结果命中了“漏洞规则库”中的规则R，则判定该资产具有R漏洞规则所对应的漏洞。在这种情况下，漏洞扫描的结果取决于漏洞扫描设备所携带的“漏洞规则库”的数量和质量。若某个信息资产所暴露出的某个漏洞在“漏洞规则库”中，则此漏洞可被漏洞扫描设备检测出来。若某个漏洞不在“漏洞规则库”中，则漏洞扫描设备无法检测出该漏洞，即便这个漏洞真实的存在。

[0027] 关于“漏洞规则库”的更新。对于新发现的漏洞，漏洞扫描设备提供商一般会定期或不定期的将这些漏洞打包成软件升级包，漏洞扫描设备通过在线或离线的升级完成“漏洞规则库”的升级，升级完成之后的漏洞扫描设备方可具备对新发现漏洞的检测能力。

[0028] 现有资产漏洞的识别方法存在以下的缺点：

[0029] 1、基于主动扫描的漏洞扫描技术需要通过网络向被扫描目标发送数据包，由于现实环境的复杂性，这可能造成目标主机的运行不稳定，造成信息资产的宕机、停止服务、数据丢失等。

[0030] 2、漏洞检测的滞后性。对于0day漏洞及未公开的漏洞，漏洞扫描设备由于“漏洞规则库”升级的滞后性，可能导致某些漏洞无法及时检测出来，进而存在恶意威胁利用该漏洞进行攻击的风险。

[0031] 3、检测能力单一。一般来说，某一厂商的漏洞检测设备都有其固有的技术及漏洞规则库的限制，无法发挥多个安全厂商及各类安全组织的漏洞检测能力，无法实现漏洞检测能力的聚合。

[0032] 4、漏洞检测的效率较低。由于传统的漏洞扫描设备技术实现方式原因，对多台资产进行漏洞检测时，效率较低，通常完成一台资产的漏洞检测需要数分钟。

[0033] 针对上述缺点，本申请采用以下方法克服上述缺点：

[0034] 实施例一：

[0035] 根据本发明实施例，提供了一种资产漏洞的识别方法实施例，需要说明的是，在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行，并且，虽然在流程图中示出了逻辑顺序，但是在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤。

[0036] 图1是根据本发明实施例的一种资产漏洞的识别方法，如图1所示，该方法包括如下步骤：

[0037] 步骤S102，获取待检测资产的资产信息；

[0038] 步骤S104，将所述资产信息发送给云数据中心，以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果，其中，所述云数据中心存储有目标威胁情报数据，所述目标威胁情报数据包括以下至少之一：安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据；

[0039] 步骤S106，获取所述漏洞识别结果，并基于所述漏洞识别结果生成预警信息。

[0040] 本申请中，通过将获取到的待检测资产的资产信息发送给云数据中心，以使云数据中心能够根据其存储的目标威胁情报数据，对待检测资产进行全面的漏洞检测，由于云数据中心能够根据汇集了多方提供的漏洞威胁情报数据对待检测资产进行漏洞检测，达到了对待检测资产进行全面漏洞检测的目的，进而解决了现有技术中资产漏洞的检测较为单一的技术问题，从而实现了对待检测资产进行全面漏洞检测的技术效果。

[0041] 需要说明的是，通过建立由安全服务提供商上传的威胁情报数据，安全情报组织上传的威胁情报数据和待检测资产所属企业上传的威胁情报数据组成的云数据中心，可以整合和对接各方安全威胁情报数据，形成威胁情报的数据协同和多方安全情报的能力协同，有利于安全威胁情报的共享机制建设。

[0042] 另外，还需要说明的是，上述的漏洞预警信息可以通过短信，邮件等方式发送给工作人员，以使工作人员能够及时对待检测资产的漏洞进行处理，从而降低待检测资产面临的风险。

[0043] 在本发明实施例中，步骤S102还包括如下步骤：

[0044] 步骤S11，向本地资产信息库发送获取指令，以使所述本地资产信息库基于所述获取指令采集所述待检测资产的属性信息和指纹信息，其中，所述待检测资产的所属责任人名称，所述指纹信息包括但不限于以下信息：所述待检测资产的出厂厂商名称，所述待检测资产的分类信息，所述待检测资产的操作系统信息；

[0045] 步骤S11，获取所述属性信息和所述指纹信息，将所述属性信息和所述指纹信息确定为所述资产信息。

[0046] 在本发明实施例中，在向本地资产信息库发送获取指令之后，本地资产信息库能够根据获取指令采集待检测资产的属性信息和指纹信息。

[0047] 然后，将获取本地资产信息库采集到的待检测资产的属性信息和指纹信息作为待检测资产的资产信息。

[0048] 需要说明的是，上述的属性信息包括但不限于以下信息：待检测资产的名称，待检

测资产的所属责任人名称等信息。

[0049] 上述的指纹信息包括但不限于以下信息:待检测资产的出厂厂商名称,待检测资产的分类信息,待检测资产的操作系统信息,其中,分类信息包括:一级分类信息和二级分类信息,一级分类包括以下种类:服务器,网络设备,安全设备,二级分类包括以下种类:防火墙,交换机,路由器,数据库。

[0050] 另外,还需要说明的是,待检测资产给的操作系统信息包括:操作系统的类型,操作系统的版本等信息。

[0051] 如果,待检测资产为数据库,那么待检测资产的指纹信息还包括代金侧资产的版本和类型。

[0052] 由于待检测资产的资产信息中包含指纹信息,因此,每个指纹信息能够确定一个待检测资产,进而可以通过资产指纹匹配技术准确的实现对待检测资产进行准确的被动的漏洞检测,降低漏洞检测时对被待检测资产的运行影响。

[0053] 另外,本地资产信息库在对待检测资产的资产信息进行采集可以通过自动方式或手动录入进行,资产信息的自动采集主要借助资产探测类设备完成,手动录入时,本地资产信息库可提供可视化的录入界面,以供工作人员将资产信息手动录入本地资产信息库中。

[0054] 在本发明实施例中,步骤S104还包括如下步骤:

[0055] 步骤S21,将所述资产信息通过所述云数据中心提供的云服务端口发送给所述云数据中心,以使所述云数据中心基于预设漏洞检测程序和所述资产信息对所述待检测资产进漏洞识别检测。

[0056] 在本发明实施例中,由于云数据中心提供的IaaS(基础设施即服务)及PaaS(平台即服务)作为基础层,利用开源的Hadoop架构搭建基于大数据处理架构的威胁情报数据库,威胁情报数据库中的数据来源可以是来自各个安全服务提供商、安全情报组织、企业自身。在PaaS(平台即服务)之上构建漏洞检测云应用程序(即,预设漏洞检测程序),本程序可调用威胁情报数据库中的数据,以便进行对待检测资产进行漏洞识别检测。同时,云应程序用通过互联网提供云服务接口,实现漏洞检测接口的服务化。

[0057] 由于云数据中心能够通过实时更新的威胁情报云数据作为基础数据,以云端接口的方式提供漏洞检测服务,可实时的为资产的0day漏洞检测和位置漏洞检测提供快速及时的检测结果,解决由于漏洞规则库滞后性所带来的风险,同时云数据中心以云端接口方式提供的漏洞检测服务能够批量对资产进行漏洞检测,提高了资产漏洞的检测效率。

[0058] 实施例二:

[0059] 本发明还提供了一种资产漏洞的识别装置,该装置用于执行本发明实施例上述内容所提供的资产漏洞的识别方法,以下是本发明实施例提供的资产漏洞的识别装置的具体介绍。

[0060] 如图2所示,上述的资产识别装置包括:获取单元10,发送单元20和预警单元30。

[0061] 所述获取单元10用于获取待检测资产的资产信息;

[0062] 所述发送单元20用于将所述资产信息发送给云数据中心,以使所述云数据中心基于所述资产信息对所述待检测资产进行漏洞识别检测,并生成漏洞识别结果,其中,所述云数据中心存储有目标威胁情报数据,所述目标威胁情报数据包括以下至少之一:安全服务提供商上传的威胁情报数据,安全情报组织上传的威胁情报数据和待检测资产所属企业上

传的威胁情报数据；

[0063] 所述预警单元30用于获取所述漏洞识别结果，并基于所述漏洞识别结果生成预警信息。

[0064] 本申请中，通过将获取到的待检测资产的资产信息发送给云数据中心，以使云数据中心能够根据其存储的目标威胁情报数据，对待检测资产进行全面的漏洞检测，由于云数据中心能够根据汇集了多方提供的漏洞威胁情报数据对待检测资产进行漏洞检测，达到了对待检测资产进行全面漏洞检测的目的，进而解决了现有技术中资产漏洞的检测较为单一的技术问题，从而实现了对待检测资产进行全面漏洞检测的技术效果。

[0065] 优选地，所述获取单元还用于：向本地资产信息库发送获取指令，以使所述本地资产信息库基于所述获取指令采集所述待检测资产的属性信息和指纹信息，其中，所述待检测资产的所属责任人名称，所述指纹信息包括但不限于以下信息：所述待检测资产的出厂厂商名称，所述待检测资产的分类信息，所述待检测资产的操作系统信息；获取所述属性信息和所述指纹信息，将所述属性信息和所述指纹信息确定为所述资产信息；其中，所述分类信息包括：一级分类信息和二级分类信息，一级分类包括以下种类：服务器，网络设备，安全设备，二级分类包括以下种类：防火墙，交换机，路由器，数据库。

[0066] 优选地，若所述待检测资产的二级分类为数据库，则所述指纹信息还包括所述待检测资产的类型和版本。

[0067] 优选地，所述发送单元还用于：将所述资产信息通过所述云数据中心提供的云服务端口发送给所述云数据中心，以使所述云数据中心基于预设漏洞检测程序和所述资产信息对所述待检测资产进行漏洞识别检测。

[0068] 实施例三：

[0069] 本发明还提供了一种资产漏洞的识别系统，该装置用于执行本发明实施例上述内容所提供的资产漏洞的识别系统，以下是本发明实施例提供的资产漏洞的识别系统的具体介绍。

[0070] 如图3所示，上述的资产识别系统包括：本地资产信息库100，云数据中心200和漏洞预警平台300。

[0071] 所述本地资产信息库100用于采集并存储待检测资产的资产信息；

[0072] 所述云数据中心200用于基于所述资产信息对所述待检测资产进行漏洞识别检测，并生成漏洞识别结果；

[0073] 所述漏洞预警平台300用于获取所述本地资产信息库中存储的待检测资产的资产信息，并将所述资产信息发送给所述云数据中心，以及基于所述漏洞识别结果生成预警信息。

[0074] 本申请中，通过将获取到的待检测资产的资产信息发送给云数据中心，以使云数据中心能够根据其存储的目标威胁情报数据，对待检测资产进行全面的漏洞检测，由于云数据中心能够根据汇集了多方提供的漏洞威胁情报数据对待检测资产进行漏洞检测，达到了对待检测资产进行全面漏洞检测的目的，进而解决了现有技术中资产漏洞的检测较为单一的技术问题，从而实现了对待检测资产进行全面漏洞检测的技术效果。

[0075] 参见图4，本发明实施例还提供一种服务器400，包括：处理器50，存储器51，总线52和通信接口53，所述处理器50、通信接口53和存储器51通过总线52连接；处理器50用于执行

存储器51中存储的可执行模块，例如计算机程序。

[0076] 其中，存储器51可能包含高速随机存取存储器(RAM，Random Access Memory)，也可能还包括非不稳定的存储器(non-volatile memory)，例如至少一个磁盘存储器。通过至少一个通信接口53(可以是有线或者无线)实现该系统网元与至少一个其他网元之间的通信连接，可以使用互联网，广域网，本地网，城域网等。

[0077] 总线52可以是ISA总线、PCI总线或EISA总线等。所述总线可以分为地址总线、数据总线、控制总线等。为便于表示，图4中仅用一个双向箭头表示，但并不表示仅有一根总线或一种类型的总线。

[0078] 其中，存储器51用于存储程序，所述处理器50在接收到执行指令后，执行所述程序，前述本发明实施例任一实施例揭示的流过程定义的装置所执行的方法可以应用于处理器50中，或者由处理器50实现。

[0079] 处理器50可能是一种集成电路芯片，具有信号的处理能力。在实现过程中，上述方法的各步骤可以通过处理器50中的硬件的集成逻辑电路或者软件形式的指令完成。上述的处理器50可以是通用处理器，包括中央处理器(Central Processing Unit，简称CPU)、网络处理器(Network Processor，简称NP)等；还可以是数字信号处理器(Digital Signal Processing，简称DSP)、专用集成电路(Application Specific Integrated Circuit，简称ASIC)、现成可编程门阵列(Field-Programmable Gate Array，简称FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本发明实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本发明实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器，闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于存储器51，处理器50读取存储器51中的信息，结合其硬件完成上述方法的步骤。

[0080] 另外，在本发明实施例的描述中，除非另有明确的规定和限定，术语“安装”、“相连”、“连接”应做广义理解，例如，可以是固定连接，也可以是可拆卸连接，或一体地连接；可以是机械连接，也可以是电连接；可以是直接相连，也可以通过中间媒介间接相连，可以是两个元件内部的连通。对于本领域的普通技术人员而言，可以具体情况理解上述术语在本发明中的具体含义。

[0081] 在本发明的描述中，需要说明的是，术语“中心”、“上”、“下”、“左”、“右”、“竖直”、“水平”、“内”、“外”等指示的方位或位置关系为基于附图所示的方位或位置关系，仅是为了便于描述本发明和简化描述，而不是指示或暗示所指的装置或元件必须具有特定的方位、以特定的方位构造和操作，因此不能理解为对本发明的限制。此外，术语“第一”、“第二”、“第三”仅用于描述目的，而不能理解为指示或暗示相对重要性。

[0082] 在本申请所提供的几个实施例中，应该理解到，所揭露的系统、装置和方法，可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，又例如，多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些通信接口，装置或单元的间接

耦合或通信连接，可以是电性，机械或其它的形式。

[0083] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0084] 另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。

[0085] 最后应说明的是：以上所述实施例，仅为本发明的具体实施方式，用以说明本发明的技术方案，而非对其限制，本发明的保护范围并不局限于此，尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，其依然可以对前述实施例所记载的技术方案进行修改或可轻易想到变化，或者对其中部分技术特征进行等同替换；而这些修改、变化或者替换，并不使相应技术方案的本质脱离本发明实施例技术方案的精神和范围，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应所述以权利要求的保护范围为准。

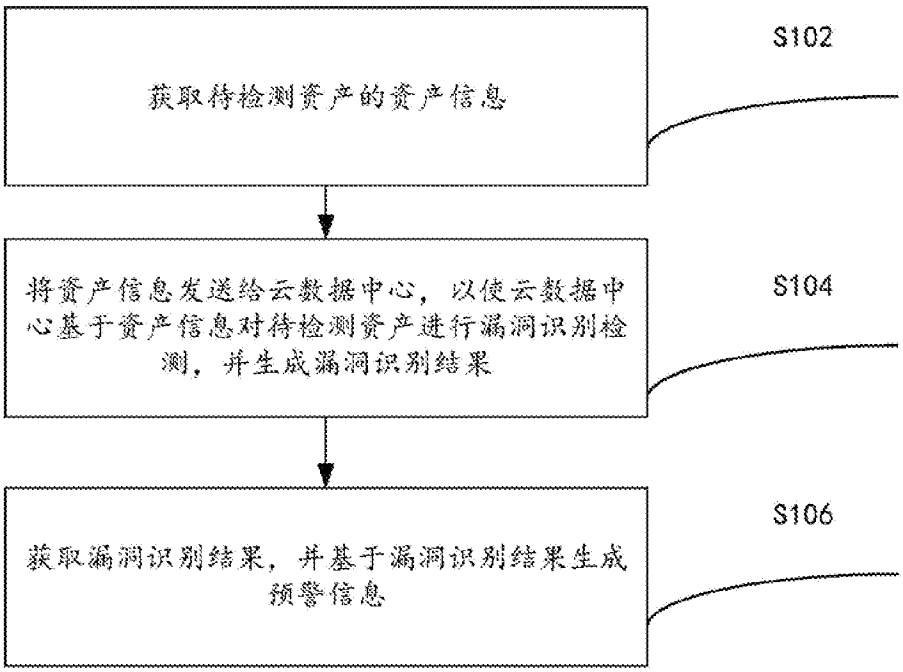

S102
获取待检测资产的资产信息
S104
将资产信息发送给云数据中心，以使云数据中心基于资产信息对待检测资产进行漏洞识别检测，并生成漏洞识别结果
S106
获取漏洞识别结果，并基于漏洞识别结果生成预警信息

图1

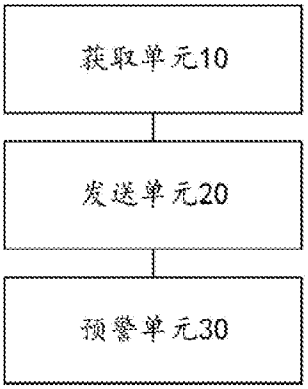

获取单元10
发送单元20
预警单元30

图2

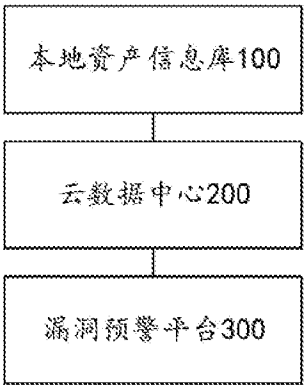

本地资产信息库100
云数据中心200
漏洞预警平台300

图3

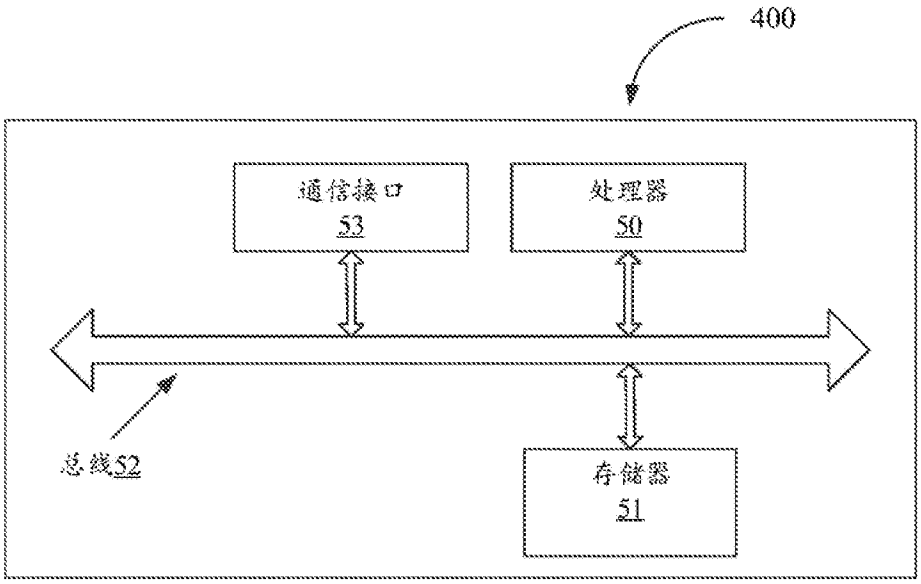

400
通信接口
53
处理器
50
总线52
存储器
51

图4

INFORMATION DISCLOSURE STATEMENT BY APPLICANT *(Use as many sheets as necessary)*				***Complete if Known***	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	2431
				Examiner Name	SU, SARAH
Sheet	1	of	1	*Attorney Docket Number*	**15635.0008-00000**

U.S. PATENTS

Examiner Initials*	Cite No.	Document Number Number-Kind Code *(if known)*	Issue Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-8,621,073-B2	12-31-2013	McClure et al.	

U.S. PUBLISHED PATENT APPLICATIONS

Examiner Initials*	Cite No.	Document Number Number-Kind Code *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-2018/0288087-A1	10-04-2018	HITTEL et al.	
		US-2017/0012998-A1	01-12-2017	Thornbury et al.	
		US-2021/0377287-A1	12-02-2021	SHUA	

Note: Submission of copies of U.S. Patents and published U.S. Patent Applications is not required.

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.	Foreign Patent Document Country Code Number Kind Code *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	Translation

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	Translation
		Final Office Action Application No. 17/658,156 dated September 13, 2022, in the United States Patent and Trademark Office. (15635.0010-00000)	N/A
		Notice of Allowance No. 17/716,491 dated October 11, 2022, in the United States Patent and Trademark Office. (15635.0006-00000)	N/A
		Notice of Allowance No. 17/716,231 dated September 6, 2022, in the United States Patent and Trademark Office. (15635.0004-00000)	N/A

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

UNITED STATES
PATENT AND TRADEMARK OFFICE

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

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/657,981	**10/28/2022 11:26:59 AM ET**	**15635.0008-00000**

Title of Invention

Forward and Rearward Facing Attack Vector Visualization

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2397	FILED BY	Wanda Whittington
PATENT CENTER #	61117493	FILING DATE	04/05/2022
CUSTOMER #	22852	FIRST NAMED INVENTOR	Avi SHUA
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Christopher Johns

Documents

TOTAL DOCUMENTS: 5

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
NPL-15635-0008-00000Final Office Action dated 09-13-2022 -17658156_NPL.pdf	21	Non Patent Literature	1317 KB
NPL-15635-0008-00000Notice of Allowability 09-06-22 -17716231_NPL.pdf	9	Non Patent Literature	357 KB
NPL-15635-0008-00000Notice of Allowance dated 10-11-22 -17716491_NPL.pdf	11	Non Patent Literature	829 KB
15635-0008-0000-as-filedsb08-10-28-2022.pdf	1	Information Disclosure Statement (IDS) Form (SB08)	33 KB

Warning: This is not a USPTO supplied IDS fillable form. Data in the form cannot be automatically loaded to other USPTO systems.

15635-0008-00000idstransmittal-as-filed10-28-2022.pdf	2	Transmittal Letter	17 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
NPL-15635-0008-00000Final Office Action dated 09-13-2022-17658156_NPL.pdf	DA1C8F743A4E567DE0EF1F34C03940DBB730169C3E11A55F08F83505D531BE3C9A4CD41D9A442832F450CA757975D9CB74BF3441128B2DEA2E27200F08FFBDFD
NPL-15635-0008-00000Notice of Allowability 09-06-22 - 17716231_NPL.pdf	85978E8718401FC06B9DB93C55A71FFD05C96B93C07D10ABBC35435058EBDDE1529863E85EFB5802DD17FF96B585B121D574D8C913E76CE550A1DA73A4D4CF1D
NPL-15635-0008-00000Notice of Allowance dated 10-11-22 - 17716491_NPL.pdf	E04EC1C51D367450510673181F166FBFEE4E783B0131F6CBCF744BE53F831D3C5DDAC81D6B5B7EAB913882D52683D25B3FF2EE32B2285FC9D8E035363068DAA5
15635-0008-0000-as-filedsb08-10-28-2022.pdf	3CADBAABABC319C31806C7E2227E098C697FF877226D8103543BD07258629475C8C8DD050FFF4300FF0DC7063874916E92A6462A3558D9D90FA73FCE643F1452
15635-0008-00000idstransmittal-as-filed10-28-2022.pdf	E8E3B03AFF88C58D700C1465FB98447B0FF41062A92D6A1DE954DE1E60AECC0FBD26E9B32280C975A45A7294603967329DC20763F9E9D3BBB70F37CA578F15A2

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

New Applications Under 35 U.S.C. 111

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

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

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

New International Application Filed with the USPTO as a Receiving Office

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

PATENT
Attorney Docket No. 15635.0008-00000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Avi SHUA	)	Group Art Unit: 2431
	)	
Application No.: 17/657,981	)	Examiner: SU, SARAH
	)	
Filed: April 5, 2022	)	Confirmation No.: 2397
	)	
For: FORWARD AND REARWARD FACING ATTACK VECTOR VISUALIZATION	)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. § 1.97(b)

Pursuant to 37 C.F.R. §§ 1.56 and 1.97(b), Applicant brings to the attention of the Examiner the listed document on the attached Form PTO/SB/08. This Information Disclosure Statement is being filed before the mailing date of a first Office Action on the merits for the above-referenced application.

Copies of the non-patent literature documents are attached.

Applicant respectfully requests that the Examiner consider the listed documents on the attached Form PTO/SB/08 and indicate that they were considered by making appropriate notations on the attached form.

This submission does not represent that a search has been made or that no better art exists and does not constitute an admission that each or all of the listed documents are material or constitute "prior art." If the Examiner applies any of the documents as prior art against any claim in the application and Applicant determines that the cited documents do not constitute "prior art" under United States law, Applicant reserves the right to present to the U.S. Patent and Trademark Office the relevant facts and law regarding the appropriate status of such documents.

Applicant further reserves the right to take appropriate action to establish the patentability of the disclosed invention over the listed documents, should one or more of the documents be applied against the claims of the present application.

If there is any fee due in connection with the filing of this Statement, please charge the fee to deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: October 28, 2022

By: /Christopher C. Johns/
Christopher C. Johns
Reg. No. 68,664
(202) 408-4000

UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
17/657,981	04/05/2022	Avi SHUA	15635.0008-00000

CONFIRMATION NO. 2397

22852
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

PUBLICATION NOTICE



Title: Forward and Rearward Facing Attack Vector Visualization

Publication No.US-2022-0342690-A1
Publication Date:10/27/2022

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Public Records Division. The Public Records Division can be reached by telephone at (571) 272-3150 or (800) 972-6382, by facsimile at (571) 273-3250, by mail addressed to the United States Patent and Trademark Office, Public Records Division, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently https://portal.uspto.gov/pair/PublicPair. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

INFORMATION DISCLOSURE STATEMENT BY APPLICANT *(Use as many sheets as necessary)*				***Complete if Known***	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	OPAP
				Examiner Name	CENTRAL, DOCKET
Sheet	1	of	3	*Attorney Docket Number*	**15635.0008-00000**

U.S. PATENTS

Examiner Initials*	Cite No.	Document Number Number-Kind Code *(if known)*	Issue Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-11,036,856B2	06-15-2021	Graun et al.	
		US-9,794,289B1	10-17-2017	Banerjee et al.	
		US-9,749,349B1	08-29-2017	Czarny et al.	
		US-9,569,201B1	02-14-2017	Lotem et al.	
		US-9,294,498B1	03-22-2016	Yampolskiy et al.	
		US-8,272,061B1	09-18-2012	Lotem et al.	
		US-9,356,942B1	05-31-2016	Joffe	
		US-9,413,774B1	08-09-2016	Liu et al.	
		US-11,075,923B1	07-27-2021	Srinivasan et al.	
		US-7,013,395B1	03-14-2006	Swiler et al.	
		US-10,868,825B1	12-15-2020	Dominessy et al.	
		US-10,152,487B1	12-11-2018	Patwardhan et al.	
		US-7,143,143B1	11-28-2006	Thompson	
		US-9,686,296B1	06-20-2017	Murchison et al.	

U.S. PUBLISHED PATENT APPLICATIONS

Examiner Initials*	Cite No.	Document Number Number-Kind Code *(if known)*	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-2020/0244692A1	07-30-2020	SHUA	
		US-2019/0327272A1	10-24-2019	NARAYANASWAMY	
		US-2020/0097662A1	03-26-2020	Hufsmith et al.	
		US-2013/0262801A1	10-03-2013	Sancheti et al.	
		US-2022/0210195A1	06-30-2022	Parekh et al.	
		US-2019/0250857A1	08-15-2019	Reddy et al.	
		US-2021/0409438A1	12-30-2021	Sundaram et al.	
		US-2020/0389483A1	12-10-2020	Dye et al.	
		US-2022/0141247A1	05-05-2022	Shakarian et al.	
		US-2020/0344256A1	10-29-2020	Alabdulhadi	
		US-2022/0191250A1	06-16-2022	Laliberte	
		US-2014/0366118A1	12-11-2014	Yin	
		US-2020/0220887A1	07-09-2020	A. K.	
		US-2018/0367515A1	12-20-2018	Anandam et al.	

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use as many sheets as necessary)				Complete if Known	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	OPAP
				Examiner Name	CENTRAL, DOCKET
Sheet	2	of	3	Attorney Docket Number	**15635.0008-00000**

U.S. PUBLISHED PATENT APPLICATIONS

Examiner Initials*	Cite No.	Document Number Number-Kind Code (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-2022/0182294A1	06-092022	Li et al.	
		US-2022/0100869A1	03-31-2022	BERGER et al.	
		US-2020/0396254A1	12-17-2020	Crabtree et al.	
		US-2016/0330237A1	11-10-2016	Edlabadkar	
		US-2018/0205755A1	07-19-2018	Kavi et al.	
		US-2021/0342297A1	11-04-2021	Gupta et al.	
		US-2022/0239681A1	07-28-2022	Newman et al.	
		US-2022/0124110A1	04-21-2022	CHHABRA et al.	
		US-2021/0099478A1	04-01-2021	Seetharamaiah et al.	
		US-2020/0228560A1	07-16-2020	Murthy et al.	

Note: Submission of copies of U.S. Patents and published U.S. Patent Applications is not required.

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.	Foreign Patent Document Country Code Number Kind Code (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	Translation
		WO 2020/106973A1	05-28-2020	ARAALI NETWORKS INC.		N/A
		WO 2019/051507A1	03-14-2019	CARBON BLACK INC.		N/A
		WO 2017/153983A1	09-14-2017	B. G. NEGEV TECHNOLOGIES AND APPLICATIONS LTD.		N/A

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	Translation
		Non-Final Office Action Application No. 17/716,491 dated August 3, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/657,986 dated August 17, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/658,156 dated July 18, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/716,583 dated August 18, 2022, in the United States Patent and Trademark Office.	N/A
		Non-Final Office Action Application No. 17/657,972 dated June 21, 2022, in the United States Patent and Trademark Office.	N/A
		Final Office Action Application No. 17/657,972 dated September 1, 2022, in the United States Patent and Trademark Office.	N/A
		Ex Parte Quayle Action Application No. 17/716,231 dated June 26, 2022, in the United States Patent and Trademark Office.	N/A

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT *(Use as many sheets as necessary)*				***Complete if Known***	
				Application Number	17/657,981
				Filing Date	April 5, 2022
				First Named Inventor	Avi SHUA
				Art Unit	OPAP
				Examiner Name	CENTRAL, DOCKET
Sheet	3	of	3	*Attorney Docket Number*	**15635.0008-00000**

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

UNITED STATES
PATENT AND TRADEMARK OFFICE

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

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
17/657,981	**09/07/2022 12:24:36 PM ET**	**15635.0008-00000**

Title of Invention

Forward and Rearward Facing Attack Vector Visualization

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	2397	FILED BY	Michelle Daniels
PATENT CENTER #	60944414	FILING DATE	04/05/2022
CUSTOMER #	22852	FIRST NAMED INVENTOR	Avi SHUA
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Christopher Johns

Documents

TOTAL DOCUMENTS: 12

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
15635.0008-IDS_before_OA.pdf	2	Transmittal Letter	16 KB
15635.0008-sb08.pdf	3	Information Disclosure Statement (IDS) Form (SB08)	43 KB
Warning: This is not a USPTO supplied IDS fillable form. Data in the form cannot be automatically loaded to other USPTO systems.			
WO2020106973A1_Original_document_20220824215001.pdf	56	Foreign Reference	4243 KB
WO2019051507A1_Original_	69	Foreign Reference	4705 KB

document_20220824215045.pdf			
WO2017153983A1_Original_document_20220824215120.pdf	16	Foreign Reference	869 KB
Non-Final Office Action dated 08-03-2022 - 17 716,491.pdf	8	Non Patent Literature	429 KB
Non-Final Office Action dated 08-17-2022 - 17 657,986.pdf	13	Non Patent Literature	758 KB
Non-Final Office Action dated 07-18-2022 - 17 658,156.pdf	16	Non Patent Literature	915 KB
Non-Final Office Action dated 08-18-2022 - 17 716,583.pdf	13	Non Patent Literature	716 KB
Final Office Action dated 09-01-2022 - 17 657,972.pdf	17	Non Patent Literature	905 KB
Non-Final Office Action dated 06-21-22 - 17 657,972.pdf	13	Non Patent Literature	396 KB
Ex Parte Quayle Action dated - 07-26-22 - 17 716,231.pdf	9	Non Patent Literature	452 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
15635.0008-IDS_before_OA.pdf	205B21A094AF34D0090C0CA26B9E9763DB4A39531221F411015508EC369A6C10DD5B61BBD0C1E7A0F57BC5D4E583DD3D2D2AE8CD3EF2760D881DBCE869235A2A
15635.0008-sb08.pdf	FDC1C626C2A0C52F48C2D3C555B9D42111E8E7E196A3DDC1A33D0DC31185FD7F4626CE22BDFBEEC185089A98AEAF746A

	CAE5D929045A45B9EFA9DB3DFA01FB6D
WO2020106973A1_Original_document_20220824215001.pdf	E11441120E6AF16378204B5DB1DB9C959C897EF13EB82278ED00352710F8416CFA18F1920B2637826471BE70B378983A7776EDDD5A74B0FBAB020D4CC33FD7D2
WO2019051507A1_Original_document_20220824215045.pdf	3FA4E891E8A81CA5667740B392A17A40B94D8FC139EF6BDE24AAA3DE00FDC11DB2E055EED6AC7B1A07144F25317AB7A704145699529F2E650840391637554 4DB
WO2017153983A1_Original_document_20220824215120.pdf	8FC6DCDF5F6EDECA183538C43C4DC946428926DBE35508AB770DDFECA2285080947FC35FD9B15DD76F0EC7C61434FED3C0D6F6159B2A9DD4FD5006EF5C2E4D6B
Non-Final Office Action dated 08-03-2022 - 17 716,491.pdf	BE91EA18215C04454FFE0E953D7B71DF9BFBD950621446233D41AE62FE0BF3D0B1BE898310FFE2EE32E6B3448AC5F3B52DFBD7A59ACEF66DEE61B4FC1BCF37C6
Non-Final Office Action dated 08-17-2022 - 17 657,986.pdf	3F17A9071D8FA03B72416E7D75C3EF081442DFBDC3A03535389794BE137078687C103D1D42F08010A213E573C386E7EC6EC50A6CF0E9EF3C5AF5E709F6A96360
Non-Final Office Action dated 07-18-2022 - 17 658,156.pdf	389051A0BCA5A56E649D2310E05863D9CBB3ED1A8ED1559EC2527F93770A48444A186BE8AC581BA3B424D2739C031B2D08D7446D0C729D043E32FB1502F75612
Non-Final Office Action dated 08-18-2022 - 17 716,583.pdf	54DE2F031B7EDE5354937C74B4E034272295DD5F865686AE9F3327E0C29EE370CC226BA43F81FF2F6F389A5BD82D72DC0A78048342AD5A134C7FAD98BA36C33A
Final Office Action dated 09-01-2022 - 17 657,972.pdf	07DA528809B2A21FD128F090CA7B0860CD357FC7DD84D8300E10840E56000B037BF119ED9CA09F0273884FABBFFC35A195780705A9652907C67E411AAD9C1D82
Non-Final Office Action dated 06-21-22 - 17 657,972.pdf	00E453A2C63CB3C66A2D0BF0073BCDFC72960A4084AB79C068DE58B2CFAFD030CC464FDAE4195FA96ACBAD125ABB61DC44DFD3A0E26F5174E3C4E026A4AD5FF5
Ex Parte Quayle Action dated - 07-26-22 - 17 716,231.pdf	B5F09983761C777884AC43965FDFF4AB54B40B08A36BB6F37C56E106364BCB7FEC045AFCA9BA9915DD28195C916D226042

FD35EA803A9A112348CD01D2504926

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

New Applications Under 35 U.S.C. 111

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

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

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

New International Application Filed with the USPTO as a Receiving Office

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

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 September 2017 (14.09.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/153983 A1

(51) International Patent Classification:
G06F 21/56 (2013.01) *G06F 21/55* (2013.01)
G06F 21/60 (2013.01) *G06F 21/50* (2013.01)

(21) International Application Number:
PCT/IL2017/050277

(22) International Filing Date:
7 March 2017 (07.03.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/304,958 8 March 2016 (08.03.2016) US

(71) Applicant: **B. G. NEGEV TECHNOLOGIES AND APPLICATIONS LTD., AT BEN-GURION UNIVERSITY** [IL/IL]; P.O.B. 653, 8410501 Beer Sehva (IL).

(72) Inventor: **GURI, Mordechai**; 28 Emek Ayalon Street, 7170656 Modi'in (IL).

(74) Agents: **FUERST, Zadok** et al.; Luzzatto & Luzzatto, P.O. Box 5352, 8415202 Beer Sheva (IL).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: SYSTEM AND METHOD FOR PERFORMING IN-CLOUD SECURITY OPERATIONS ON CONNECTED DEVICES

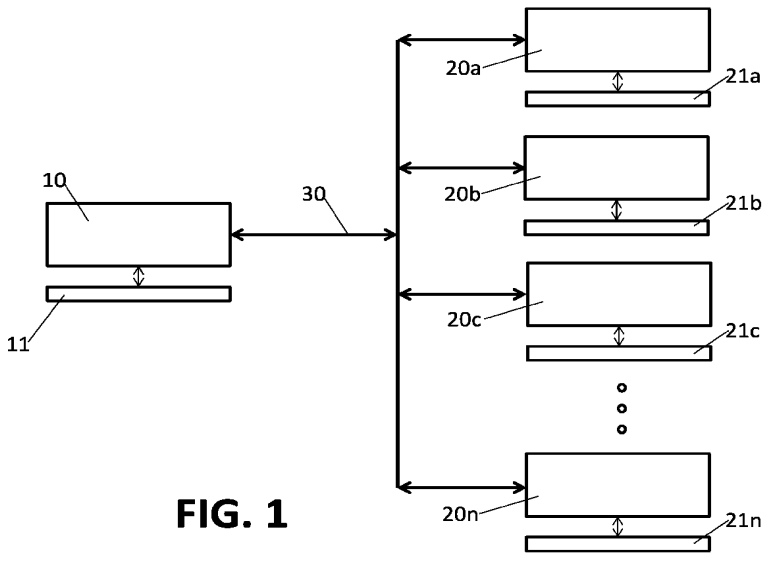


FIG. 1

(57) **Abstract**: The invention relates to a system for protecting IoT devices from malicious code, which comprises: (a) a memory extracting module at each of said IoT devices, for extracting a copy of at least a portion of the memory content from the IoT device, and sending the same to an in-cloud server; and (b) an in-cloude server for receiving said memory content, and performing an integrity check for a possible existance of malicious code within said memory content.

WO 2017/153983 A1

SYSTEM AND METHOD FOR PERFORMING IN-CLOUD SECURITY OPERATIONS ON CONNECTED DEVICES

Field of Invention

The field of the invention relates in general to methods and systems for securing computerized environments and devices. More specifically, the invention relates to a method and a system for checking the integrity and authenticity of "Internet of Things" (IoT) type devices, thereby detecting whether they have been infected by malicious code.

Background of the Invention

The "Internet of Things (IoT)" is the internetworking of physical devices, vehicles (also referred to as "connected devices" and "smart devices"), buildings, and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system and is able to interoperate within the existing Internet infrastructure. According to Wikipedia, Experts estimate that the IoT will consist of almost 50 billion objects ("things") by 2020.

Typically, IoT offers advanced connectivity of devices, systems, and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications. The interconnection of these embedded

devices (including smart objects), is expected to be adopted in in nearly all fields of automation.

For example, the range of IoT devices includes heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, or field operation devices that assist firefighters in search and rescue operations. These devices use sensors to collect useful data with the help of various existing technologies, and then autonomously flow the data between other devices. Other examples include home automation (smart home devices) such as the control and automation of lighting, heating (like smart thermostat), ventilation, air conditioning (HVAC) systems, and appliances such as washer/dryers, robotic vacuums, air purifiers, ovens or refrigerators/freezers that use Wi-Fi for remote monitoring.

As discussed, the IoT devices that are used in many fields and structures, are manufactured by a huge number of different manufacturers. While the protocol for interconnectivity and communication between various of these device has been defined and standartized, still a vast majority of these devices are characterized by:

a. They apply a great variety of proprietary operating systems (OS), software and hardware;

b. They have a limited, in many cases very limited processing power; and

c. The devices are in many cases very cheap.

Although the IoT devices are typically characterized by said (a) – (c) above, still these devices in many cases are used to control or sense very critical elements. The exploitation of these devices by hostile entities, for example, by injection of malicious code, can cause very significant damages, such as stealing of data, manipulation of the operation of the devices, or using the attacked device as a platform for attacking other device (e.g., bot). Therefore, in contrast to (a) the low cost of each of said devices; (b) the fact that the devices may have a very limited

processing power; and (c) the fact that each of the devices may have a different proprietary oprating system that may require a dedicated protection software; there is still a real need to protect the IoT devices from damages resulting from malicious code. This is particularly due to the fact that all these devices are accessible to hackers via the Internet.

The prior art techniques that have been so far adopted for protecting IoT devices from malicious code are typically tradional, for example:

a. Use of a firewall, which substantially isolates the internal network of the IoT devices from the external domain;
b. The use of conventional anti-virus software, whenever possible. This is, however, impractical in IoT devices having proprietary operating systems, in view of the huge variety of the proprietary operating systems involved. Furthermore, this is in many cases impractical in view of the high difference between the value of the IoT device and the cost of developing such software to account for this large variety of proprietary operating systems. Finally, the limited processing power of the IoT devices does not always enable the use of anti-virus software.
c. The use of network based security systems such as firewalls, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS). These systems are monitoring the network traffic and try to detect attacks or malicious activities. However, this approach is limited only to attacks that can be detect from the network traffic. Moreover, this approach is very limited in detecting attacks that are hiding within encrypted traffic.

More specifically, the prior art has typically applies the classic approach that is typically used for end-points and workstations, which involves running detection programs on the IoT devices to detect, scan and analyze for malicious code within the (1) persistent storage (e.g., disk) and (2) non-persistent storage (e.g., memory).

However, this approach is not applicable in many case, in view of the limited computational power of the IoT device.

In another aspect, cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services) which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned, or third-party data centers that may be located far from the user–ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over an electricity network.

It is therefore an object of the present invention to provide a method and system for protecting IoT devices from malicious code.

It is another object of the present invention to provide a method and system that can protect IoT devices, that wereotherwise remained unprotected.

It is still another object of the present invention to provide such a system that can detect malicious code in IoT devices, provide an alert, and remove the malicious code.

It is still another object of the present invention to provide upgrades and updates to the operating systems of the IoT devices, whether these operating systems are standard, or proprietary.

Other objects and advantages of the invention will become apparent as the description proceeds.

Summary of the Invention

The invention relates to a system for protecting IoT devices from malicaious code, which comprises: (a) a memory extracting module at each of said IoT devices, for extracting a copy of at least a portion of the memory content from the IoT device, and sending the same to an in-cloud server; and (b) an in-cloude server for receiving said memory content, and performing an integrity check for a possible existance of malicious code within said memory content.

In an embodiment of the invention, said in-cloud server performs one or more of the following: (a) analysis of the memory to find malware using static analysis methods; analysis of the memory to find malicious behavior using behavioral and heuristics methods; (b) reconstruction of the state of an OS of the IoT to determine and report important structural elements; (c) check of the integrity of the OS and its memory to possibly find fault in the integrity or hidden processes; (d) a cross-view check on resources to find rootkits and hidden operations; and (e) a cross-view check and validation of memory contents of plurality of IoT devices.

In an embodiment of the invention, and following said integrity check, said in-cloud server performs one or more of the following:
(1) logging of the results;
(2) reporting the results, raising a warning or an alert in a case of detection of an unexpected code or behaveior; or
(3) communicating and responding to an IoT request.

In an embodiment of the invention, the memory, a copy of which is sent to the in-cloud server, is either a persistent memory or a non-persistent memory.

In an embodiment of the invention, said memory extraction module is embedded within a kernel of a respective operating system of the IoT device.

In an embodiment of the invention, said memory extraction module is positioned within a trusted layer at the IoT device.

In an embodiment of the invention, said memory extraction module is positioned within a Trusted Execution Environment at the processor of the IoT device.

Brief description of the Drawings

In the drawings:

- Fig. 1 illustrates the general structure of the system of the present invention.

Detailed Description of Preferred Embodiments of the Invention

As noted, there are many cases in which the classical techniques are not applicable for protecting IoT devices from damages of malicious codes. The present invention provides alternative system and method for protecting IoT devices.

Fig. 1 shows a general structure of the system of the present invention. The system comprises in general an in-cloud protection server 10, which is used to protect a plurality of remote IoT devices 20a – 20n from malicious code. Several of the devices 20a- 20n may be stand-alone devices, others may be a part of IoT networks. In general, the in-cloud server 10 may serve a very large number of IoT devices. For example, a single in-cloud server may serve hundreds of thouthands, even millions of IoT devices 20, that are spread along the globe.

Each of the devices 20a-20n comprises a memory extraction modul 21a-21n, respectively. In one embodiment, the memory extraction module 21 is embedded within the kernel of the respective operating system of device 20. In another embodiment, the memory extraction module 21 is positioned within a trusted layer, such as hypervisor, at device 20. In still another embodiment, the memory extraction module may be positioned within a Trusted Execution Environment (TEE) at the processor of the IoT device.

The memory extraction modules 21 at each of the devices 20 extracts, either upon demand from server 10, or independently the memory content (persistent and/or non persistent), and transmits the same to the in-cloud server 10 for inspection and verification.

Upon receipt of the memory content (or a portion thereof) of a device 20 from a respective memory extraction module 21, the in-cloud server 10 performs one or more of the following operations:

(1) Analysis of the memory to find malware using static analysis methods;

(2) Analysis of the memory to find malicious behavior using behavioral and heuristics methods;

(3) Reconstruction of the state of the OS to determine and report important structural elements, such as:

a. Process/thread list
b. Communication ports
c. Kernel modules
d. Objects in memory
e. Object in cache
f. Open/close files
g. System status
h. Bootstrap information
i. Memory corruptions

(4) Check of the integrity of the OS and its memory to find fault in the integrity or hidden processes;

(5) A cross-view check on resources to find rootkits and hidden operations. The cross-view collaborates with components as reported from within the OS; and.

(6) A cross-view check and validation of memory contents of plurality of IoT devices.

Following the above perations, the in-cloud server may perform one ore more of the following operations:

(1) Logging of the results;
(2) Reporting the results, raising a warning or an alert in a case of detection of an unexpected code or behaveior; or
(3) Communicating and responding to an IoT request.

For its proper operation, server 10 has a database 11 that contains authentic and reliable data for comparison and verification with the memory content which is received from the IoT devices. For example, the database 11 may contain a copy of the authentic OS which is used in each IoT device 20 and its version number. The database content, whether relating to standard data or proprietary data, is accumulated by the operator of server 10, for example, by contacting the manufacturers of devices 20. The fact that a single server 10 can serve a huge number of devices 20 significantly reduces the costs of obtaining such data. Moreover, the communication between server 10 and devices 20 is typically performed over a secured channel 30. The system of the present invention also overcomes the problem which is associated with the typical IoT devices, namely the lack of sufficient processing power to perform the integrity check in the classical approach.

Server 10 may also perform an update of the OS of each of the IoT devices, when it becomes necessary. The integrity check and/or update may be performed from time to time, or periodically.

As shown, the in-cloud system of the present invention provides protection to IoT devices from damages resulting from malicious code. The system of the present invention provides such a protection in cases where the classical approach is either inapplicable (for example due to lack of sufficient processing power), or impractical (for example, due to the relatively high costs involved in providing of such protection).

While some embodiments of the invention have been described by way of illustration, it will be apparent that the invention can be carried into practice with many modifications, variations and adaptations, and with the use of numerous equivalents or alternative solutions that are within the scope of persons skilled in the art, without departing from the spirit of the invention or exceeding the scope of the claims.

CLAIMS

1. A system for protecting IoT devices from malicious code, which comprises:
   a. a memory extracting module at each of said IoT devices, for extracting a copy of at least a portion of the memory content from the IoT device, and sending the same to an in-cloud server; and
   b. an in-cloude server for receiving said memory content, and performing an integrity check for a possible existance of malicious code within said memory content.

2. A system according to claim 1, wherein said in-cloud server performs one or more of the following:
   a. analysis of the memory to find malware using static analysis methods;
   b. analysis of the memory to find malicious behavior using behavioral and heuristics methods;
   c. reconstruction of the state of an OS of the IoT to determine and report important structural elements;
   d. check of the integrity of the OS and its memory to possibly find fault in the integrity or hidden processes;
   e. a cross-view check on resources to find rootkits and hidden operations; and
   f. a cross-view check and validation of memory contents of plurality of IoT devices.

3. A system according to claim 1, wherein following said integrity check, said in-cloud server performs one or more of the following:
   (1) logging of the results;
   (2) reporting the results, raising a warning or an alert in a case of detection of an unexpected code or behaveior; or
   (3) communicating and responding to an IoT request.

4. A system according to claim 1, wherein the memory, a copy of which is sent to the in-cloud server, is either a persistent memory or a non-persistent memory.

5. A system according to claim 1, wherein said memory extraction module is embedded within a kernel of a respective operating system of the IoT device.

6. A system according to claim 1, wherein said memory extraction module is positioned within a trusted layer at the IoT device.

7. A system according to claim 1, wherein said memory extraction module is positioned within a Trusted Execution Environment at the processor of the IoT device.

1/1

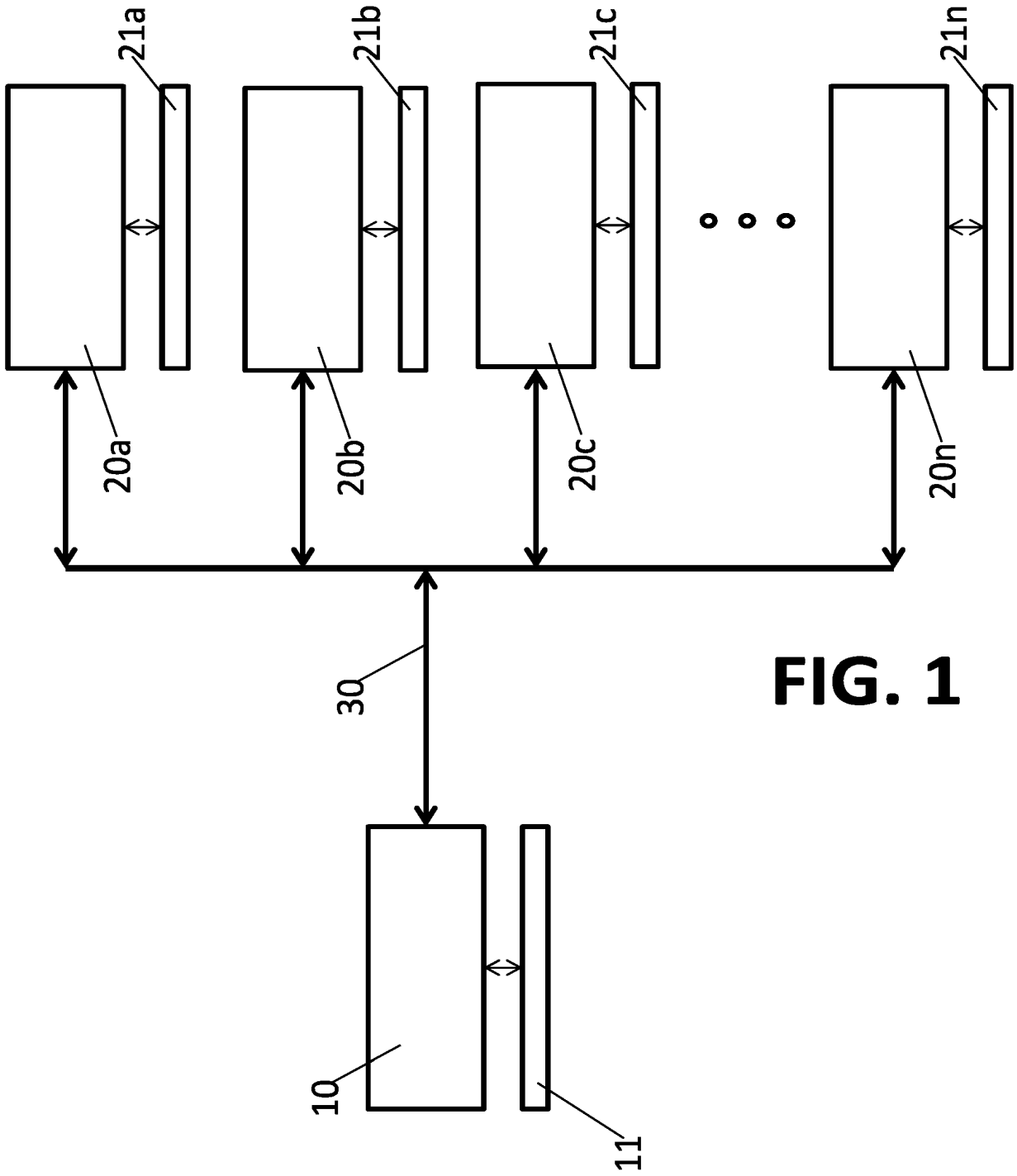


FIG. 1

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL2017/050277

A. CLASSIFICATION OF SUBJECT MATTER
IPC (2017.01) G06F 21/56, G06F 21/60, G06F 21/55, G06F 21/50

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC (2017.01) G06F 21/56, G06F 21/60, G06F 21/55, G06F 21/50, H04L 29/00, G06F 11/00, G06F 12/14

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See extra sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2007277241 A1 SYMANTEC CORPORATION 29 Nov 2007 (2007/11/29) Abstract, ¶¶ 0008, 0022, 0023, 0061, 0062	1-7
Y	US 2014047544 A1 JAKOBSSON BJORN MARKUS 13 Feb 2014 (2014/02/13) Abstract, ¶¶0024, 0025, 0028, 0029, 0039, 0040, 0046, 0051-0055, 0063, Fig 6	1-7
A	US 8904525 B1 Hodgman et al. 02 Dec 2014 (2014/12/02) Entire Document	1-7
A	US 2013227636 A1 APPTHORITY INC 29 Aug 2013 (2013/08/29) Entire Document	1-7
A	US 2013111547 A1 SCARGO INC 02 May 2013 (2013/05/02) Entire Document	1-7

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 Jun 2017	14 Jun 2017
Name and mailing address of the ISA: Israel Patent Office Technology Park, Bldg.5, Malcha, Jerusalem, 9695101, Israel Facsimile No. 972-2-5651616	Authorized officer COPPENHAGEN Uri Telephone No. 972-2-5657811

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IL2017/050277

Patent document cited search report	Publication date	Patent family member(s)	Publication Date
US 2007277241 A1	29 Nov 2007	US 2007277241 A1	29 Nov 2007
		US 7870394 B2	11 Jan 2011
US 2014047544 A1	13 Feb 2014	US 2014047544 A1	13 Feb 2014
		US 9411955 B2	09 Aug 2016
US 8904525 B1	02 Dec 2014	US 8904525 B1	02 Dec 2014
US 2013227636 A1	29 Aug 2013	US 2013227636 A1	29 Aug 2013
		US 8918881 B2	23 Dec 2014
		US 2015143455 A1	21 May 2015
		US 9438631 B2	06 Sep 2016
		WO 2013126259 A1	29 Aug 2013
US 2013111547 A1	02 May 2013	US 2013111547 A1	02 May 2013
		US 9223978 B2	29 Dec 2015
		US 2015350237 A1	03 Dec 2015
		US 9460285 B2	04 Oct 2016
		US 2016373486 A1	22 Dec 2016
		WO 2013063474 A1	02 May 2013

B. FIELDS SEARCHED:

* Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Databases consulted: Esp@cenet, Google Patents
Search terms used: malicious code, malware, internet of things device, iot device, device, cloud, network, extract, copy, audit, check, verify, screen, scan, monitor, memory, cache, data, integrity

PATENT
Attorney Docket No. 15635.0008-00000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Avi SHUA	)	Group Art Unit: OPAP
	)	
Application No.: 17/657,981	)	Examiner: CENTRAL, DOCKET
	)	
Filed: April 5, 2022	)	Confirmation No.: 2397
	)	
For: FORWARD AND REARWARD FACING ATTACK VECTOR VISUALIZATION	)	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. § 1.97(b)

Pursuant to 37 C.F.R. §§ 1.56 and 1.97(b), Applicant brings to the attention of the Examiner the listed document on the attached Form PTO/SB/08. This Information Disclosure Statement is being filed before the mailing date of a first Office Action on the merits for the above-referenced application.

Copies of the foreign and non-patent literature documents are attached.

Applicant respectfully requests that the Examiner consider the listed documents on the attached Form PTO/SB/08 and indicate that they were considered by making appropriate notations on the attached form.

This submission does not represent that a search has been made or that no better art exists and does not constitute an admission that each or all of the listed documents are material or constitute "prior art." If the Examiner applies any of the documents as prior art against any claim in the application and Applicant determines that the cited documents do not constitute "prior art" under United States law, Applicant reserves the right to present to the U.S. Patent and Trademark Office the relevant facts and law regarding the appropriate status of such documents.

Applicant further reserves the right to take appropriate action to establish the patentability of the disclosed invention over the listed documents, should one or more of the documents be applied against the claims of the present application.

If there is any fee due in connection with the filing of this Statement, please charge the fee to deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: September 7, 2022

By: /Christopher C. Johns/
Christopher C. Johns
Reg. No. 68,664
(202) 408-4000

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 March 2019 (14.03.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/051507 A1

(51) International Patent Classification:
G06F 21/56 (2013.01) *G06F 21/55* (2013.01)
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/US2018/050528

(22) International Filing Date:
11 September 2018 (11.09.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/557,132 11 September 2017 (11.09.2017) US

(71) Applicant: **CARBON BLACK, INC.** [US/US]; 1100 Winter Street, Waltham, MA 02451 (US).

(72) Inventors: **KRAEMER, Jeffrey Albin**; 27 Kenilworth Road, Wellesley, MA 02482 (US). **MALINOWSKI, Adam Karol**; 46 Mary Street, Apt. 1, Arlington, MA 02474 (US).

(74) Agent: **LEHRER, Joel E.** et al.; Goodwin Procter LLP, 100 Northern Avenue, Boston, MA 02210 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) **Title:** METHODS FOR BEHAVIORAL DETECTION AND PREVENTION OF CYBERATTACKS, AND RELATED APPARATUS AND TECHNIQUES

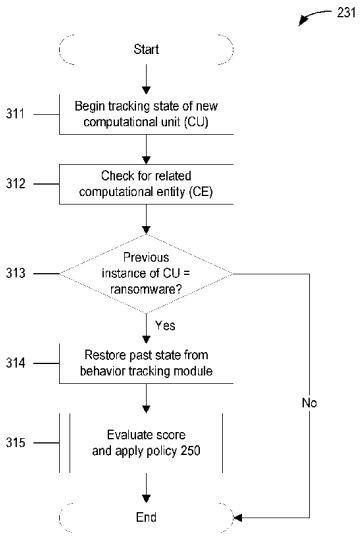


FIGURE 3A

(57) **Abstract:** A security engine may use event-stream processing and behavioral techniques to detect ransomware. The engine may detect process behavior associated with encrypting a file, encrypting a storage device, or disabling a backup file, and may assign a ransomware category to the process based thereon. The engine may initiate protection actions to protect system resources from the process, which may continue to execute. The engine may monitor the process for specific behavior corresponding to its ransomware category. Based on the extent to which such specific behavior is detected, the engine may determine that the process is not ransomware, assign a ransomware subcategory to the process, or adjust the process's threat score. Monitoring of the process may continue, and the threat score may be updated based on the process's behavior. If the threat score exceeds a threshold corresponding to the ransomware category (or subcategory), a corresponding policy action may be initiated.

WO 2019/051507 A1

METHODS FOR BEHAVIORAL DETECTION AND PREVENTION OF CYBERATTACKS, AND RELATED APPARATUS AND TECHNIQUES

CROSS-REFERENCE TO RELATED APPLICATION(S)

This application claims priority to and the benefit of U.S. Provisional Patent Application Serial Number 62/557,132, titled "Methods for Behavioral Detection and Prevention of Cyberattacks, and Related Apparatus and Methods" and filed under Attorney Docket No. BIT-015PR on September 11, 2018, which is hereby incorporated by reference herein in its entirety.

FIELD OF INVENTION

The present disclosure relates generally to computer security systems and techniques. Some embodiments described herein relate specifically to detection and prevention of ransomware-based attacks on endpoint devices (e.g., desktops, laptops, terminals, servers, embedded systems, etc.).

BACKGROUND

As the Internet and other networked computer systems become increasingly integrated into public activities (e.g., management and operation of governmental organizations) and private activities (e.g., personal activities, management and operation of households and businesses, etc.), cybersecurity threats pose an increasingly significant threat to such pursuits. Some examples of cybersecurity threats include malicious software ("malware") and ransomware (collectively, "threatware"). Malware generally operates to disrupt operation of computer systems (e.g., by taking control of computational resources and using those resources for unauthorized purposes, by disabling individual computers or entire networks, by damaging or otherwise sabotaging system components, etc.) and/or to steal resources from computer systems (e.g., by gathering sensitive data). Malware can be deployed in many forms, including computer viruses, worms, trojan horses, spyware, adware, scareware, keystroke loggers, rootkits, bots, crimeware, phishing scams, etc.

Computer users devote significant resources to detecting cybersecurity threats and preventing such threats from disrupting the operations of their computer systems or stealing their computer-based resources. Conventional cybersecurity engines have relied extensively on static, signature-based techniques for detecting malware. In general, static, signature-based malware detection involves obtaining a copy of a file that is known to contain threatware, analyzing the static features of the file (e.g., the sequence of bytes contained in the file) to extract a static signature that is characteristic of the threatware, and adding the threatware's static signature to a

database (often referred to as a "blacklist") of known cybersecurity threats. When a user attempts to access (e.g., download, open, or execute) a file, the cybersecurity engine scans the file and extracts the file's static signature. If the file's static signature matches a signature on the blacklist, the cybersecurity engine detects the presence of a threat and intervenes to prevent the threatware from executing (e.g., by quarantining or deleting the file).

Static, signature-based techniques for detecting threatware are generally useful for quickly detecting known threats. However, these techniques can generally be circumvented by new threats that are not yet blacklisted (e.g., zero-day threatware or next-generation threatware) or by threatware that modifies itself to avoid matching a static signature on the blacklist (e.g., oligomorphic, polymorphic, or metamorphic threatware).

Some cybersecurity engines rely on behavior-based techniques for detecting cybersecurity threats. In general, behavior-based threat detection involves monitoring the execution of a computational unit (e.g., a thread, a process, an application, etc.) and identifying suspicious features of the computational unit's execution (e.g., suspicious behaviors, which may include unpacking code, modifying the host file, logging keystrokes, etc.). When suspicious behaviors are identified, the cybersecurity engine intervenes to protect the computer system (e.g., by terminating or quarantining the computational unit) and assess the threat (e.g., by initiating a forensic investigation of the computational unit, etc.). Behavior-based threatware detection techniques can protect computer systems from certain types of threats (for example, zero-day threatware and self-modifying threatware) that are difficult to detect using blacklists of static signatures.

The spectrum of potential threats ("threatscape") posed by threatening software (e.g., malware, ransomware, etc.) continues to evolve rapidly. Ransomware, for example, has quickly evolved to become one of the most threatening attack methods used by modern cybercriminals and one of the most significant cybersecurity challenges facing organizations that rely on enterprise computing. "Ransomware" generally refers to a type of software that threatens to publish or perpetually block access to a computer user's data unless the user pays a ransom to an attacker. Access to the user's data is generally blocked by encrypting the user's files or storage devices without providing the decryption key to the user.

SUMMARY OF THE INVENTION

Behavior-based threatware detection techniques can protect computer systems from certain types of threatware (for example, zero-day threatware and self-modifying threatware) that are difficult to detect using blacklists of static signatures. However, behavior-based threatware detection techniques can place significant demands on the host computer system's resources, and

can therefore substantially degrade the host computer system's performance. For example, a cybersecurity engine that performs behavior-based threatware detection may track a substantial portion of the behavior of a substantial fraction of the computational units executing on a computer system (e.g., by monitoring the execution of the computational units, detecting behaviors of interest, and/or generating tracking data indicative of such behaviors). Such tracking can consume a significant portion of the computer system's computing resources (e.g., processor cycles, storage space, communication bandwidth, etc.).

Another problem associated with behavior-based threatware detection techniques relates to a tradeoff between false negative outcomes (e.g., erroneously classifying threatware as benign software) and false positive outcomes (e.g., erroneously classifying benign software as threatware). A cybersecurity engine that produces too many false negative outcomes may fail to detect threatware, rendering the engine ineffective and exposing the computer system to dangerous attacks. On the other hand, an engine that produces too many false positive outcomes may identify legitimate software as suspicious and initiate unnecessary interventions (e.g., forensic investigations) that disrupt the legitimate activities of a user or an organization. Initiating unnecessary interventions also wastes resources, including computing resources, energy resources (e.g., electrical power), human resources (e.g., the time and attention of security experts), and others.

Thus, techniques are needed for (1) reducing the amount of computing resources used for behavioral detection of ransomware, and/or (2) reducing the false positive rate of behavioral ransomware detection techniques without increasing (or without significantly increasing) the false negative rate of ransomware detection. The inventors have recognized and appreciated that the aforementioned improvement(s) in the efficiency of behavioral ransomware detection can be achieved using the following, adaptive technique. Initially, a security engine may monitor a computational entity for behavior associated with encrypting a file, encrypting a storage device, and/or disabling a backup file. If such behavior is detected, the security engine may assign a potential ransomware category to the computational entity based on the detected behaviors. Optionally, after assigning a potential ransomware category, the security engine may initiate protection actions to protect storage resources of the computer system from potentially malicious behaviors of the computational entity, which may continue to execute. The security engine may then monitor the computational entity for specific behavior corresponding to the assigned ransomware category. Based on the extent to which such specific behavior is (or is not) detected, the security engine may determine that the computational entity is not ransomware, or may assign a potential ransomware subcategory and a ransomware score to the computational entity. In the latter case, monitoring of the computational entity for behavior specific to the

assigned ransomware category (and/or subcategory) may continue, and the ransomware score may be updated from time to time based on the extent to which such behavior is (or is not) detected. If the ransomware score exceeds a threshold corresponding to the ransomware category (and/or subcategory), a mitigation action associated with the ransomware subcategory may be initiated.

In this way, the tracking of the computational entity's behavior and the interpretation of the tracked behavior (e.g., the determination as to whether the software is ransomware or not) may both be tailored based on what is already known about the computational entity, thereby achieving lower resource utilization and/or a lower rate of false positives than conventional techniques.

In general, one innovative aspect of the subject matter described in this specification can be embodied in a computer-implemented method for ransomware detection, including: based on first events initiated by a computational entity in a computer system, detecting one or more instances of the computational entity engaging in one or more first behaviors associated with encrypting a file, encrypting a storage device, and/or disabling a backup of at least a portion of the computer system; assigning at least one potential ransomware category to the computational entity based on the detected instances of the first behaviors; and after assigning the potential ransomware category to the computational entity, initiating one or more protection actions to protect one or more storage resources against one or more modification operations, and based on second events initiated by the computational entity, detecting one or more instances of the computational entity engaging in one or more second behaviors corresponding to the potential ransomware category.

Other embodiments of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the method. A system of one or more computers can be configured to perform particular actions by virtue of having software, firmware, hardware, or a combination of them installed on the system (e.g., instructions stored in one or more storage devices) that in operation causes or cause the system to perform the actions. One or more computer programs can be configured to perform particular actions by virtue of including instructions that, when executed by data processing apparatus, cause the apparatus to perform the actions.

The foregoing and other embodiments can each optionally include one or more of the following features, alone or in combination. In some embodiments, the first behaviors associated with encrypting a file include enumerating directories of a file system and/or modifying a private file, the first behaviors associated with encrypting a storage device include modifying the storage device, and the first behaviors associated with disabling a backup of at

least a portion of the computer system include disabling and/or deleting a backup file, wherein the backup file stores backup data for a plurality of files and/or settings of the computer system.

In some embodiments, detecting the instance of the computational entity enumerating directories of a storage device includes detecting an enumeration operation initiated by the computational entity on a specified first directory, wherein the specified first directory is (1) a top-level directory of the storage device, (2) a user directory in a file system of the storage device, or (3) an immediate subdirectory of a user directory in the file system of the storage device. In some embodiments, a second directory is nested within the first directory, and detecting the instance of the computational entity enumerating directories of the storage device further includes: detecting an enumeration operation initiated by the computational entity on the second directory; and detecting a file modification operation initiated by the computational entity, wherein the file modification operation targets a particular file in the second directory. In some embodiments, detecting the instance of the computational entity enumerating directories of the storage device further includes: for each intermediate directory in a nesting hierarchy between the first directory and the second directory, detecting an enumeration operation initiated by the computational entity on the intermediate directory. In some embodiments, the top-level directory of the storage device is a mount point of the storage device. In some embodiments, the storage device is an external storage device or a network storage device.

In some embodiments, detecting the instance of the computational entity modifying a storage device includes detecting the computational entity (1) bypassing an operating system of the computer system when accessing the storage device, (2) writing to one or more specified regions of the storage device, and/or (3) modifying one or more file system structures of the storage device. In some embodiments, the specified regions of the storage device include a master boot record (MBR) sector of the storage device and/or a partition table of the storage device. In some embodiments, the file system structures of the storage device include a file system structure pointer and/or an inode. In some embodiments, detecting the instance of the computational entity modifying a private file includes detecting a file modification operation initiated by the computational entity, wherein the file modification operation targets a particular file that is private to a different computational entity.

In some embodiments, assigning the potential ransomware category to the computational entity based on the detected instances of the first behaviors includes: assigning a first category to the computational entity if the detected instances include an instance of the first behavior associated with encrypting a file; assigning a second category to the computational entity if the detected instances include an instance of the first behavior associated with encrypting a storage device; and assigning a third category to the computational entity if the detected instances

include an instance of the first behavior associated with disabling a backup of at least a portion of the computer system. In some embodiments, the potential ransomware category assigned to the computational entity is the first category, the modification operations include one or more file modification operations targeting one or more files, and the protection actions include intercepting the file modification operations and creating backup copies of the files before permitting the file modification operations to modify the files. In some embodiments, permitting the file modification operations to modify the files includes permitting the file modification operations to encrypt the files.

In some embodiments, the potential ransomware category assigned to the computational entity is the second category, the modification operations include one or more storage device modification operations targeting one or more storage devices, and the protection actions include preventing the storage device modification operations from executing. In some embodiments, preventing the storage device modification operations from executing includes preventing the storage device modification operations from (1) bypassing an operating system of the computer system when accessing a storage device, (2) writing to one or more specified regions of the storage device, and/or (3) modifying one or more file system structures of the storage device. In some embodiments, the specified regions of the storage device include a master boot record (MBR) sector of the storage device and/or a partition table of the storage device. In some embodiments, the file system structures of the storage device include a file system structure pointer and/or an inode.

In some embodiments, the potential ransomware category assigned to the computational entity is the third category, the modification operations include one or more file modification operations targeting one or more backup files, and the protection actions include intercepting the file modification operations and creating backup copies of the backup files before permitting the file modification operations to modify the backup files.

The actions of the method may further include assigning at least one potential ransomware subcategory to the computational entity based on the detected instances of the second behaviors. In some embodiments, the potential ransomware category assigned to the computational entity is the first category, and the second behaviors corresponding to the first category include a second behavior associated with encrypting files in-place, a second behavior associated with encrypting files out-of-place, a second behavior associated with encrypting files using a single computational unit, and/or a second behavior associated with encrypting files using a plurality of related computational units. In some embodiments, the second behavior associated with encrypting files in-place includes modifying a file not created by the computational entity. In some embodiments, the second behavior associated with encrypting

files out-of-place includes creating a copy of a file not created by the computational entity, modifying the copy, and deleting the file not created by the computational entity. In some embodiments, the potential ransomware category assigned to the computational entity is the second category, and the second behaviors corresponding to the second category include (1) attempting to modify a master boot record (MBR) of the storage device wherein the storage device is a bootable storage device, and/or (2) attempting to modify a file system structure of the storage device wherein the storage device is a non-bootable storage device.

The actions of the method may further include assigning a score to the computational entity based on the assigned potential ransomware category, the assigned potential ransomware subcategory, and/or the detected instances of the second behaviors. In some embodiments, assigning the score to the computational entity includes setting the score to an initial value based on the assigned potential ransomware category. In some embodiments, the potential ransomware category assigned to the computational entity is the first category, detecting the computational entity engaging in the instance of the second behavior includes detecting an increase in a number of files modified by the computational entity, an increase in a number of distinct file type classes associated with files modified by the computational entity, and/or an increase in a rate of file modification by the computational entity, and assigning the score to the computational entity includes increasing the score based on the increased number of modified files, the increased number of distinct file type classes, and/or the increased rate of file modification.

In some embodiments, the potential ransomware category assigned to the computational entity is the first category, the second behavior includes the computational entity modifying a name of a file, modifying a header of a file, and/or modifying a file such that a file header and a file extension of the modified file are inconsistent with each other, and assigning the score to the computational entity includes increasing the score based on the detected instances of the second behaviors. In some embodiments, the potential ransomware category assigned to the computational entity is the first category and the potential ransomware subcategory assigned to the computational entity indicates that the computational entity performs in-place encryption, the second behavior includes modifying a particular amount of data in a plurality of files, modifying data at a particular offset in a plurality of files, and/or writing particular data to a plurality of files, and assigning the score to the computational entity includes increasing the score based on the detected instances of the second behaviors.

In some embodiments, the potential ransomware category assigned to the computational entity is the first category and the potential ransomware subcategory assigned to the computational entity indicates that the computational entity performs out-of-place encryption,

the second behavior includes specifying file extensions of a plurality of files such that the specified file extensions are all identical to each other or all different from each other, and assigning the score to the computational entity includes increasing the score based on the detected instances of the second behaviors. In some embodiments, the second behavior includes persistently scheduling the computational entity, creating a network connection, querying the computer system for system information, exhibiting suspicious provenance, obfuscating the computational entity, and/or parallelizing the computational entity, and assigning the score to the computational entity includes increasing the score based on the detected instances of the second behaviors.

In some embodiments, the potential ransomware category assigned to the computational entity is the first category, the second behavior includes modifying files of a sanctioned file type, modifying files in a temporary file storage area, and/or modifying private files of the computational entity, and assigning the score to the computational entity includes decreasing the score based on the detected instances of the second behaviors. In some embodiments, determining whether the assigned score exceeds a threshold score associated with the assigned potential ransomware subcategory, and if so, initiating a mitigation action associated with the ransomware subcategory. In some embodiments, the mitigation action includes terminating execution of the computational entity, preventing the computational entity from accessing a communication network, preventing the computational entity from modifying the storage resources, and/or replacing one or more files modified by the computational entity with respective backup files.

In some embodiments, the computational entity includes a single computational unit or a plurality of related computational units, wherein each computational unit is a thread, a process, or an application. In some embodiments, the modification operations are initiated by the computational entity.

In general, another innovative aspect of the subject matter described in this specification can be embodied in a computer-implemented method for ransomware detection, including: based on first events initiated by a computational entity in a computer system, detecting one or more instances of the computational entity engaging in one or more first behaviors associated with encrypting a file, encrypting a storage device, and/or disabling a backup of at least a portion of the computer system; assigning at least one potential ransomware category to the computational entity based on the detected instances of the first behaviors; and after assigning the potential ransomware category to the computational entity, based on second events initiated by the computational entity, detecting one or more instances of the computational entity engaging in one or more second behaviors corresponding to the potential ransomware category, assigning at

least one potential ransomware subcategory to the computational entity based on the detected instances of the second behaviors, assigning a score to the computational entity based on the assigned potential ransomware category, the assigned potential ransomware subcategory, and/or the detected instances of the second behaviors, and determining whether the assigned score exceeds a threshold score associated with the assigned potential ransomware subcategory, and if so, initiating a mitigation action associated with the ransomware subcategory.

Other embodiments of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the method. A system of one or more computers can be configured to perform particular actions by virtue of having software, firmware, hardware, or a combination of them installed on the system (e.g., instructions stored in one or more storage devices) that in operation causes or cause the system to perform the actions. One or more computer programs can be configured to perform particular actions by virtue of including instructions that, when executed by data processing apparatus, cause the apparatus to perform the actions.

The foregoing and other embodiments can each optionally include one or more of the following features, alone or in combination. In some embodiments, each of the first behaviors is selected from a group including enumerating directories of a storage device, modifying a private file, modifying a storage device, and disabling a backup of at least a portion of the computer system.

The actions of the method may further include, after assigning the potential ransomware category to the computational entity, initiating one or more protection actions to protect one or more storage resources against one or more modification operations. In some embodiments, the modification operations are initiated by the computational entity. In some embodiments, assigning the potential ransomware category to the computational entity based on the detected instances of the first behaviors includes: assigning a first category to the computational entity if the detected instances include an instance of the first behavior associated with encrypting a file; assigning a second category to the computational entity if the detected instances include an instance the first behavior associated with encrypting a storage device; and assigning a third category to the computational entity if the detected instances include an instance of the first behavior associated with disabling a backup of at least a portion of the computer system.

In some embodiments, the potential ransomware category assigned to the computational entity is the first category, the modification operations include one or more file modification operations targeting one or more files, and the protection actions include intercepting the file modification operations and creating backup copies of the files before permitting the file modification operations to modify the files. In some embodiments, the potential ransomware

category assigned to the computational entity is the second category, the modification operations include one or more storage device modification operations targeting one or more storage devices, and the protection actions include preventing the storage device modification operations from executing. In some embodiments, the potential ransomware category assigned to the computational entity is the third category, the modification operations include one or more file modification operations targeting one or more backup files, and the protection actions include intercepting the file modification operations and creating backup copies of the backup files before permitting the file modification operations to modify the backup files. In some embodiments, the mitigation action includes terminating execution of the computational entity, preventing the computational entity from accessing a communication network, preventing the computational entity from modifying one or more storage resources of the computer system, and/or replacing one or more files modified by the computational entity with respective backup files.

Particular embodiments of the subject matter described in this specification can be implemented so as to realize one or more of the following advantages. In some embodiments, a behavioral security engine leverages event-stream processing to update (e.g., continually update) risk profiles of computational entities based on a stream of computer system activity. When one or more potentially threatening events are detected (e.g., when a cluster of multiple potentially threatening events is detected), the behavioral security engine defends against (e.g., blocks, prevents, or mitigates) an attack by the potential threatware (e.g., malware or ransomware). The threatware can be file-based or fileless.

In some embodiments, a behavioral security engine detects fast-evolving threats (e.g., ransomware threats, malware threats, etc.), including fileless attacks and unknown (e.g., zero-day) attacks, and protects computer systems (e.g., endpoints) from such threats. In some embodiments, a behavioral security engine detects streaming ransomware, defends against streaming ransomware, and/or prevents streaming ransomware attacks. In some embodiments, a behavioral security engine detects and/or prevents ransomware attacks, even if the attack uses an unknown file (e.g., an unknown executable file) or no file at all. In some embodiments, a behavioral security engine is able to detect and/or prevent ransomware attacks even if the security engine is offline (e.g., disconnected from the Internet, from the cloud, from the enterprise network, etc.). In some embodiments, a behavioral security engine detects and/or prevents ransomware attacks with little or no performance impact on the computer systems protected by the security engine. In some embodiments, a behavioral security engine detects and/or prevents ransomware attacks using little or no additional security resources and/or information technology resources beyond those already allocated to cybersecurity applications.

In some embodiments, a behavioral security engine protects files and/or boot sectors from encryption by ransomware.

In some embodiments, a behavioral security engine detects ransomware attacks with a low rate of false negative detection events and a low rate of false positive detection events. In some embodiments, the behavioral security engine functions as a complete, non-disruptive prevention solution effective against data loss resulting from techniques used by existing and emerging ransomware attacks, including "zero-day" attacks. In some embodiments, a behavioral security engine provides protection against ransomware-induced data loss on two or more levels, including: (1) file-system level and (2) raw storage device level. At the file-system level, some embodiments of the security engine can protect user data files (e.g., user documents or media files), application data, and system files. The scope of file-system level protection can span across local file systems, external drives, network shares, and backups. At the raw storage device level, some embodiments of the security engine can protect system boot records, partition tables, entire physical devices, and file-system structures.

The details of one or more embodiments of the subject matter described in this specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

The foregoing Summary, including the description of some embodiments, motivations therefor, and/or advantages thereof, is intended to assist the reader in understanding the present disclosure, and does not in any way limit the scope of any of the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Certain advantages of some embodiments may be understood by referring to the following description taken in conjunction with the accompanying drawings. In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating principles of some embodiments of the invention.

FIG. 1 is a block diagram of a behavioral security engine, according to some embodiments.

FIG. 2 is a flowchart of a ransomware detection method, according to some embodiments.

FIG. 3A is a flowchart of a method for handling an event associated with creating a new computational unit, according to some embodiments.

FIG. 3B is a flowchart of a method for handling an event associated with accessing a raw storage device, according to some embodiments.

FIG. 3C is a flowchart of a method for handling an event associated with accessing a backup file, according to some embodiments.

FIG. 3D is a flowchart of a method for handling an event associated with enumerating a directory, according to some embodiments.

FIG. 3E is a flowchart of a method for handling an event associated with accessing a file, according to some embodiments.

FIG. 3F is a flowchart of a method for handling an event associated with accessing a network, according to some embodiments.

FIG. 3G is a flowchart of a method for handling an event associated with performing an application programming interface (API) call, according to some embodiments.

FIG. 3H is a flowchart of a method for evaluating the threat score of a computational entity and applying a security policy to the computational entity, according to some embodiments.

FIG. 4 is a flowchart of another ransomware detection method, according to some embodiments.

FIG. 5 is a block diagram of an implementation of a computer system.

DETAILED DESCRIPTION

Threat Detection and Response

Referring to FIG. 1, a behavioral security engine 100 may include a tracking module 110, a heuristics module 120, a behavior scoring module 130, a configuration module 140, a policy module 150, and a protection and mitigation module 160. The tracking module 110 may include an event monitoring module 112 and a behavior tracking module 114. The heuristics module 120 may include a categorization module 122, a subcategorization module 124, a score increasing module 126, and a score decreasing module 128.

The behavioral security engine 100 may be a component of a cybersecurity engine, and its modules may cooperate detect potential threats from threatware, including file-based threats and/or fileless (e.g., streaming) threatware threats, by monitoring and analyzing events (e.g., a stream of events) on a computer system (e.g., an endpoint computer system). The behavioral security engine's components may also cooperate to respond to detected threats. Some embodiments of the components of the behavior security engine 100 are described in further detail below.

In some embodiments, the behavioral security engine 100 may perform the following actions: (1) the event monitoring module 112 may track low-level events in a computer system; (2) the behavior tracking module 114 may map events to states and track them; (3) the categorization module 122 may assign a threat category (e.g., a ransomware category) to a potential threat and update the threat category as more events in the computer system are tracked; (4) the subcategorization module 124 may assign a threat subcategory (e.g., a ransomware category) to a potential threat and update the threat subcategory as more events in the computer system are tracked; (5) the behavior scoring module 130 may assign a threat score to the potential threat adjust the t threat score as more events in the computer system are tracked; and (6) if the threat score exceeds a threshold associated with the threat category (or subcategory) assigned to the potential threat, the policy module 150 and protection and mitigation module 160 may cooperate to perform a threat response action specified by the configuration module 140 for the detected type of threat. The configuration module 140 may be configured (e.g., by an administrator) with a security policy that defines actions to be taken upon detection of particular categories of threats (e.g., particular categories of ransomware threats).

Tracking:

The tracking module 110 may monitor events occurring in the computer system and track the behavior of computational entities in the computer system based on the monitored events. A computational entity may be, for example, a thread, a process, an application, or a related set of two or more threads, processes, and/or applications. The tracking module 110 may include an event monitoring module 112, which may monitor low-level events in the computer system. In some embodiments, the event monitoring module 112 tracks events from one or more layers of the operating system ("OS") of a computer system in real-time, within the context of a computational entity, and analyzes the monitored events. The types of events monitored may include, without limitation: (1) computational unit creation events, (2) script and/or library load events, (3) file input/output ("I/O") events (e.g., directory read / enumerate; file read, write, create, delete, rename / move; etc.), (4) network events (e.g., create incoming and/or outgoing network connections, etc.), and (5) system calls and/or calls made by a process to an application programming interface ("API") of the OS. Some examples of API calls that may be monitored include, without limitation, (1) API calls related to searching a filesystem, gathering system information, disabling backups, or mounting a disk partition; (2) API calls characteristic to code injection (e.g., from a malicious actor into ransomware host process); and (3) API calls querying system information.

The tracking module 110 may also include a behavior tracking module 114, which may receive data relating to monitored events from the event monitoring module 112. The behavior tracking module 114 may track the behavior of computational entities and maintain threat profiles for the computational entities. Each computational entity's threat profile may include data indicating one or more profile states of the computational entity. The behavior tracking module 114 may update a computational entity's threat profile (e.g., update the profile states in the threat profile) based on monitored events associated with the computational entity, thereby mapping the events to corresponding states.

In some embodiments, the behavior tracking module 114 augments data corresponding to tracked events with additional static and/or dynamic information including, without limitation, context details and/or state information associated with the event. The behavior tracking module 114 may map events to internally tracked states that include additional metadata (e.g., counters and/or timestamps). Such states may be tracked on different levels, for example, (1) per computational entity performing the action, for example, a process enumerating directories, reading and writing files; (2) per unique application identifier (e.g., sha256 file hash), to track behavior of an application across processes, and (3) on system level (e.g., system states impacting or responding to application behavior).

In some embodiments, the behavior tracking module 114 extends the notion of behavioral state tracking from individual computational units (e.g., individual processes) to sets of related computational units (e.g., multiple-process chains). The behavior tracking module 114 may determine whether suspicious behaviors of one computational unit are likely performed on behalf of another computational unit (e.g., a main process in control). In addition, the behavior tracking module 114 may determine such likelihood, identify the effective actor and the extent of the set of related computational units, and track behavior across the entire set of related computational units, ultimately enabling the behavioral security engine 100 to enforce policies for all computational units in a set responsible for malicious behavior (e.g., ransomware behavior). In addition, the behavior tracking module 114 may use heuristics (described below) to differentiate potentially malicious behaviors from safe behaviors of computational units that are part of the OS, thereby reducing false positive detection of threatware.

Heuristics:

In some embodiments, the heuristics module 120 implements processes (e.g., heuristic processes) for assigning threatware categories (e.g., ransomware categories) to computational entities, assigning threatware subcategories (e.g., ransomware subcategories) to computational entities, and/or adjusting threat scores of computational entities. From time to time (e.g.,

periodically or in response to specific events), the heuristics module 120 may query the behavior tracking module 114 to determine the profile states of a computational entity, and may notify the behavior scoring module 130 when a heuristic that affects a computational entity's category, subcategory, and/or score is triggered (e.g., by a change in the state of the computational entity). In some embodiments, the heuristics module 120 may notify the protection and mitigation module 160 when a heuristic associated with initiation a protection or mitigation action (e.g., initiating a backup process) is triggered. The heuristics module 120 may track the values of heuristic states and categories internally.

Categorization and Subcategorization:

The heuristic module's categorization module 122 may implement categorization heuristics suitable for assigning threat categories to computational entities based on their behavior. In some embodiments, the categorization module 122 may assign a computational entity the threat category that matches (e.g., most closely matches) the computational entity's current set of profile states.

In some embodiments, the categorization module 122 uses the categorization heuristics to detect ransomware-specific behaviors, at least one of which is generally characteristic to any type of ransomware. These categorization heuristics may act as first-level filters and triggers. These first-level filters may facilitate the maintenance of a low false-positive ransomware detection rate, and/or may narrow the monitoring scope of the tracking module 110, thereby lowering overall impact on system performance. Suspicious computational entities may be assigned ransomware categories (for example, the following ransomware categories) based on detection of certain behaviors exhibited by the applications:

Category A: Encrypting files on local, remote (network), or external drives;

Category B: Encrypting raw storage device or parts of it (e.g., master boot record (MBR) sector on boot drive, or entire backup drive); and/or

Category C: Disabling system backups.

A suspicious computational entity that exhibits behaviors associated with two or ransomware categories may be assigned to all of the corresponding ransomware categories at the same time.

A computational entity may be assigned to ransomware Category A if the entity exhibits behaviors associated with encrypting files, including, without limitation, (1) enumerating a storage device or a directory of a file system, and/or (2) modifying private files.

Storage Device and/or Directory Enumeration: Heuristics for detecting enumeration of a storage device (e.g., disk) or directory may take advantage of the observation that ransomware generally has a starting point during a search to locate data files. The starting point

generally includes, but is not limited to, well-known top-level directories (e.g., the driver letter root, user folder, user's documents and desktop, etc.). In some embodiments, several top level directories on the endpoint device are monitored for enumeration, including the root of each mount point (internal, external drives, network shares, backups), user folders, and several well-known subfolders under the user folders. Additional high-value directories may be monitored (e.g., high-value directories that could be derived by ransomware in some other way than using recursive enumeration, such as by querying the system state for mount points or user names). For example, network share mounts that often use directory structure conventions may be monitored.

In some embodiments, descendant directories deeper in the directory tree are not monitored for the starting point directory enumeration. This technique may help the behavioral security engine 100 maintain a low false positive rate, without increasing false negatives, because real ransomware generally does not possess a knowledge of target machine directory structure, beyond the starting point directories, without performing the initial discovery.

In some embodiments, after a computational entity enumerates a starting point directory, the descendant directories may be recursively monitored for enumeration to establish whether a continuous recursive directory enumeration occurred all the way down to a directory in which a potential file encryption operation is detected. This heuristic can further reduce false positives in cases in which a legitimate application enumerates one of the known starting point directories and independently, often for other reasons, enumerates a directory known to that application, which may be several levels down from the starting point directory, but without enumerating any directories in between.

On some operating systems, false positive mitigation techniques may be used to rule out directory enumeration by a computational unit while the computational unit is loading and enumerating folders to locate dynamic libraries to be linked. For example, a useful false positive mitigation technique in this case may be to discard events initiated by a computational unit prior to initiating the execution of the computational unit's main method.

Modifying Private Files: Some files are known to be private only to a specific application and are known to be maintained only by that application. Examples may include document files created by an installer of another application. When a computational entity attempts to modify a file that is private to a separate computational entity, that behavior may be indicative of ransomware. (In addition or in the alternative, modification of private files may be used by the subcategorization module 124 to subcategorize suspected malware as "file-encrypting.")

A computational entity may be assigned to ransomware Category B if the entity exhibits behaviors associated with encrypting a raw storage device or a portion thereof, including, without limitation, modifying a raw storage device.

Modifying Raw Device: Examples of raw device modification operations may include attempting to access a raw storage device and write to specific regions thereof (e.g., master boot record (MBR) sector in case of the boot device, partition table or arbitrary region of non-bootable data drive, etc.), bypassing the OS. In case of a computational entity that includes a set of related computational units, the effective actor may be derived by traversing the process hierarchy. To reduce or avoid potential false positives during an OS upgrade, raw device modification operations initiated by computational units that are well known OS components and are not invoked by an external computational unit may be discarded.

Raw device modification heuristics may extend to detection of modification of low-level file system structures (e.g., inodes). For example, ransomware may traverse and encrypt chosen inode data directly, without going through file-system virtualization layer, like most user-space applications. Ransomware could also modify file system structure pointers to hide files from user, without actually encrypting the inode data, thereby achieving a similar end-effect.

In some embodiments, the categorization module 122 may assign separate categories to computational entities that attempt to write to specific regions of the raw device (e.g., the MBR or partition table) and computational entities that attempt to modify low-level file system structures. For example, the categorization module 122 may assign Category B1 to the former computational entities, and Category B2 to the latter computational entities.

A computational entity may be assigned to ransomware Category C if the entity exhibits behaviors associated with disabling a computer system's backup capabilities, including, without limitation, deleting / disabling backup files and/or disabling backup / restore functionality.

Disabling System Backups: Examples of behaviors associated with disabling system backups may include (1) deleting, encrypting, or otherwise modifying backup files, and (2) disabling a computer system's ability to create new backups or restore data and settings from existing backups. In the case of a computational entity that includes a related set of computational units, the effective actor may be determined to derive intent and avoid triggering false positives in legitimate use-cases (e.g., when a user is re-configuring backups).

Since some of these behaviors could be also exhibited by legitimate software, additional heuristic techniques (described below) may be by the categorization module 122 to further increase detection and prevention accuracy and reduce the false positive rate.

When a ransomware category is assigned to a computational entity, the behavior scoring module 130 may use scoring heuristics specific to that category to determine an initial threat

score for the computational entity. Suitable scoring heuristics for computational entities assigned to ransomware Categories A, B, B1, B2, and/or C are described in further detail below.

After a ransomware category is assigned to a computational entity, the subcategorization module 124 may use subcategorization heuristics specific to the assigned category to further classify potential ransomware into subcategories. These heuristic methods may factor in complexity and uncertainty characteristics to detection of different malicious techniques. For example, Category A ransomware may be divided into subcategories based on any suitable criteria, including but not limited to: (1) whether the computational entity encrypts files in-place, out-of-place, or both, (2) whether the computational entity has multiple computational units or a single computational unit, and (3) whether the computational entity utilizes built-in system utilities.

When a ransomware subcategory is assigned to a computational entity, the behavior scoring module 130 may use scoring heuristics specific to that subcategory to adjust the threat score for the computational entity. Suitable scoring heuristics for computational entities assigned to the various ransomware subcategories are described in further detail below.

After a ransomware category is assigned to a computational entity, the categorization module 122 may adjust the category assigned to the computational entity dynamically as more behavior is processed and more is known about the entity. Likewise, after a ransomware subcategory is assigned to a computational entity, the subcategorization module 124 may adjust the subcategory assigned to the computational entity dynamically as more behavior is processed and more is known about the entity.

In general, categorization and subcategorization may allow for better control of prevention sensitivity for a given category or subcategory of ransomware, without affecting scoring of other categories or subcategories.

In some embodiments, one or more of the following heuristic techniques may be used by the categorization module 122 for categorization, by the subcategorization module 124 for subcategorization, and/or by the scoring module 130 for initial scoring:

- Tracking of a computational entity's use of low level IO operations and pathways that can lead to net-resulting file modification.
- Tracking of file modifications initiated by a computational entity's use of, in relation to file IO path, file node attributes and content.
- Tracking of a computational entity's modifications of files in directories and subdirectories that the entity previously enumerated during discovery stage.
- Tracking of number of files modified by the computational entity.

- Filtering file modification events based on file type. Modifications of all files may be accounted for and/or prevented, however, modifications of particular file types may carry more weight than others. For example, data files may be assigned more weight.
- Filtering file modifications based on file location. The system may not track modifications of files located in specific directories (e.g., user folders, on network shares or backup devices, etc.) when the modifications are performed by specific computational entities. For example, events associated with known backup software modifying files on backup store may be filtered out.
- Filtering file modifications based on file location and file type. More weight may be assigned to modification of non-data files on non-local drives (e.g., network drives, external drives, etc.). This method may increase detection sensitivity and reduce data-loss when ransomware attempts to encrypt non-data files on non-local drives. Non-local drives can be particularly vulnerable to ransomware because (1) ransomware can generally encrypt such drives entirely without taking away a mechanism to deliver a ransom message to the user, and (2) network shares and backup drives may be of especially high value to the user.
- Tracking of different types of file modifications, by keeping track of counters and ratios of different IO operations (e.g., reads, writes, creates and/or deletes) on files. These counter values and ratios may be updated regularly (e.g., after every IO operation). This data may be used to determine whether (1) a file of interest is being directly modified (in-place encryption), or (2) a copy of the file is being modified, followed by deletion of the original file (out-of-place encryption). This technique may play a role in assigning Category A ransomware to the corresponding subcategory (e.g., the in-place encrypting subcategory or the out-of-place encrypting subcategory). Once the computational entity is subcategorized, the tracking module 110 may track the file encryption behaviors specific to the subcategory more closely, and may assign a more accurate scoring heuristic and more appropriate threat score thresholds specific to the subcategory.
- Tracking of file IO operation patterns indicating legitimate applications, to lower the false positive rate. If a computational entity exhibits specific sequences or patterns of file IO operations (e.g., read, create, write, and/or delete), the heuristics module 120 may determine that the entity is non-threatening, and the tracking module 110 may filters out IO operations of such entities. For example, software installers may modify a small number of pre-existing files, while also generating a larger number of new files. Such a pattern does not fit the behavioral signature of out-of-place encryption and can therefore be rejected to reduce false positives.

- Tracking of file modification and disk enumeration spread across a related set of computational units. This technique may be used for subcategorization and for tracking the behavior of a computational entity that includes multiple computational units. For example, an entity assigned to a ransomware category (e.g., Category A) may be subcategorized further as a single-process entity or a multi-process entity (independently from being subcategorized as an in-place encrypting entity or an out-of-place encrypting entity) for more accurate scoring and independent threshold assignment. In addition, after the entity is categorized as a multi-process entity, the behavior (e.g., heuristic ransomware behavior) of the entity's computational units may be tracked and attributed to the entire entity. The entity's computational units may include known built-in OS tools, in which case invocation of the tool is treated similarly to invocation of an API call. The effective actor among the entity's computational units may be identified by walking the process hierarchy and identifying the process higher in the hierarchy that is not part of the OS and is not a known verified application. In this way, ransomware behavior may be tracked across processes spawned by the actor process or its descendants. For example, a malicious actor may delegate enumeration or encryption tasks to one or multiple sub-processes. When the entity's threat score reaches the assigned threshold, a prevention policy may be applied to the entire entity to ensure that no dangling processes continue executing sub-tasks after terminating only part of the process chain. Due to higher complexity in detection of multi-process ransomware and the intent not being always clear, scoring heuristics and thresholds may be tuned accordingly to maintain a low false positive rate.
- For detection and categorization of Class B ransomware, the categorization module 122 may determine the region of the storage device being modified and correlate that region with the storage device type. For a boot-type storage device, attempts to modify the master boot record (MBR) region may be highly suspicious. For a non-boot-type storage device (e.g., a backup device), modification of any portion of the raw device may be suspicious. These heuristics take advantage of the observation that ransomware that encrypts data at the raw device level generally still needs to be able to execute some piece of code to display the ransomware message with instructions.

Scoring:

The behavior scoring module 130 may maintain a threat score for each computational entity of interest, and the threat score may be persisted across application invocations. The behavior scoring module 130 may receive score adjustment notifications from the heuristics

module 120, and may adjust the threat scores for the affected computational entities accordingly. When a computational entity's threat score reaches a corresponding threshold associated with a protection or mitigation policy, the behavior scoring module 130 may signal the protection and mitigation module 160 to apply the corresponding protection or mitigation policy.

As described above, the scoring techniques used by the behavior scoring module 130 for a given computational entity may be specific to the ransomware category and/or subcategory assigned to the entity. For example, the scoring algorithm, scoring units, and/or scoring weights (e.g., multipliers) used by the behavior scoring module 130 to calculate a computational entity's threat score may be specific to the entity's ransomware category and/or subcategory. In some embodiments, the behavior scoring module 130 may also adjust the scoring units and/or scoring weights for a computational entity in response to score adjustment notifications received from the heuristics module 120. As another example, the threshold at which a computational entity's threat score triggers application of a protection or mitigation policy may be specific to the entity's ransomware category and/or subcategory.

In some embodiments, the scoring algorithm (e.g., scoring progression), scoring units, scoring weights, and/or thresholds specific to the different threat classes may be configurable for flexibility and tuning. For example, the rate of increase of a threat score may be configurable to facilitate better tuning of the scoring algorithm and reflect how the various score increasing and score decreasing heuristics can influence the score progression, also accounting for different weights. In some cases, a score increasing or decreasing heuristic may adjust the score by a number of units multiplied by the current multiplier assigned to the entity. In some cases, a score increasing or decreasing heuristic may adjust the heuristic multiplier for that entity, thereby allowing the heuristic to affect the score progression when other scoring heuristics trigger.

In some embodiments, the behavior scoring module 130 may receive a notification from the score increasing module 126 of the heuristics module 120 when one or more of the following score-increasing heuristics triggers (e.g., for a computational entity that has already been assigned a potential ransomware category):

- Computational entity modifies multiple files. In some embodiments, the higher the number of files modified by the potential ransomware entity, the greater the increase in the threat score.
- Computational entity modifies files of multiple file type classes, which distinguishes the entity from an entity that modifies only files of a single, specific file type class. For example, an unknown computational entity modifying Office documents, text, images, video, and audio files is unlikely to be a file editor. In some embodiments, for every file

modified by a potential ransomware entity, the tracking module 110 detects and tracks the original file type and assigns it a file type class (e.g., document, image, video, audio, text, etc.). In some embodiments, the number of file type classes the entity modifies is tracked, and when the number of file type classes modified by the entity reaches a specified threshold, the threat score of the entity is increased.

- Detection of file content, name and attribute modification indicating a more definite encryption vs. a non-malicious file content modification, using the following heuristics:
  - Comparison of new file names compared to old file names for files with modified content. File name change patterns could indicate clear intent to encrypt as some ransomware uses this technique as a simple way of communicating to the endpoint user which files have been encrypted,
  - file magic header after modification different than original magic header,
  - file magic header after modification not matching file extension.
  - file magic matching original but modification of every file follows a pattern: similar data size modified, or at similar offset, or similar content written to each file.
- Rate of file modification: application modifies (in-place or not-in-place) many files within a small time window.
- File extension naming patterns for new files created as a result of not-in-place encryption. Once application is sub-categorized as not-in-place encrypting, file extensions of new files created by its process are tracked per every new file created by the process. The method looks for continuity in file extension names generically, rather than only relying on static signatures of extension names commonly used for encrypted files. Typical ransomware is more likely to use either 1) the same arbitrary file extension string for every new encrypted file; often a human readable word used as a hint to the ransomware victim or 2) a different arbitrary file extension string for every new encrypted file, such as a string containing some pseudo-randomness. Potential not-in-place encrypting ransomware that follow either of the 2 these file extension patterns get an additional heuristic score boost, while the ones that break these patterns are not affected by this method.
- Persistence – application configures itself as a service or schedules itself as a recurring job.
- Tracking application behavior across process instances, i.e. handling ransomware that is re-run. If previous instance of the application exhibited ransomware behavior, the behavior is tracked per hash and then inherited by the newer process instance. This

method allows for applying policy sooner on next execution, preventing ransomware from achieving incremental progress as it is being re-executed. On next process execution, new process is either blocked from running, or it is terminated on one of the first file modification attempts. The resulting action depends on accumulation of behavior and confidence during the previous process run. The goal is to prevent ransomware from making incremental progress, while reducing a chance of recurring false positive.

- Application attempts a network connection (such as to connect to potential C2). This heuristic is of more value if the application connects to server before modifying data.
- Application queries unique system information, such as hardware serial number, UUID, MAC address, IP address, hostname.
- Application queries CPU information, such as to detect number of cores and whether the environment is virtualized.
- Application invokes randomization API, such as to generate a unique encryption key.
- Applications invoked cryptographic API.
- Application invokes multiple instances of itself – to achieve parallelization. In addition to increasing the score, the system tracks multiple instances in order to apply policy to all instances of the process, much like in case of generic multi process ransomware chain..
- Application vector – applications originating from email applications, web browsers, chat clients, external or network drives.
- Date/Time when application was dropped before exhibiting ransomware behavior – applications that appeared on endpoint more recently (threshold is configurable), are treated as more suspect.
- Application obfuscates itself by either: running out of unusual folder (not where programs are typically installed), unusual name or extension (such as a document extension), with unusual attributes (such as hidden), by deleting or moving the file the code was loaded from.
- Application links to a known encryption library (dynamically or statically)
- Detection of IPC and data passed between processes involved in a potential multi-process ransomware chain. This method helps determine intent and whether process is performing a specific task on behalf of ancestor process or on its own. For example, if a sub-process is spawned, which then performs disk enumeration, heuristic looks for data being passed back to the potential process in control to establish clear intent and whether this behavior should be attributed to that process, effectively extending the multi-process chain.

- Application reputation: applications that are not yet widely known and appear to be potential 0-day attacks, or applications that are not signed by a trusted and verified authority.
- Application has been previously detected as compromised by one of code injection mechanism.
- Application known to process in memory data content that is more prone to a vulnerability and in-memory exploit, or application that has already undergone buffer overflow.
- Commonly known application exhibiting unusual behavior, outside of established norms, indicating potential compromise, even without earlier detection by the system of the actual compromise vector. This method accounts for new, unknown vulnerabilities in core applications. For example, calculator app communicating on the network.

In some embodiments, the behavior scoring module 130 may receive a notification from the score decreasing module 128 of the heuristics module 120 when one or more of the following score-decreasing heuristics triggers (e.g., for a computational entity that has already been assigned a potential ransomware category):

- Verified application that is a known modifier of files of certain sanctioned type (for example, a known photo editor application is modifying image files). Such sanctioned file modifications are bypassed by the system, while still keeping track of modification of other file types to account for a case of application being compromised.
- Application modifying files in temporary file area - such file modifications are ignored.
- Application modifying its own private folder data files, such as an email application modifying e-mail files in data folder corresponding to the actor process.
- In multi-process chain, a method is used to determine if the effective actor in control is part of the operating system. The method uses a combination of file path, verified certificate and file reputation. If the effective process in control is an operating system component, the behavior is not policy-enforced, but it can be optionally tracked and reported. This technique reduces false positives with built-in backup utilities, during OS upgrade and operations driven by GUI controls, such as re-partitioning.
- While calculating multi-process chain, system takes into account standard operating system services that play a role of "launchers", i.e. execute a process on behalf of another process (e.g., WMI, launchd, etc.). In such case, behavior is conducted across such chain, but policy enforcement excludes those standard services that are long-lived and spawned by the OS, and not by the malicious actor.

In some embodiments, the behavior scoring module 130 uses additional heuristic methods to analyze isolated "interesting" behaviors, and the results of such analysis may be used to increase the score of an entity that has been previously classified as potential ransomware. In turn, behaviors that are characteristic of legitimate applications may decrease the current score or bypass one of other heuristics that would otherwise contribute to a false positive. These methods may be helpful to increase certainty.

In some embodiments, the scoring increment associated with each of the foregoing scoring heuristics is relatively low. In some embodiments, none of the scoring heuristics alone may increase the score decisively enough to reach the assigned threshold and trigger the application of a protection or mitigation policy. However, having more scoring methods in the system may allow for higher thresholds to be set and therefore better separation between a real threat and a false positive without sacrificing threat detection.

Configuration, Policy and Protection / Mitigation:

The configuration module 140 may configure (or permit a user or administrator to configure) various parameters and settings of the behavioral security engine, including but not limited to (1) units and weights of the scoring heuristics, (2) categorization and subcategorization parameters, (3) threshold scores that trigger application of protection or mitigation actions for different ransomware categories and subcategories, and (4) protection and mitigation policies (e.g., the particular policy that is triggered when a computational entity's score reaches a particular threshold).

A protection or mitigation policy may define action(s) to be taken against a malicious entity when the entity's threat score reaches the corresponding threshold. Options for responding to a threat may include, without limitation (1) terminating the malicious entity, (2) isolating (e.g., "sandboxing") the malicious entity (e.g., allowing the entity to run, but without allowing the entity to modify data), or (3) allowing the entity to perform all operations and logging the entity's operations.

The policy module 150 may store policy settings retrieved from the configuration module 140. The policy module 150 may receive a notification from the behavior scoring module 130 when an entity's threat score exceeds and associated threshold. In some embodiments, the policy module 150 determines which protection / mitigation action (e.g., terminate, isolate, or log) to perform based on the policy settings and/or the entity's reputation. In some embodiments, the policy module 150 notifies the protection and mitigation module 160 to perform the selected protection / mitigation action.

The protection and mitigation module 160 may enforce protection and mitigation policies in response to receipt of a corresponding notification from the policy module 150. In some embodiments, the protection and mitigation module 160 has the ability to enforce a policy action by denying completion of a file operation, network operation, and/or process creation operation in-line with the operation being requested. In some embodiments, the protection and mitigation module 160 may communicate with the OS to terminate or isolate a running process.

In some embodiments, the protection and mitigation module 160 may optionally (e.g., using a setting) enable the techniques described below in the Section titled "Data Backup and Recovery" (e.g., after a computational entity has been categorized as potential ransomware). These techniques may serve two main purposes: (1) to reduce any data loss that may occur during the transient state of low threat score, e.g., after low number of files are potentially encrypted and before the policy action is taken, and (2) to allow for additional flexibility in increasing the heuristic thresholds, leading to lower false positives.

As just one example, after a computational entity is categorized as Category A ransomware, the protection and mitigation module may begin backing up all files modified by the entity, in-line with and prior to completion of the entity's attempt to modify the files. If the protection and mitigation module 160 subsequently receives a notification from the policy module to restore the files modified by the entity (e.g., in response to the behavior scoring module 130 indicating that the entity's threat score has exceeded the corresponding threshold), the module 160 may then restore the backed-up files.

Data Backup and Recovery

The techniques described in this Section may be used to enhance a behavioral security engine 100 for endpoint devices (e.g., desktops, laptops, terminals, servers, embedded systems) against data loss (e.g., via ransomware) or attempts to make the underlying operating system inoperative (e.g., via destructive malware). The techniques described in this Section may enhance such systems in one or both of the following ways:

1. Protection against file data loss: Behavioral detection systems are generally characterized by a lag between the initial malicious behavior taking place (e.g., data file encryption or destruction) by a suspect application and the time when policy action is taken (e.g., the malicious application is terminated). During this time window, the system can be prone to allow for some data loss, before action is taken to terminate or restrict the potentially malicious application. For example, to maintain low false positives, a conventional behavioral detection system may choose to gather more evidence and gain sufficient confidence. By

contrast, the techniques described in this Section may protect the file data integrity during the period of lower confidence.

2. Reduction of false positives: Since the technique protects against data loss during the period of low confidence, the technique may allow for additional flexibility in tuning sensitivity until the state of higher confidence is reached, consequently reducing (e.g., minimizing) the false positive rate. When policy action is taken, data files at risk that were modified or deleted during the period of low confidence can be completely restored.

A data backup and recovery module may be incorporated into the behavioral security engine 100 as follows:

- The data backup and recovery module receives notifications about file and directory I/O operations (e.g., create, delete, move / rename, read, write). The notifications may be received in-line with the operations taking place.
- The data backup and recovery module receives notifications from the behavioral security engine 100 on interesting state changes of monitored applications.

The security engine 100 may notify the data backup and recovery module (which may be embedded within or external to the security engine 100) when to start performing backups of files affected by a potentially malicious application. For example, the security engine 100 can send the notification early, when a sample has been pre-categorized as potentially malicious, but before any data loss occurred. From that point onwards, the data backup and recovery module may monitor file operations performed by the application / process (or process chain). The data backup and recovery module may be capable of stalling the original file I/O operation (operation requested by the monitored application), for example write or delete, creating a copy of each file that the monitored application is preparing to modify or delete, and storing the copies in a backup area. The file backup operation can be implemented either directly in the file-filter driver or in a user-space agent, as long as the original operation is stalled until the file backup is complete.

The backup area may be protected by file-filter driver, as such, applications that are not part of the security engine are not allowed to modify it. When the prevention system reaches threshold and applies policy at higher certainty point, the files are available for restoration. The restore functionality can be triggered automatically by the prevention system, or by user action.

The techniques described herein may be applicable to Windows, macOS, Linux, and/or other software platforms. There may be differences in implementation details across platforms (e.g., because some data point inputs associated with specific detection methods may not be available on every platform, and/or because low-level operating system features or services available to ransomware differ slightly across different platforms). For example, some of the

techniques described herein may differ in implementation and/or score weighting on each platform.

Some Embodiments of Ransomware Detection Methods

Referring to FIG. 2, a ransomware detection method 200 may include steps of detecting 210 an event, determining 220 whether the detected event is a trigger event, and if so, handling the event in accordance with its type. For example, if the trigger event is the creation of a new computational unit, the event may be handled by performing a process 231 for handling creation of a new computational unit. If the trigger event is a computational entity accessing a raw storage device, the event may be handled by performing a process 232 for handling an access of a raw storage device. If the trigger event is a computational entity accessing a backup file, the event may be handled by performing a process 233 for handling an access of a backup file. If the trigger event is a computational entity enumerating a directory, the event may be handled by performing a process 234 for handling directory enumeration. If the trigger event is a computational entity accessing a file, the event may be handled by performing a process 235 for handling an access of a file. If the trigger event is a computational entity accessing a network, the event may be handled by performing a process 237 for handling an access of a network. If the trigger event is a computational entity performing an API call, the event may be handled by performing a process 238 for handling an API call.

The foregoing examples of trigger events and corresponding handler processes is non-limiting; other trigger events may be detected and handled.

If the detected event is not a trigger event, the event detection step 210 may be repeated. Likewise, after a trigger is handled, the event detection step 210 may be repeated. Some techniques for detecting events (e.g., process creation and termination events; file read / write / create / delete / move / mmap events; network activity events; API and system call events, etc.) are described above.

Non-limiting embodiments of the handler processes 231-235, 237, and 238 are illustrated in FIGS. 3A-3G, respectively.

Referring to FIG. 3A, a process 231 for handling an event associated with creating a new computational unit may include steps 311-315. In step 311, the tracking module 110 begins tracking the profile state of a new computation unit (CU). In some embodiments, the profile state of the new CU is initialized such that all elements of the profile state (e.g., file IO counters, network state, system call state, threat score, score weights, CU category, CU subcategory, etc.) are initialized to default values. In some embodiments, the tracking module 110 obtains reputation information indicating a reputation of an application A1 that created the new CU, and

stores that reputation information in the CU's state profile. Some examples of suitable techniques for tracking the profile state of a CU during the CU's execution are described above.

In step 312, the tracking module 110 determines whether the new CU is a single-CU computational entity or part of a multi-CU computational entity. If the new CU is part of a multiple-CU computational entity, the new CU is added to the computational entity's state profile, thereby linking the new CU to the computational entity. Some examples of suitable techniques for determining whether a CU is part of a related set of CUs forming a computational entity are described above.

In step 313, the tracking module 110 determines whether a previous instance of a CU corresponding to the application A1 was categorized as ransomware. If not, the process 231 terminates. If so, the tracking module 110 restores (314) the state of the previous instance of the CU in the state profile of the new CU and the behavioral security engine 100 performs (315) a process 250 for evaluating the CU's threat score and applying a corresponding policy, if any.

Referring to FIG. 3B, a process 232 for handling an event associated with accessing a raw storage device may include steps 321-328. At step 321, the tracking module 110 determines whether the event indicates that a computational entity (CE) is attempting to modify a master boot record (MBR) of a boot-type storage device.

At step 321, if the event indicates that the CE is attempting to modify the MBR, the process 232 proceeds from step 321 to step 322. At step 322, the tracking module 110 determines whether the CE is part of a larger computational entity, and if so, links the computational entities. At step 323, the categorization module 122 assigns potential ransomware Category B (or B1) to the CE. At step 324, the behavioral security engine 100 updates the CE's threat score. At step 325, the behavioral security engine 100 performs a process 250 for evaluating the CE's threat score and applying a corresponding policy, if any. Process 232 then ends.

At step 321, if the event does not indicate that the CE is attempting to modify the MBR, the process 232 proceeds from step 321 to step 326. At step 326, if the event indicates that the CE is attempting to modify the file system of a storage device, the process 232 proceeds to step 327. At step 327, the tracking module 110 determines whether the CE is part of a larger computational entity, and if so, links the computational entities. At step 328, the categorization module 122 assigns potential ransomware Category B (or B2) to the CE. At step 324, the behavioral security engine 100 updates the CE's threat score. At step 325, the behavioral security engine 100 performs a process 250 for evaluating the CE's threat score and applying a corresponding policy, if any. Process 232 then ends.

At step 326, if the event does not indicate that the CE is attempting to modify the file system of a storage device, the process 232 ends.

Referring to FIG. 3C, a process 233 for handling an event associated with accessing a backup file may include steps 331-335. At step 331, if the event indicates that the CE is attempting to modify the backup capabilities of the computer system (e.g., by disabling the backup functionality, deleting a backup file, modifying a backup file, etc.), the process 233 proceeds from step 331 to step 332. At step 332, the tracking module 110 determines whether the CE is part of a larger computational entity, and if so, links the computational entities. At step 333, the categorization module 122 assigns potential ransomware Category C to the CE. At step 334, the behavioral security engine 100 updates the CE's threat score. At step 335, the behavioral security engine 100 performs a process 250 for evaluating the CE's threat score and applying a corresponding policy, if any. Process 233 then ends.

At step 331, if the event does not indicate that the CE is attempting to modify the backup capabilities of the computer, the process 233 ends.

Referring to FIG. 3D, a process 234 for handling an event associated with enumerating a directory may include steps 341-347. At step 341, if the directory being enumerated by the CE is one of a specified set of suspicious enumeration starting points, the process 234 proceeds from step 341 to step 342. At step 342, the categorization module 122 assigns potential ransomware Category A to the CE. At step 343, the behavioral security engine 100 adds the directory to a set of enumerated directories in the CE's profile state. At step 344, the behavioral security engine 100 activates the scoring and subcategorization heuristics associated with Category A ransomware for the CE. Process 234 then ends.

At step 341, if the directory being enumerated by the CE is not one of the specified set of suspicious enumeration starting points, the process 234 proceeds from step 341 to step 345. At step 345, if the directory being enumerated by the CE is a child (e.g., a direct child) of a directory that is already in the set of enumerated directories in the CE's profile state, the process 234 proceeds from step 345 to step 346. At step 346, the behavioral security engine 100 adds the directory to the set of enumerated directories in the CE's profile state. At step 347, the tracking module 110 determines whether the CE is part of a larger computational entity, and if so, links the computational entities. Process 234 then ends.

At step 345, if the directory being enumerated by the CE is not a child (e.g., a direct child) of a directory that is already in the set of enumerated directories in the CE's profile state, the process 234 ends.

Referring to FIG. 3E, a process 235 for handling an event associated with accessing a file may include steps 351-363. At step 351, if the event indicates that the CE is attempting to

modify a file that is private to a different computational entity, the process 235 proceeds from step 351 to step 352. At step 352, the tracking module 110 determines whether the CE is part of a larger computational entity, and if so, links the computational entities. At step 353, the categorization module 122 assigns potential ransomware Category A to the CE. At step 354, the behavioral security engine 100 updates the CE's threat score. At step 355, the behavioral security engine 100 performs a process 250 for evaluating the CE's threat score and applying a corresponding policy, if any. Process 235 then proceeds to step 356. Likewise, at step 351, if the event indicates that the CE is attempting to modify a file that is not private to a different computational entity, the process 235 proceeds from step 351 to step 356.

At step 356, if the CE has already been assigned a potential malware category, the process 235 proceeds from step 356 to step 357. At step 357, if the file is in a directory in the CE's set of enumerated directories (as indicated in the CE's profile state), the process 235 proceeds from step 357 to step 358. On the other hand, if the CE has not been categorized as potential ransomware category or the file is not in one of the CE's enumerated directories, process 235 ends.

At step 358, the tracking module 110 maps attributes of the file and the event (e.g., file type, etc.) to heuristic states. At step 359, the behavioral security engine 100 runs the subcategorization heuristics associated with the CE's ransomware category (e.g., Category A). At step 360, the behavioral security engine 100 runs the score adjustment heuristics. At step 361, the behavioral security engine 100 updates the CE's threat score. At step 362, the behavioral security engine 100 performs a process 250 for evaluating the CE's threat score and applying a corresponding policy, if any. At step 363, the protection and mitigation module 160 backs up the file. Process 235 then ends.

Referring to FIG. 3F, a process 237 for handling an event associated with accessing a network may include steps 371-374. At step 371, if the CE attempting to access the network has been categorized as potential Class A ransomware, the process 237 proceeds from step 371 to step 372. At step 372, the behavioral security engine 100 runs the score adjustment heuristics. At step 373, the behavioral security engine 100 updates the CE's threat score. At step 374, the behavioral security engine 100 performs a process 250 for evaluating the CE's threat score and applying a corresponding policy, if any. Process 237 then ends.

At step 371, if the CE attempting to access the network has not been categorized as potential Class A ransomware, the process 237 ends.

Referring to FIG. 3G, a process 238 for handling an event associated with performing an application programming interface (API) call (or system call) may include steps 381-385. At step 381, if the API call (or system call) is one of a specified set of API calls (or system calls)

associated with ransomware, the process 238 proceeds from step 381 to step 382. At step 382, if the CE attempting to access the network has been categorized as potential Class A ransomware, the process 238 proceeds from step 382 to step 383. On the other hand, if the API call (or system call) is not one of the specified set of calls or the CE has not been categorized as potential Class A ransomware, process 238 ends.

At step 383, the behavioral security engine 100 runs the score adjustment heuristics. At step 384, the behavioral security engine 100 updates the CE's threat score. At step 385, the behavioral security engine 100 performs a process 250 for evaluating the CE's threat score and applying a corresponding policy, if any. Process 238 then ends.

Referring to FIG. 3H, a process 250 for evaluating the threat score of a computational entity and applying a security policy to the entity may include steps 391-398. At step 391, the behavioral security engine 100 determines whether the CE's threat score exceeds a corresponding threshold. If not, the process 250 ends.

If the CE's threat score exceeds the threshold, the process 250 proceeds to step 392. At step 392, if a prevention policy is enabled, the behavioral security engine 100 terminates the CE (step 395) and restores the files modified by the CE (step 398). If the prevention policy is not enabled, the process 250 proceeds to step 393. At step 393, if an isolation policy is enabled, the behavioral security engine isolates the CE (step 396). At step 394, if a logging policy is enabled, the behavioral security engine initiates logging of the CE's behavior. At step 398, the behavioral security engine restores the files modified by the CE.

Referring to FIG. 4, a ransomware detection method 400 may include steps 410-470. The steps of the ransomware detection method 400 may be performed, for example, by the behavioral security engine 110, using the techniques described above.

At step 410, one or more instances of a computational entity (CE) engaging in one or more first behaviors associated with encrypting a file, encrypting a storage device, and/or disabling a backup of at least a portion of a computer system are detected (e.g., by the tracking module 110). The detected instances of the first behaviors may be detected based on first events initiated by the CE in the computer system. Such events may be detected, for example, by the event monitoring module 114.

At step 420, a potential ransomware category is assigned to the CE based, at least in part, on the detected instances of the first behaviors. The ransomware category may be assigned to the CE, for example, by the categorization module 122.

At step 430, after the potential ransomware category is assigned to the CE, a protection action may be initiated to protect resources of the computer system against ransomware attacks. The protection action may be initiated, for example, by the behavior scoring module 130, the

policy module 150, and/or the protection and mitigation module 160. The protection action may, for example, protect storage resources of the computer system against modification operations.

At step 440, after the potential ransomware category is assigned to the CE, one or more instances of the CE engaging in one or more second behaviors corresponding to the ransomware category may be detected. The instances of the second behaviors may be detected, for example, by the behavior tracking module 114. In some embodiments, the instances of the second behaviors are detected based on second events initiated by the CE in the computer system. Such events may be detected, for example, by the event monitoring module 112. In some cases, the tracking module 110 may be configured to ignore the CE initiating the second events and/or engaging in the instances of the second behaviors unless the ransomware category has been assigned to the CE.

At step 450, a potential ransomware subcategory is assigned to the CE based on the detected instances of the second behaviors. The ransomware subcategory may be assigned to the CE, for example, by the subcategorization module 124.

At step 460, a threat score may be assigned to the CE (e.g., the threat score may be initialized or adjusted) based on the potential ransomware category, the potential ransomware subcategory, and/or the detected instances of the second behaviors. The threat score may be determined, for example, by the behavior scoring module 130 based on data provided by the heuristics module 120.

At step 470, a determination is made as to whether the threat score exceeds a threshold score associated with the ransomware category or subcategory assigned to the CE. This determination may be made, for example, by the behavior scoring module 130. If the threat score exceeds the corresponding threshold score, a policy action (e.g., a mitigation action or a protection action) associated with the ransomware category or subcategory assigned to the CE may be initiated. The policy action may be initiated, for example, by the behavior scoring module 130, the policy module 150, and/or the protection and mitigation module 160.

One of ordinary skill in the art will appreciate that the ransomware detection and prevention methods described herein use a variety of concepts to attain low a false negative rate, a low false positive rate, and a very low impact on computer system performance. The term "ransomware" generally refers to a large and diverse set of data-encrypting malware. The techniques used by ransomware may vary greatly: MBR and raw disk encrypting; various strains of file-encrypting; strains targeting specific file types, strains utilizing different file IO and encryption techniques, single CU or multi CU strains, strains utilizing built-in OS tools, strains using different file system discovery techniques, etc. Thus, ransomware detection and protection

techniques that are robust and do not significantly degrade system performance are very difficult to devise.

The techniques described herein address this tension by (1) targeting foundational behaviors that are highly likely to be used by any data-encrypting malware, (2) assigning potential ransomware categories and subcategories to computational entities (CEs) based on their behavior and customizing (or tailoring) the detection, protection, and mitigation policies applied to each CE based on those categories and subcategories, and (3) using a rich set of category-specific and subcategory-specific heuristics to increase and decrease the CE's threat score based on the CE's behavior. In this way, the techniques described herein can further increase overall detection confidence, yielding a balanced system suitable for effective ransomware prevention.

Some embodiments of the behavioral security engine 100 may exhibit the following characteristics:

(1) A multi-tier architecture may be used, with a rich set of heuristic methods present in each tier.

(2) An initial categorization tier may be used to analyze a limited, well-defined set of behavioral events for every CE and to categorize potential ransomware into specific high-level categories. This tier may act as pre-filter that filters out obvious non-ransomware while continuing to track any potential ransomware. The pre-filtering may have a variety of benefits. For example, by monitoring high-level types of behavior any file-encrypting malware is highly likely to exhibit, the initial categorization tier ensures that a very high percentage of potential ransomware (e.g., all ransomware) (including zero-day ransomware) passes through the filter, yielding a low false negative rate. Filtering out the non-ransomware also lowers the false positive rate and increases the effectiveness of the more specialized heuristics in the other tiers by filtering out irrelevant events that could otherwise confuse the heuristics or thwart their analysis. As another example, this approach conserves system resources (and reduces the impact on system performance) by reducing the number and types of events that are collected before the first trigger point and limiting the scope of computational analysis done for every CE.

(3) Once the initial high-level category is determined and if further behavioral analysis is warranted to gain confidence, the secondary categorization tier opens up the event stream relevant to that high-level category and adapts the behavioral analysis by selecting algorithms specific to that category to establish a high-confidence sub-category. For example, the actual encryption technique may be identified in this step. The benefit is an adaptive choice of specific lower-level encryption-detection algorithms that are well-suited for the encryption method, which can be an important factor in achieving high accuracy and both low false negative and false positives. This tier can also detect a starting point in a potential encryption of the first data

unit, which can trigger the engine to start backing up files and prevent data loss until a prevention action is applied.

(4) Once the specific encryption techniques are identified, another tier of specialized heuristics (for the specific encryption technique) may be used to increase and decrease the CE's threat score. A rich set of heuristics may be used to monitor specific behaviors, patterns and anti-patterns that have a net effect of balancing each other. The use of such methods may increase the confidence in the engine's analysis, further decreasing false positives and false negatives. Also, the system architecture may facilitate easy extensibility and addition of novel methods in this category.

Further Description of Some Embodiments

In some examples, some or all of the processing described above can be carried out on an endpoint device (e.g., a personal computing device, laptop device, tablet device, smartphone device, etc.), on one or more centralized computing devices, or via cloud-based processing by one or more servers. In some examples, some types of processing occur on one device and other types of processing occur on another device. In some examples, some or all of the data described above can be stored on a personal computing device, in data storage hosted on one or more centralized computing devices, or via cloud-based storage. In some examples, some data are stored in one location and other data are stored in another location. In some examples, quantum computing can be used. In some examples, functional programming languages can be used. In some examples, electrical memory, such as flash-based memory, can be used.

FIG. 5 is a block diagram of an example computer system 500 that may be used in implementing the technology described in this document. General-purpose computers, network appliances, mobile devices, or other electronic systems may also include at least portions of the system 500. The system 500 includes a processor 510, a memory 520, a storage device 530, and an input/output device 540. Each of the components 510, 520, 530, and 540 may be interconnected, for example, using a system bus 550. The processor 510 is capable of processing instructions for execution within the system 500. In some implementations, the processor 510 is a single-threaded processor. In some implementations, the processor 510 is a multi-threaded processor. The processor 510 is capable of processing instructions stored in the memory 520 or on the storage device 530.

The memory 520 stores information within the system 500. In some implementations, the memory 520 is a non-transitory computer-readable medium. In some implementations, the memory 520 is a volatile memory unit. In some implementations, the memory 520 is a non-volatile memory unit.

The storage device 530 is capable of providing mass storage for the system 500. In some implementations, the storage device 530 is a non-transitory computer-readable medium. In various different implementations, the storage device 530 may include, for example, a hard disk device, an optical disk device, a solid-date drive, a flash drive, or some other large capacity storage device. For example, the storage device may store long-term data (e.g., database data, file system data, etc.). The input/output device 540 provides input/output operations for the system 500. In some implementations, the input/output device 540 may include one or more of a network interface devices, e.g., an Ethernet card, a serial communication device, e.g., an RS-232 port, and/or a wireless interface device, e.g., an 802.11 card, a 3G wireless modem, or a 4G wireless modem. In some implementations, the input/output device may include driver devices configured to receive input data and send output data to other input/output devices, e.g., keyboard, printer and display devices 260. In some examples, mobile computing devices, mobile communication devices, and other devices may be used.

In some implementations, at least a portion of the approaches described above may be realized by instructions that upon execution cause one or more processing devices to carry out the processes and functions described above. Such instructions may include, for example, interpreted instructions such as script instructions, or executable code, or other instructions stored in a non-transitory computer readable medium. The storage device 530 may be implemented in a distributed way over a network, for example as a server farm or a set of widely distributed servers, or may be implemented in a single computing device.

Although an example processing system has been described in FIG. 5, embodiments of the subject matter, functional operations and processes described in this specification can be implemented in other types of digital electronic circuitry, in tangibly-embodied computer software or firmware, in computer hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented as one or more computer programs, i.e., one or more modules of computer program instructions encoded on a tangible nonvolatile program carrier for execution by, or to control the operation of, data processing apparatus. Alternatively or in addition, the program instructions can be encoded on an artificially generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. The computer storage medium can be a machine-readable storage device, a machine-readable storage substrate, a random or serial access memory device, or a combination of one or more of them.

The term "system" may encompass all kinds of apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. A processing system may include special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit). A processing system may include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

A computer program (which may also be referred to or described as a program, software, a software application, a module, a software module, a script, or code) can be written in any form of programming language, including compiled or interpreted languages, or declarative or procedural languages, and it can be deployed in any form, including as a standalone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program may, but need not, correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

The processes and logic flows described in this specification can be performed by one or more programmable computers executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

Computers suitable for the execution of a computer program can include, by way of example, general or special purpose microprocessors or both, or any other kind of central processing unit. Generally, a central processing unit will receive instructions and data from a read-only memory or a random access memory or both. A computer generally includes a central processing unit for performing or executing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Moreover, a computer can be embedded in another device, e.g., a mobile telephone, a personal digital assistant (PDA), a mobile audio or video player, a game console, a

Global Positioning System (GPS) receiver, or a portable storage device (e.g., a universal serial bus (USB) flash drive), to name just a few.

Computer readable media suitable for storing computer program instructions and data include all forms of nonvolatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

To provide for interaction with a user, embodiments of the subject matter described in this specification can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube) or LCD (liquid crystal display) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's user device in response to requests received from the web browser.

Embodiments of the subject matter described in this specification can be implemented in a computing system that includes a back end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

While this specification contains many specific implementation details, these should not be construed as limitations on the scope of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments. Certain features that are described in

this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various system components in the embodiments described above should not be understood as requiring such separation in all embodiments, and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products.

Particular embodiments of the subject matter have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous. Other steps or stages may be provided, or steps or stages may be eliminated, from the described processes. Accordingly, other implementations are within the scope of the following claims.

Terminology

The phraseology and terminology used herein is for the purpose of description and should not be regarded as limiting.

The term “approximately”, the phrase “approximately equal to”, and other similar phrases, as used in the specification and the claims (e.g., “X has a value of approximately Y” or “X is approximately equal to Y”), should be understood to mean that one value (X) is within a predetermined range of another value (Y). The predetermined range may be plus or minus 20%, 10%, 5%, 3%, 1%, 0.1%, or less than 0.1%, unless otherwise indicated.

The indefinite articles “a” and “an,” as used in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean “at least one.” The phrase “and/or,” as used in the specification and in the claims, should be understood to mean “either or

both" of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with "and/or" should be construed in the same fashion, i.e., "one or more" of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the "and/or" clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to "A and/or B", when used in conjunction with open-ended language such as "comprising" can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment, to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

As used in the specification and in the claims, "or" should be understood to have the same meaning as "and/or" as defined above. For example, when separating items in a list, "or" or "and/or" shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as "only one of or "exactly one of," or, when used in the claims, "consisting of," will refer to the inclusion of exactly one element of a number or list of elements. In general, the term "or" as used shall only be interpreted as indicating exclusive alternatives (i.e. "one or the other but not both") when preceded by terms of exclusivity, such as "either," "one of," "only one of," or "exactly one of." "Consisting essentially of," when used in the claims, shall have its ordinary meaning as used in the field of patent law.

As used in the specification and in the claims, the phrase "at least one," in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase "at least one" refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, "at least one of A and B" (or, equivalently, "at least one of A or B," or, equivalently "at least one of A and/or B") can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally including more than one, B, with no A present (and optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

The use of "including," "comprising," "having," "containing," "involving," and variations thereof, is meant to encompass the items listed thereafter and additional items.

Use of ordinal terms such as "first," "second," "third," etc., in the claims to modify a claim element does not by itself connote any priority, precedence, or order of one claim element over another or the temporal order in which acts of a method are performed. Ordinal terms are used merely as labels to distinguish one claim element having a certain name from another element having a same name (but for use of the ordinal term), to distinguish the claim elements.

Equivalents

Having thus described several aspects of at least one embodiment of this invention, it is to be appreciated that various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications, and improvements are intended to be part of this disclosure, and are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description and drawings are by way of example only.

What is claimed is:

CLAIMS

1. A computer-implemented method for ransomware detection, comprising:

based on first events initiated by a computational entity in a computer system, detecting one or more instances of the computational entity engaging in one or more first behaviors associated with encrypting a file, encrypting a storage device, and/or disabling a backup of at least a portion of the computer system;

assigning at least one potential ransomware category to the computational entity based on the detected instances of the first behaviors; and

after assigning the potential ransomware category to the computational entity,

initiating one or more protection actions to protect one or more storage resources against one or more modification operations, and

based on second events initiated by the computational entity, detecting one or more instances of the computational entity engaging in one or more second behaviors corresponding to the potential ransomware category.

2. The method of claim 1, wherein:

the first behaviors associated with encrypting a file include enumerating directories of a file system and/or modifying a private file,

the first behaviors associated with encrypting a storage device include modifying the storage device, and

the first behaviors associated with disabling a backup of at least a portion of the computer system include disabling and/or deleting a backup file, wherein the backup file stores backup data for a plurality of files and/or settings of the computer system.

3. The method of claim 2, wherein detecting the instance of the computational entity enumerating directories of a storage device comprises:

detecting an enumeration operation initiated by the computational entity on a specified first directory, wherein the specified first directory is (1) a top-level directory of the storage device, (2) a user directory in a file system of the storage device, or (3) an immediate subdirectory of a user directory in the file system of the storage device.

4. The method of claim 3, wherein a second directory is nested within the first directory, and wherein detecting the instance of the computational entity enumerating directories of the storage device further comprises: