

UNITED STATES PROVISIONAL PATENT APPLICATION

for

CLOUD CYBERSECURITY

by

Avi Shua

TECHNICAL FIELD

[001] The cloud estate of an organization may include its complete inventory of cloud assets including running and stopped workloads of all types including VMs, containers, storage objects, load balancers, IAM configurations, and more. Organizations are searching for effective ways to scan their cloud estate to look for risks stemming from vulnerabilities, misconfigurations, malware, lateral movement risk, weak and leaked passwords, and improperly secured PII.

[002] Disclosed embodiment introduces a new approach that secures the entire cloud estate and helps meet compliance mandates, but without disrupting business operations in live environments.

BACKGROUND

[003] Cloud workloads are vastly different than '90s-style physical servers running on bare metal. Unfortunately, many organizations ended up having the same agents and scanners from their on-prem days for their cloud environments. The tools were not reimagined for the cloud.

[004] Relying on agents for security visibility in the cloud is fundamentally flawed. Visibility is critically limited to only those assets that are already known and accessible. What's more, the assets must be capable of having an agent installed and maintained, and the assets must have ongoing network connectivity to the backend. Yet in the fast-paced world of DevOps, developers do not want to be bothered with deploying agents on VMs, in containers, and in serverless configurations—let alone dealing with their never-ending maintenance.

[005] 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.

[006] An unauthenticated scan can only examine publicly visible information and is not able to provide detailed information about assets. It is essentially acting as a friendly attacker. An unauthenticated scan can easily miss identifying some assets and vulnerabilities, making it much less effective. For example, say you have a website called `mydomain.com/email` campaign that is not linked from your main website. The site will not be scanned unless the scanner is manually configured, but no organization can ensure it's set up that way. This leaves many organizations exposed to vulnerabilities in areas where the scanner cannot reach.

[007] While unauthenticated scanners act like an attacker, they often get stuck where a real attacker would not. For example, a CAPTCHA can easily prevent any automatic mechanism (including scanners) from registering. However, these techniques won't have any effect on a human who tries to attack the same system. Orca found a critical vulnerability in a section of a customer's website that is only accessible to registered users. A network scanner would get blocked here, but a real attacker could register as a user and trigger a vulnerability leading to a breach.

[008] 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 will never detect 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.

[009] 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

[0010] This summary is provided for the convenience of the reader to provide a basic understanding of a few exemplary embodiments 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.

[0011] Before the cloud, we secured physical hosts. That meant spending time installing multiple security agents—one for each server. But at least we were living in a fairly static world. IP addresses were assigned to physical assets and they seldom changed. Even then, as every security veteran knows, agent integration was tedious and coverage rarely reached 100% of assets. Then the cloud started making virtual what

used to be physical. So we used what we had. We took security agents that ran on physical hosts and ran them on virtual machines. Cloud security deserves better.

[0012] In a cloud environment, you're scaling utilization up and down frequently—possibly thousands of times per hour across multiple clouds—and all within a CI/CD pipeline that builds your infrastructure. You have containers and VMs to deal with, and agents carry huge operational costs.

[0013] Orca Security takes a radical new approach. With no legacy on-prem environments to protect, Orca was free to create a cloud-native security platform without the constraints of agents and network scanners.

[0014] Orca delivers instant-on, work-load level visibility across 100% of AWS, Azure, and GCP assets without running a single opcode in the customer environment, 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: I/S, OS, applications, and data; (3) Discover and see previously missed assets.

DRAWINGS

[0015] 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. In the drawings.

DETAILED DESCRIPTION

[0016] 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, this detailed description provides examples of

Orca's implementations. 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.

[0017] 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.

[0018] 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.

[0019] 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.

[0020] 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.

[0021] The memory 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.

[0022] 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.

[0023] 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.

[0024] 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.

[0025] 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.

[0026] “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.

[0027] 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.

[0028] 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.

SideScanning Technology

[0029] Orca Security uses a novel technology called SideScanning.™ SideScanning is a radical approach because Orca does not necessarily go *inside* each workload to inspect data. Instead, it uses an out-of-band process to reach cloud workloads through the

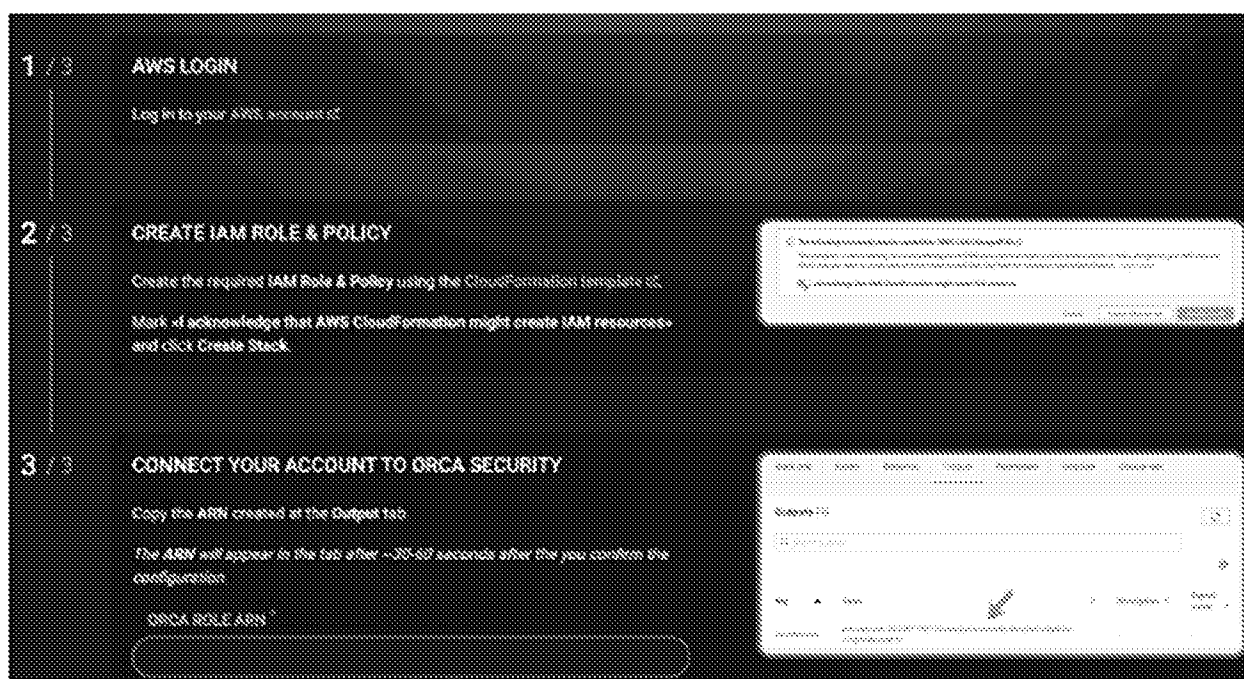
runtime storage layer, combining this with metadata gathered from cloud provider APIs. Orca may provide deep and contextualized visibility of cloud environments. It covers 100% of an organization's assets with necessarily needing an agent or network scanner.

[0030] Orca Security may involve a one-time, essentially instantaneous, impact-free integration into AWS, Azure, or GCP. Following its one-time integration, Orca scans the configuration, network layout, and security configuration. (This may be performed via a scanner account.) It does so while also reading into virtual machines, disks, databases, and datastores, as well as logs for cloud assets. It then analyzes the data and builds a full-stack inventory. Next it automatically assesses the security state of discovered asset throughout the technology stack, including all four cloud layers: I/S, OS, apps, and data. The system may monitor cloud change (e.g. delta) logs (e.g. 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) to act on any changes (e.g. perform supplementary scan) and contextualize associated risks discovered within a customer cloud account.

[0031] An apt analogy is to think of a medical MRI. Instead of probing inside the body with needles and scalpels, such imaging is an out-of-band method of obtaining a detailed picture of the organs and tissue within. The person is never physically touched. SideScanning is similar in that it is able to build a full model of the cloud environment without affecting it in any way—and all assets and their associated risks are clearly visible. Orca can probe the read-only view it has obtained in an entirely touchless manner.

[0032] Orca may not affect or run on any virtual cloud assets, where it might consume resources. This lets an organization fully deploy Orca across 100% of its cloud environment without worrying about potential side effects on performance. And Orca does this without the friction of working with disparate teams (e.g. DevOps) to assess that the timing for deployment is correct.

Overview – Orca’s onboarding process is simple and quick. A client provides Orca with a role and establish trust between the client account (e.g., source or target account) and Orca’s production account (e.g. scanner account). This is one example of trusted relationship. The role has a few permissions, the most important being read-only permissions and permissions to read the block storage layer (containing block storage volumes). The entire process is encapsulated within a cloud formation template, which means that an administrator may need to only click once to open the template, then again to apply it. A third mouse click copies and pastes the resulting ARN in the Orca user interface.



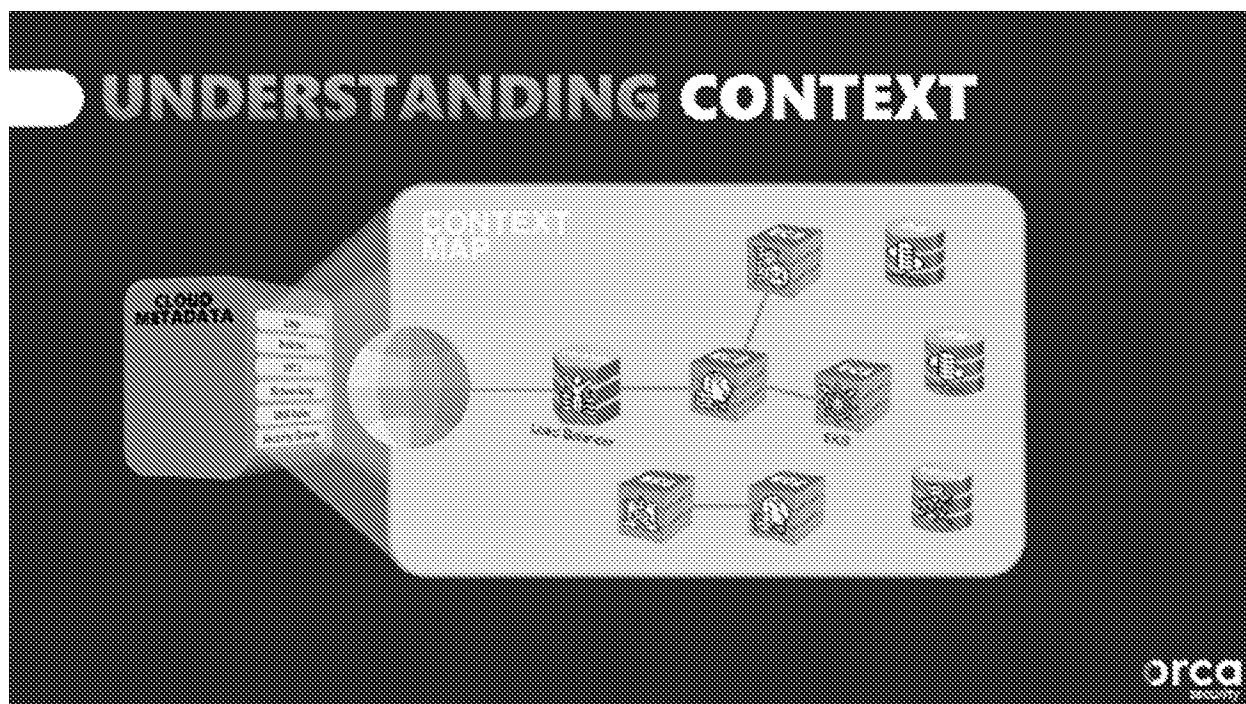
[0033]Permission Detail – Orca’s read-only permission enables it to visualize and build a map of a customer’s entire environment. The same permission also lets Orca create a temporary snapshot of the block storage for its subsequent analysis. Here it applies Orca-specific tags to the snapshot, then has permission to delete snapshots having the Orca-specific tags. (Orca cannot delete other snapshots in your customer account.)

[0034]Orca requests permission to read encrypted key management service (KMS) volumes to open snapshots in that account. Orca does not need to copy the customer key, but rather uses it to re-encrypt the snapshots using its own key to continue the examination. It is a lightweight operation that records blocks that are part of the snapshot and the copy itself. The solution just records a reference count to those blocks and copies them like a copy and write operation. For example, the system may passively perform cryptographic analysis on the plurality of identified keys to identify non-functional first fingerprints that uniquely identify each of the plurality of keys and compare them to match keys with compute resources without using the keys to access the compute resources.

[0035]After a customer adds an account to Orca, the SideScanning™ process starts with building a map of the customer’s organization’s entire estate. A map may be in a form of graph, a plurality of interconnected vectors connecting the plurality of entities based on the networking configuration. The map/graph may be traversed to identify vectors originating in the internet and reaching the workload. The map/graph may be displayed on graphical user interface. Additional databases or other data structures may be used to enhance the map/graph by identifying services publicly accessible via the

internet and capable of serving as an internet proxy. Every asset in the account may be enumerated in all regions (e.g. geographic locations), including: API gateway resources, API gateway REST APIs, Autoscaling groups, CloudTrail logs, CloudFront services, Databases—such as ElasticCache, ElasticSearch, DocumentDb, DynamoDB, Neptune, RDS and Aurora, Redshift and Kafka clusters, EC2 instances, 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, and users, KMS keys, Lambda functions.

[0036] After the cloud assets are enumerated, the SideScanning™ process may split into two paths. The first is the control plane path, where Orca builds an infrastructure map that acts as a guide for its assessment and analysis processes. Because putting risks in context is a valuable feature in some embodiments, the map also enables Orca to contextualize assets and their associated risks discovered within the customer cloud account.



[0037] During this phase, Orca may provide definitive alerts (e.g. risk notifications) regarding: S3 buckets exposed to the world, snapshots that have been published to the entire world, other misconfigurations at the cloud control plane level. A categorical risk level assessment may be performed. By way of a non-limiting example, the following risk levels may be utilized: (1) compromise - attacker already gained access; (2) imminent compromise - attack vector exists and can be used; (3) hazardous - serious security implication, but no full vector exists; (4) informational - no action required, but need to know e.g. low space alert. Risk levels may be displayed via alerts or other risk notifications on a graphical user interface, or by any other means for alerting end user.

[0038] Orca's control plane path may yield a complete overview of the cloud estate. Looking at all assets, Orca can see which are connected to understand the relationships among them. It can detect risks in this phase without having to drill down.

[0039] For each region (e.g. geographic location) in a customer's cloud account, Orca may enumerate all the possible compute assets, taking snapshots of their data volumes to share with the Orca production account. Identification of a geographic location may include a region identifier, site identifier, datacenter identifier, physical address, network address, workload name, or any other identifier which may be required via cloud provider API. A collector—an ephemeral EC2 instance—is created within the Orca Security Cloud account (usually in the form of a spot instance). The collector successively mounts each snapshot, then may immediately delete it so customer billing isn't affected. Next, the collector starts reorganizing the volumes. It mounts them in the same way as the scanned OS would have done (this is one example of reconstructing a state of the workload from a block storage volume) and runs several data collection steps (e.g. assessing the reconstructed state): Vulnerability Scanning, Configuration Scanning, Malware Scanning, Lateral Movement Risk, Exploitable Keys and Weak Password Detection, Sensitive Information Scanning, Container Scanning.

[0040] Orca does not need to run a clone of a customer's workload (e.g. within target or source account). Rather, the collector may rebuild the correct configuration and mounts the volumes in their native file systems in Orca's collection machine, which may be performed by determining the file system of an associated workload. This is done to ensure that Orca—and not any other entity—controls the scanning process. Also, this assists it in being more deliberate about the information it gathered from the machine.

[0041] A snapshot of the disk state may be created through AWS or other cloud providers while the machine is running. The process can snapshot multiple volumes at the same point in time, resulting in deep and consistent visibility. Additionally, the

system may provide a timeline visualization interface enabling users to interact with a timeline and quickly compare overall risk changes over time by allowing seamless navigation to a specific point in time. A timeline visualization interface is consistent with other aspects of the current disclosure and may serve to represent overall cybersecurity risk snapshots at different points in time by collecting and aggregating scan-derived data to, for example, permit management to assess the effectiveness of measures taken.

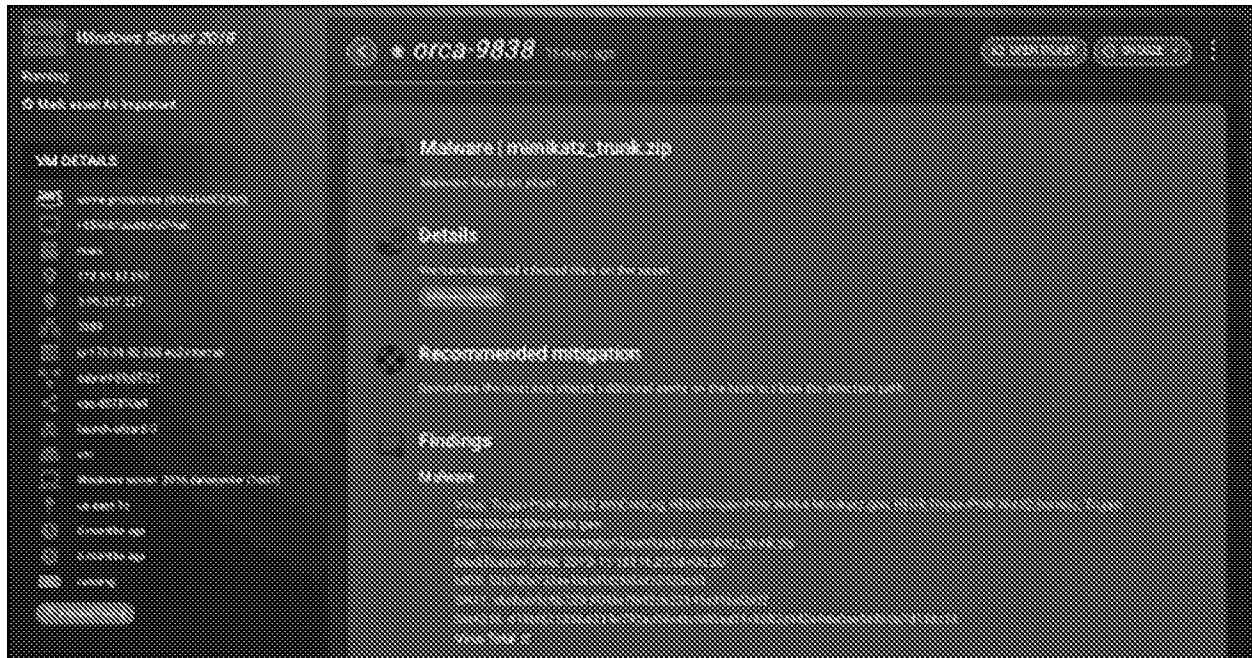
[0042] In performing vulnerability assessment, Orca may extract the OS packages, installed software applications, libraries, and program language libraries such as Java archives, Python packages, Go modules, Ruby gems, PHP packages, and Node.js modules, or any other software applications. It gathers library versions, software versions and other identifying characteristics, and in a later phase tries matching them to known vulnerabilities in its vulnerability database (e.g. data structure of known vulnerabilities). This may be performed by activating a scanner at the geographic location associated with the workload region. Among others, this database (or any other data structure) includes aggregated 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. or any other vulnerability data.

[0043] Because some packages might only be vulnerable in specific configurations, the vulnerability scanning results are augmented with configuration-specific details in the backend. Thus, Orca may gather configuration information—such as which user is on a machine, its services, and password hashes—in addition to application-specific configurations for Apache, Nginx, SSH, and other services. Orca may perform the first-level analysis on all that is collected to remove sensitive information, only sending high-level configuration information to its backend. At this point, Orca may run the CIS Benchmarks on the workloads to check for misconfigurations. E.g. a lookup of the identified installed software version may be performed and cloud provider API may be required to determine network accessibility of the associated workload.

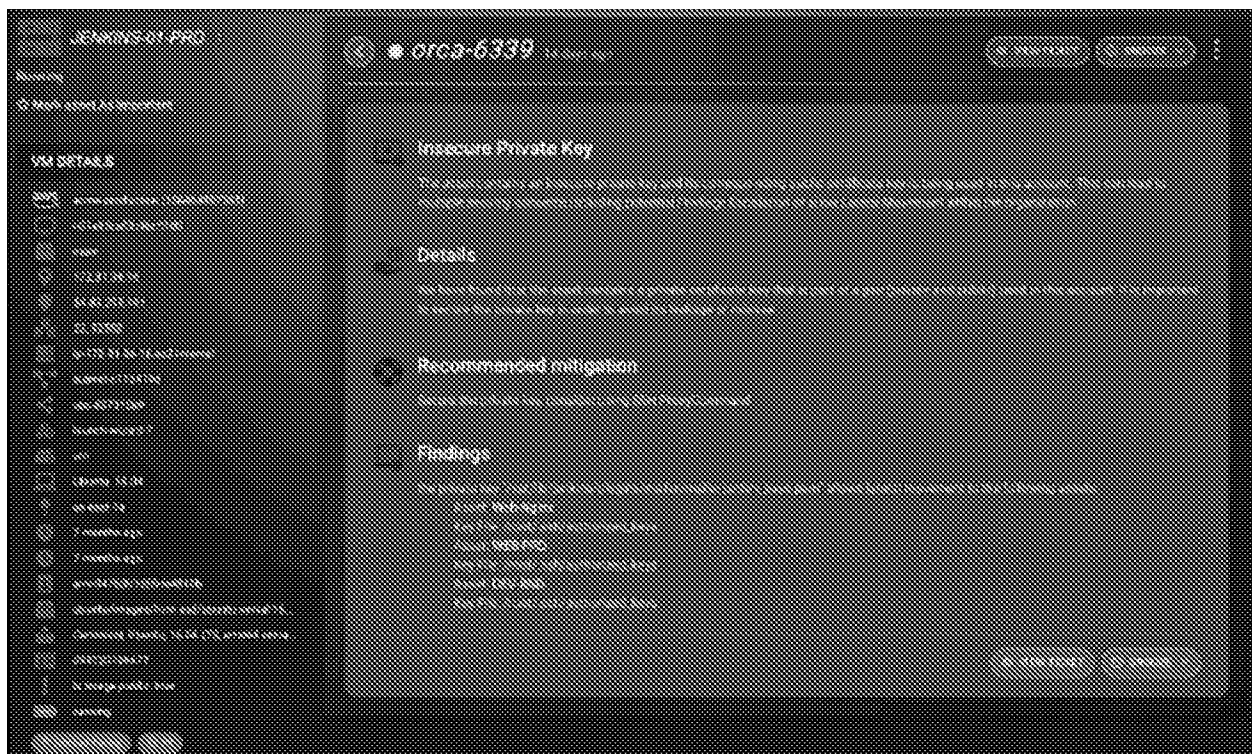
[0044] The Malware Scanning collector performs deep malware scanning across the entire file system; it uses a smart, third-party heuristic engine. For example, another security solution that only compares hashes may not detect polymorphic malware, but Orca does may do so with its deep scanning capabilities.

[0045] Malware scanning is performed on the Orca side (e.g. secondary system), to avoid affecting production workloads (e.g. primary system). Orca is free to devote significant compute resources to scanning because—unlike an agent—it is not limited by the customer machine's CPU and available memory.

[0046] Another benefit of this approach is that the scanner cannot be tampered by malware (e.g. malicious code) running on a machine, as the scanner runs on the Orca host (e.g. secondary system). This enables Orca to detect advanced threats, such as rootkits that can easily circumvent other forms of detection.



[0047] 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.



[0048] For each scanned machine, Orca gathers the remote access keys installed there, as well as any keys that could provide access to other network resources. Orca looks for passwords and IT scripts containing passwords that could be used by attackers against the environment—as well as AWS keys, SSH keys, or other key types that provide unchecked access to important resources. In essence, Orca acts like a whitelisted attacker. Once it reaches a machine, it looks for everything an attacker searches for and enumerates that in detailed reports.

[0049] Suppose there is a weak, unprotected password stored in one of the environments. For example, if someone's personal email has been compromised at any point, Orca looks for similar names and—either using known dictionaries or the account owner's previously leaked passwords—attempts a brute force login to the machine being tested. Orca makes note of the stored password along with a corresponding pointer to it, the password is not shared outside of its ephemeral collector.

[0050] This is also how Orca handles keys and other sensitive information. For example, for SSH keys Orca only extracts the key digest—the key hash. For AWS keys Orca extracts only the access key ID (which is not confidential) and the permissions the key is able to access. The key digest enables Orca to compare private and public keys and to show where lateral movement is not only possible but quite trivial to accomplish.

[0051] It's possible that keys having no meaning—such as test keys—are discovered. Rather than report a false positive, Orca tests the validity of the keys, extracts their permissions, and reports that. For other keys having multi-machine interactions, such as SSH keys, Orca uses the information to verify which other workloads, if any, can be opened by them. Here it reports, "This machine had a private key stored in an

unprotected place; it can provide root access to these other machines where the matching public key is installed."

[0052] In addition, Orca highlights bugs or other configuration risks that might only be exploitable from internal machines yet still facilitate an attacker's lateral movement. A dashboard may be provided that permits a user to select an asset, and to view on a common dashboard all the cybersecurity risks associated with that asset may be generated. Dashboard may be displayed on a graphical user interface and retrieve workload component cybersecurity risk information consistent with the current disclosure.

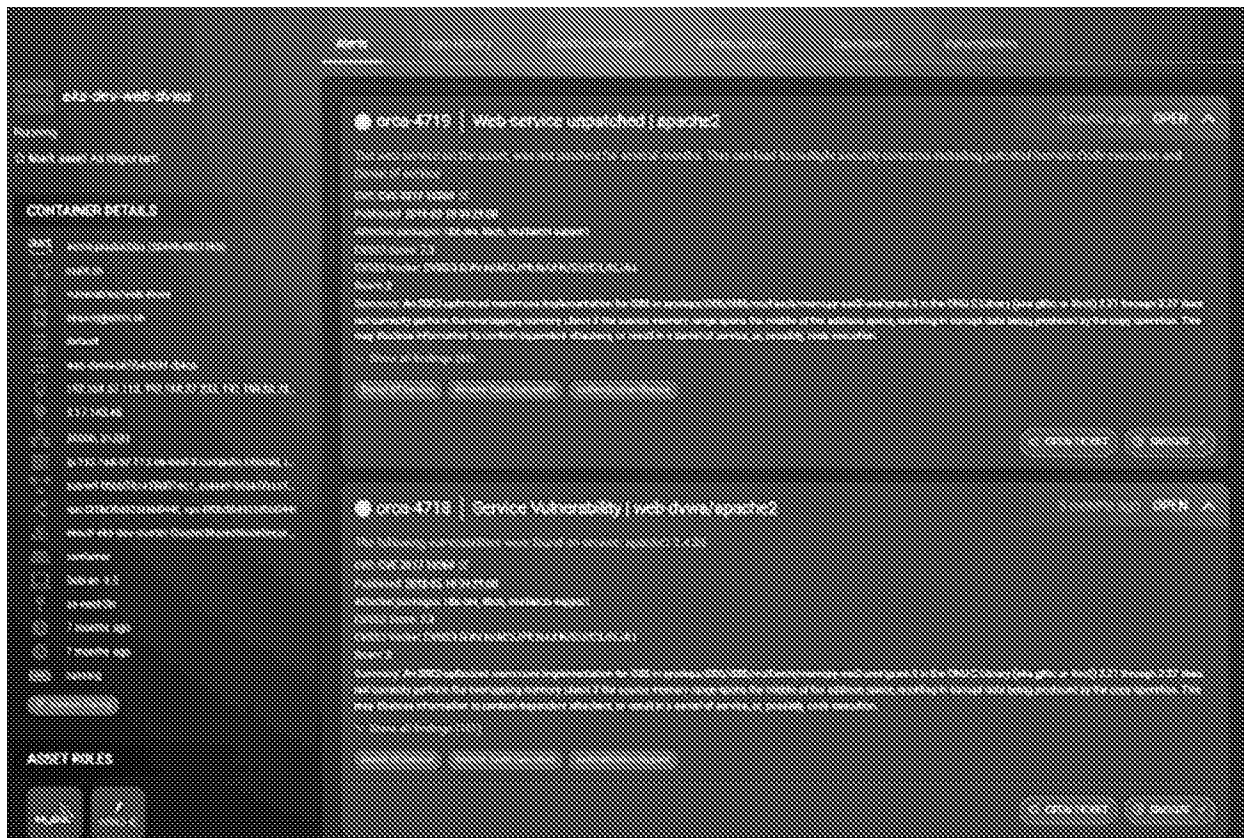
[0053] Another data collector searches the entire file system for sensitive information, such as PII, Social Security numbers, healthcare information, credit card numbers, and the like. It also searches data repository history. This is because it is not uncommon for an entire production environment repository to be cloned, with no one remembering the copy contains sensitive information. Orca tags such areas as risky, noting their location in its vulnerability report.

[0054] To be certain that such alerts do not constitute false positives, Orca may perform statistical scans on the workload level. It is very likely 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 contain one valid SSN by pure chance.

[0055] Orca utilizes the low-level access it has via SideScanning to scan containerized environments regardless of the orchestration mechanism. When reaching a workload that includes containers, the SideScanning container scanning component reconstructs the container runtime layered file system (LFS), and recursively runs the scanning

processes mentioned above — vulnerabilities, configuration, malware, lateral movement risk, sensitive data, etc.

[0056] Furthermore, Orca reads the container's network configuration to extend the contextual map built during the SideScanning control plane assessment phase (described later).



[0057] There are several benefits to the approach of scanning containers via SideScanning vs competing approaches: Unlike CI/CD or registry scanning solutions, Orca may scan the runtime view of the container, as opposed to the initial one. This means that it can detect deviations from baseline, including compromises of the environment. Orca may be agnostic to the orchestration layer and may cover all of the containers regardless of any integration to the orchestration layer (Kubernetes or other) that can be skipped.

[0058] Native PaaS environments, such as AWS Fargate are handled similarly — the major difference being the integration is done directly to Fargate and not via the underlying VM, which may not be accessible in these types of deployments.

[0059] Orca tears down its collectors and securely sends the gathered information to its backend to begin its data analysis and reporting phase.

[0060] All the gathered information is analyzed to produce actionable, context-based alerts, and reports. This thoughtful approach shows what each vulnerability is, where it's located, and its priority. In this way, security engineers and DevOps teams can easily assess how to best allocate their time and attention.

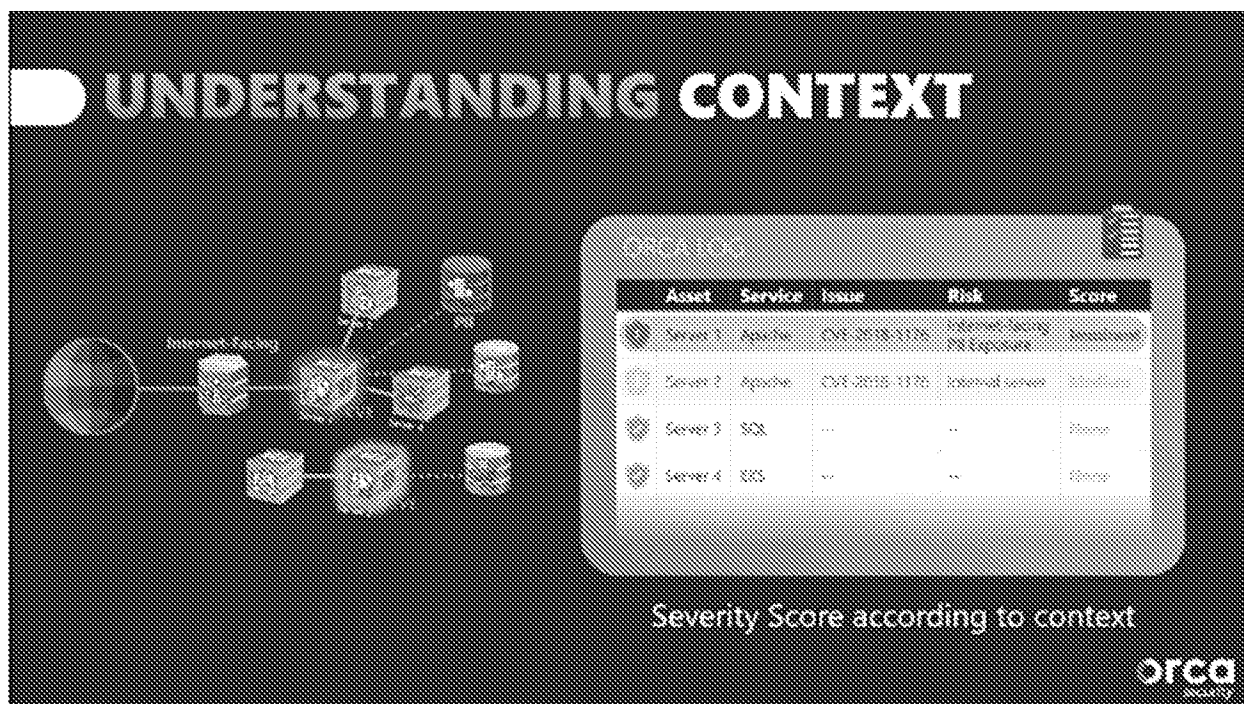
[0061] Orca combines conclusions from different environmental perspectives (e.g., metadata) into a single model. It takes the asset map from the control plane—where the service meshes and containers talk to each other—and enriches this information with the risk data gathered from the workload deep scanning. For example, Orca takes a customer's instance, maps the running services on it, then takes into consideration the vulnerability data collected in the data path. Orca contextualizes the information to establish whether an asset is internet-facing and easily accessible to attackers, or if it's private and hence a less important vulnerability. (A resource that isn't accessible by any other machine in the account represents less risk than one that's internet-facing.) For example, based on the contextual map, the system 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 additional, the system 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. Such analysis may be presented to a user via graphical user interface.

[0062] Consider a vulnerability in a web service. Orca scores it as: High risk if it's connected to the internet, either directly or indirectly via a load balancer or reverse proxy; Medium severity if it's only accessible internally; Low severity if it's blocked by a security group configuration of the cloud provider.

[0063] Another example would be if a machine is stopped. The machine could have an important 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.

[0064] Orca also evaluates network misconfigurations and their implications. Entities capable of routing traffic (e.g. routers, switches, firewalls, load balancers, proxy, etc) may be queried using a cloud provider API to determine network configurations. A common problem we have witnessed is when organizations use an external CI/CD service (such as Bitbucket) and whitelists their IP ranges—in effect whitelisting all of these services' customers while exposing internal services to the internet.



[0065] Understanding risk context is critical; it's the difference between effective security and dreaded analyst alert fatigue. Orca assumes responsibility for the heavy lifting associated with this additional context and assesses the real and effective risk. Orca's mission is to provide the best-contextualized security intelligence possible.

[0066] This is in contrast to point security tools that perform independent vulnerability detection. Getting the set of tools to talk to one another and provide a clear context about each finding is nearly impossible. The onus is put on customers to first establish context before being able to understand and subsequently prioritize risk; only then can they ultimately address the incomplete set of reported vulnerabilities. Traditional SIEM tools that ingest dissimilar data often suffer a similar fate, as they don't intimately understand the meaning of each alert created by the different tools and their collective meaning.

[0067] Orca’s context-based approach applies to containers just as it does to discrete workloads. In order for the context graph to reach the container level, Orca reads the network information during the container scanning phase and incorporates it into the overall context map. This means Orca understands which services within which containers are exposed externally and within which ports. E.g. a port on which vulnerable application is accessible may be identified based on known software vulnerabilities for a versions of software application associated with a workload. Network accessibility information may be quired via a cloud provider API and used to identify specific vulnerabilities susceptible to attack from outside the workload. All such context is merged into the “master map” discussed earlier, producing a map that includes the relationships between all of the workloads, discrete and containerized alike.

[0068] A common finding is when a machine hasn’t been patched for an extended period and has a large number—hundreds or even thousands—of vulnerabilities. While other security tools might send out one alert for each vulnerability, Orca aggregates the information to combine them with the appropriate context to show they’re related. For example, if a machine goes unpatched for a long period of time because it’s not internet-facing or connected to other machines, its risk score will be low—even though it has hundreds of vulnerabilities. This helps analysts in prioritizing which security fixes to address first.

[0069] Orca Security is revolutionary—both in its SideScanning approach to gathering cloud estate information and the way it presents risks and vulnerabilities in context. No other cloud security tool can deliver 100% deep, workload-level visibility across multiple cloud accounts and cloud platforms—with no impact whatsoever on the running cloud

environment. Orca doesn't require any agents or network scanners to deeply and thoroughly scan the full-stack cloud inventory.

[0070] Orca detects risks such as vulnerabilities, malware, misconfiguration, and lateral movement risk in the cloud. It sees what other security tools miss. Following data gathering and analysis, Orca reports all issues in full context, including where each risk resides and its true level of risk. This lets its customers prioritize their mitigation actions while mitigating cloud risk.

[0071] Orca Security is the cloud security innovation leader, providing instant-on, workload-level security and visibility into AWS, Azure, and GCP – without the gaps in coverage and operational costs of agents.

[0072] Delivered as SaaS, Orca Security's SideScanning™ technology reads your cloud configuration and workloads' runtime block storage out-of-band, detecting vulnerabilities, malware, misconfigurations, lateral movement risk, weak and leaked passwords, and unsecured PII.

[0073] Orca Security deploys in minutes – not months – because no opcode runs within your cloud environment. With Orca, there are no overlooked assets, no DevOps headaches, and no performance hits on live environments. As cybersecurity risks are resolved, cybersecurity risk, although addressed, may not be fully mitigated, leaving open vulnerabilities thought to be resolved. Orca solves this problem by integrating with third party systems (e.g., a ticketing system) and initiating a scan on assets whose tickets are marked as being resolved. If the scan determines incomplete resolution, the ticket may be returned for additional mitigation. The system may exchange scan-derived metadata with a third party system to confirm resolution and initiate a post-resolution

scan of the asset to assess mitigation of the at least one cybersecurity risk. If the post-resolution scan confirms mitigation, the system may update the ticketing tracker to indicate a successful resolution of the at least one cybersecurity risk. Alternatively, if the in an event post-resolution scan fails to confirm mitigation, system may update the ticketing tracker to reflect failed resolution.

[0074] And unlike legacy tools that operate in silos, Orca treats your cloud as an interconnected web of assets, prioritizing risk based on environmental context. This does away with thousands of meaningless security alerts to provide just the critical few that matter, along with their precise path to remediation.

Identify Lateral Movement Risks in Your Cloud (Pre-Breach)

[0075] Cyberattackers progressively move through a network as they search for key data and assets that ultimately fulfill their mission. This is referred to as lateral movement.

[0076] When an attacker compromises a cloud asset, usually that machine is not their target, but rather the path to it. After successful infiltration, they perform reconnaissance—exploring not just that machine, but also its network for information (risks) that might provide them with escalated privileges and access to other assets. So armed, they then use their gathered information to move from the initially compromised machine across the cloud estate until their mission is fulfilled.

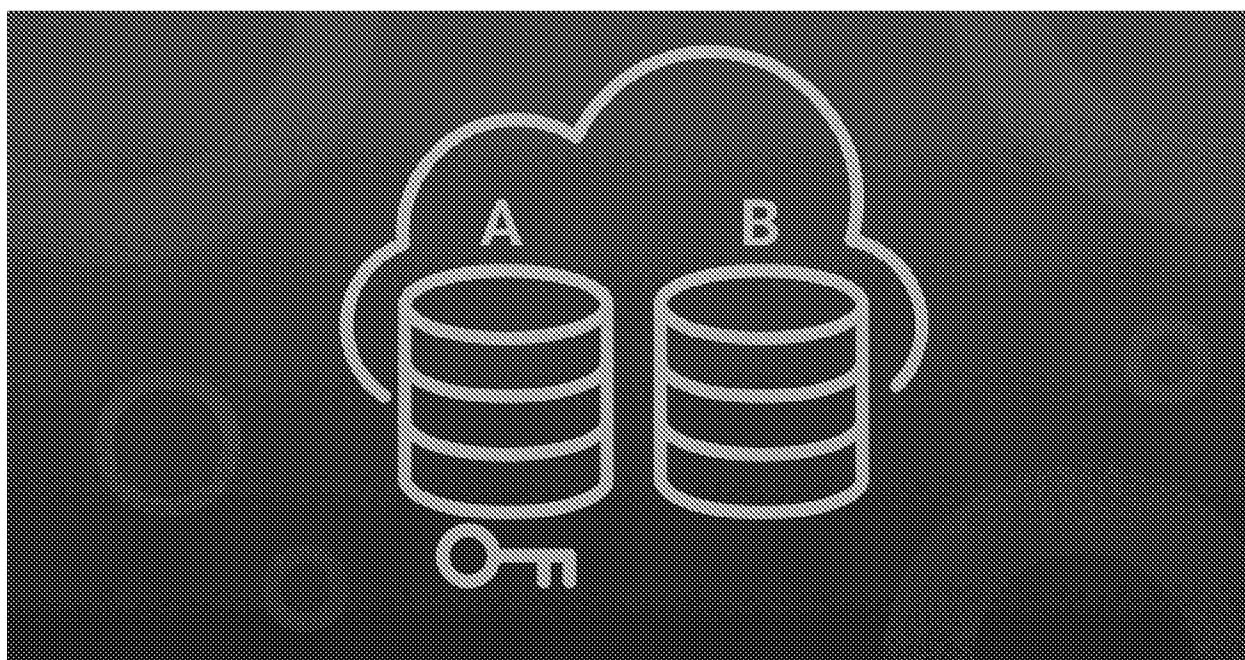
[0077] A difference exists between detecting *actual* lateral movement and detecting the *risk* of lateral movement.

[0078] *Actual Lateral Movement: Existing security tools can analyze the movement of network traffic between assets; anomalies can be indicative of malicious intent.*

[0079] Risk of Lateral Movement: Information stored in a manner that, if discovered, could be leveraged to access other machines or allow a further foothold in an environment.

[0080] Existing solutions can detect actual lateral movement, but can't detect private keys, passwords in shell history, or other information that could be leveraged by an attacker to move laterally on your environment.

[0081] Consider the following example. Servers A and B never communicate with one another, but A has a key that allows root access to B.

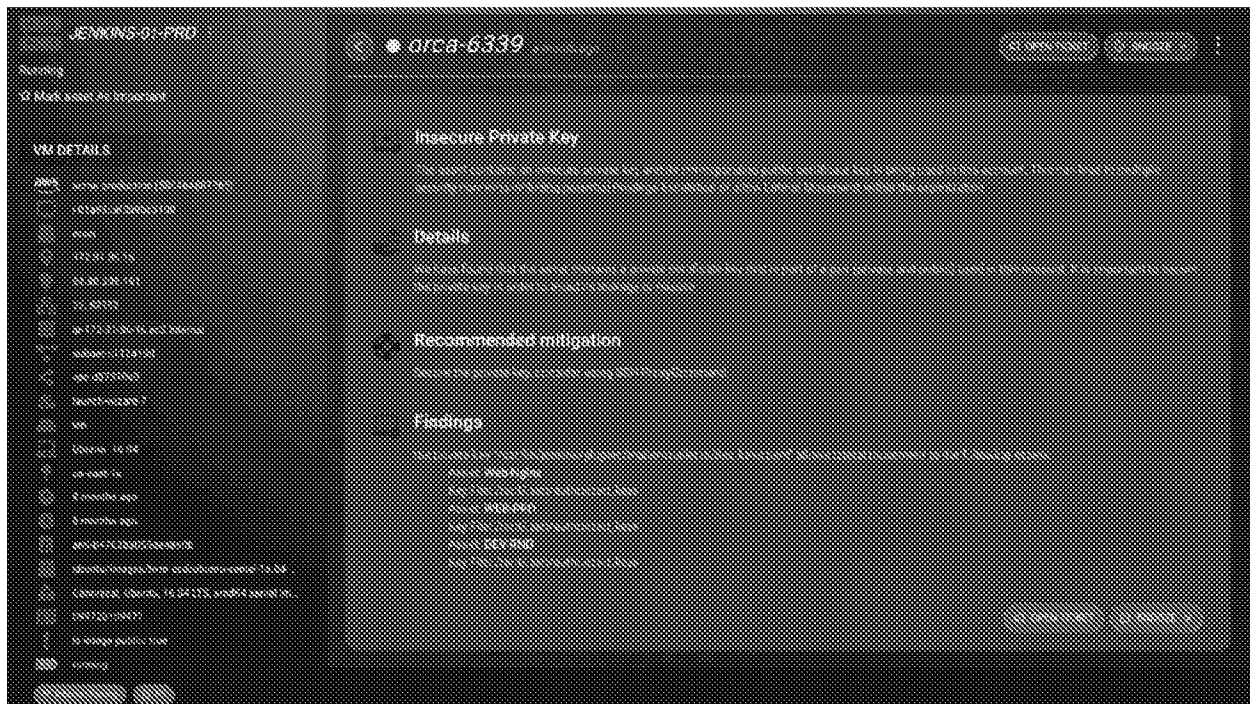


[0082] Existing tools would inform you there is no lateral movement. They would be correct because there is no traffic between the two machines. But they would completely miss an insecurely stored SSH key on server A that provides root access to server B.

[0083] This poses a lateral movement risk. For an attacker to stumble upon an insecurely stored key is like finding gold. They now test to learn which machines the key provides access to and voila, they now have access to both A and B.

[0084] To prevent an attacker from moving laterally, it's essential to know where sensitive information, such as insecurely stored keys, is located across your entire cloud environment.

[0085] Orca Security scans your entire cloud account in a holistic manner. It peers into each machine's filesystem for private keys, such as on server A, taking a hash of each found. It then checks the authorized key configuration across all other assets for public keys. If any are found, such as on server B, Orca calculates hashes and compares the private key hash to find a match. If there is, Orca sends an alert as seen in the image below:



[0086] From the image above we see that Orca provides: The path to the insecurely stored key: /var/lib/jenkins/plugins/mailer/mailer.js/dev_keys.pem. A recommended solution. The identity of the machines the key allows access to e.g., (Web-Nginx, WEB-PRD, DEV-RND). The user account on the machine in this case (root user for all machines in this example). Other lateral movement risks Orca finds are: Passwords in Shell History. Sensitive Info in Git Repository. Sensitive AWS Keys on Systems.

How Orca's Cloud Security Solution Detects Weak Passwords

[0087] Weak passwords are detrimental to any organization's security. This holds for commonly used passwords, passwords that are complex but are reused across services and applications, and even highly secure passwords that have been leaked

[0088] We strongly advocate for strong authentication methods on top of passwords, such as biometrics and multi-factor authentication, to boost password security.

[0089] Having said that, to be effective, these methods must be applied consistently throughout the entire organization. And enterprises need a way to verify that this is the case.

[0090] Often, enterprises will put tremendous effort into ensuring that their main services are secured correctly with multi-factor authentication and certificates while neglecting password security best practices when it comes to 'side services.' Often, the users of these services and applications are left to their own devices, and as a result end up relying on plain passwords and practicing poor cyber hygiene.

[0091] Password-based attacks come in many shapes. The following common types of attack can be prevented with stronger authentication methods.

[0092] Brute force attack: To break into a specific account, a hacker uses a script to attempt to login with all possible password combinations. Brute force accounts don't start randomly; they begin with the easiest-to-guess and most commonly used passwords.

[0093] Password Spraying: A more opportunistic type of attack, password spraying attempts to break into any account from a large number of accounts matching the known usernames with a few commonly used passwords.

[0094] As a result of massive stolen credential dumps that have been happening recently, high-volume password-spraying campaigns attempting to break into companies' cloud accounts are becoming increasingly common.

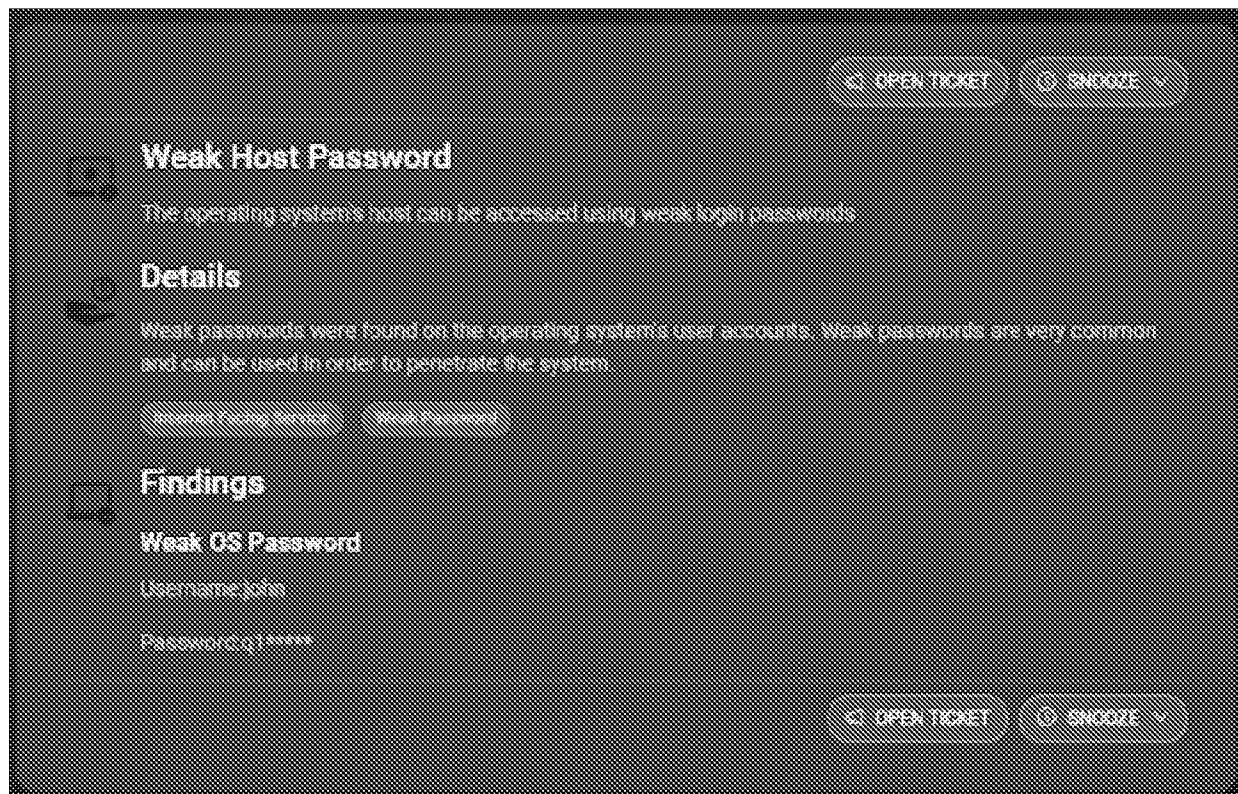
[0095] Password-spraying attacks are extremely dangerous because they often allow hackers to break into accounts without being locked out or triggering an alert to the IT team. Because password-spraying attacks don't generate an alarm or lock out a user account, a hacker can continually attempt to match usernames with common passwords until they succeed.

[0096] Credential Stuffing: This attack takes advantage of the fact that many users reuse login credentials across services. An attacker will use leaked username/password combinations from an unrelated service, attempting to login to the company's cloud account with the same credentials. The attacker is hoping that some fraction of leaked credential users and the targeted organization will have some users in common who are guilty of reusing passwords.

[0097] Credential stuffing is challenging to detect. Often, the only indication the victimized company has that they are being attacked is the rise in the overall volume of

login attempts. Even then, the company will have difficulty stopping these attempts without impacting the ability of legitimate users to log in to the service.

[0098] When successful, all of the attacks mentioned above can be extremely detrimental to a cloud-based organization.



[0099] To stay one step ahead of the hackers and bolster password security, Orca's patent-pending SideScanning™ technology accesses the organization's block storage (e.g. block storage volumes of individual workloads). It then attempts to match discovered passwords to commonly used passwords and known leaked password databases. For common passwords, we also perform offline brute force attacks to test how easy it is for a threat actor to break into the company's accounts.

[00100] Next, we perform a fuzzy search on the username in our extensive leaked password database. For the fuzzy search, we use the Damerau-Levenshtein edit distance algorithm.

[00101] By fuzzy search, we mean, for example, that for the username johndoe@acme.com, we will try to match passwords from john.doe@acme.net, johndoe@gmail.com and so on. All of the passwords we discover will be checked against the hashed password to see if it matches. We then send an alert to the customer of their breached password and location.

[00102] How does Fuzzy Search Work? The primary metric for fuzzy search is edit distance. For example: “cat” and “bat” are edit distance 1 because we only need to substitute c→b. “run” and “burn” are edit distance 2, insert “b” and swap u & r.

[00103] We deploy fuzzy search instead of exact search because many users use similar but not identical usernames and passwords across services and accounts. This happens due to availability, human cognitive limitations, as well as company policies. For example, the same user might technically not “reuse” an identical password across services and have a relatively strong password, while leaving their account vulnerable and easy to crack, like, Victory@19 and Victory@20.

[00104] Our extensive database of leaked credentials contains passwords, usernames, emails, and nicknames. To detect vulnerable passwords, we specifically use the Damerau-Levenshtein distance algorithm. The algorithm measures how many edits (deletion, inserts, substitutions, and transpositions) are required to transform word1 to word2.

[00105] Our fuzzy matching breaks down the target identifiers into words, and then it tries to match those to the target words. For example, if we had an entry in the database for john.doe@gmail.com, we will break it into “john,” “doe,” “gmail,” “com.” We will try to match to “john” “doe” but not to “gmail” & “com” since they are not unique enough.

[00106] Solving the password reuse problem with Orca. Thanks to SideScanning™ technology and Fuzzy Search, we can discover vulnerable passwords that are not necessarily an exact match. We locate similar credentials and notify our customers when those passwords need to be strengthened or if they were involved in an outside breach.

[00107] Orca’s intelligent alerts give users a better recommendation for their password security, providing a roadmap to tighter security posture.

[00108] Real-world examples: An Orca Security customer was running a vulnerable Internet-facing file server that was used to communicate sensitive data with clients. Unfortunately, the admin password for the service was an easy to crack password and username combination. Orca’s SideScanning™ technology located the weak credentials and suggested that the customer change their server passwords in order to secure their data.

[00109] In another case, a customer had a bastion server used by employees, which utilized SSH on their production servers. One of the employees had his personal email credentials leaked. Orca’s SideScanning™ technology was able to match the password to the employee, thereby preventing compromise to the organization.

[00110] We've seen that these serious security policy violations happen far more frequently on services where the rest of the security guidelines aren't followed, and therefore, miss the integration and installation of agent based security tools. SideScanning™ scans every single workload with no per asset integration, which enables organizations to see into their entire cloud infrastructure.

Not all Risks are Equal: Why Context Matters in Cloud Security.

[00111] Provisioning assets in the cloud has never been easier and with this, Gartner estimates that 90% of organizations will share sensitive data due to uncontrolled public cloud usage over the next six years. As deployments grow more complex, so too does the risk of exposing assets and data to the outside world.

[00112] Popular security scanners that were designed for the pre-cloud era and deployments, scan each machine separately – and therefore lack the ability to detect, analyze and alert users, especially when there are multiple machines at play. CSPMs, on the other hand, look at the entire picture, but completely miss the finite details as they are unable to provide in depth visibility.

[00113] How Orca Security Prioritizes Your Security. Cloud security isn't just a matter of scanning individual assets, but understanding the connection between different assets in order to understand what's important and what's not.

[00114] Orca Security is the only solution with SideScanning™ technology that drills into all four layers of the cloud stack: the infrastructure layer, operating system (OS) layer, application layer, and the data layer. Our platform integrates directly into the cloud environment to assess running workloads for vulnerabilities, misconfigurations,

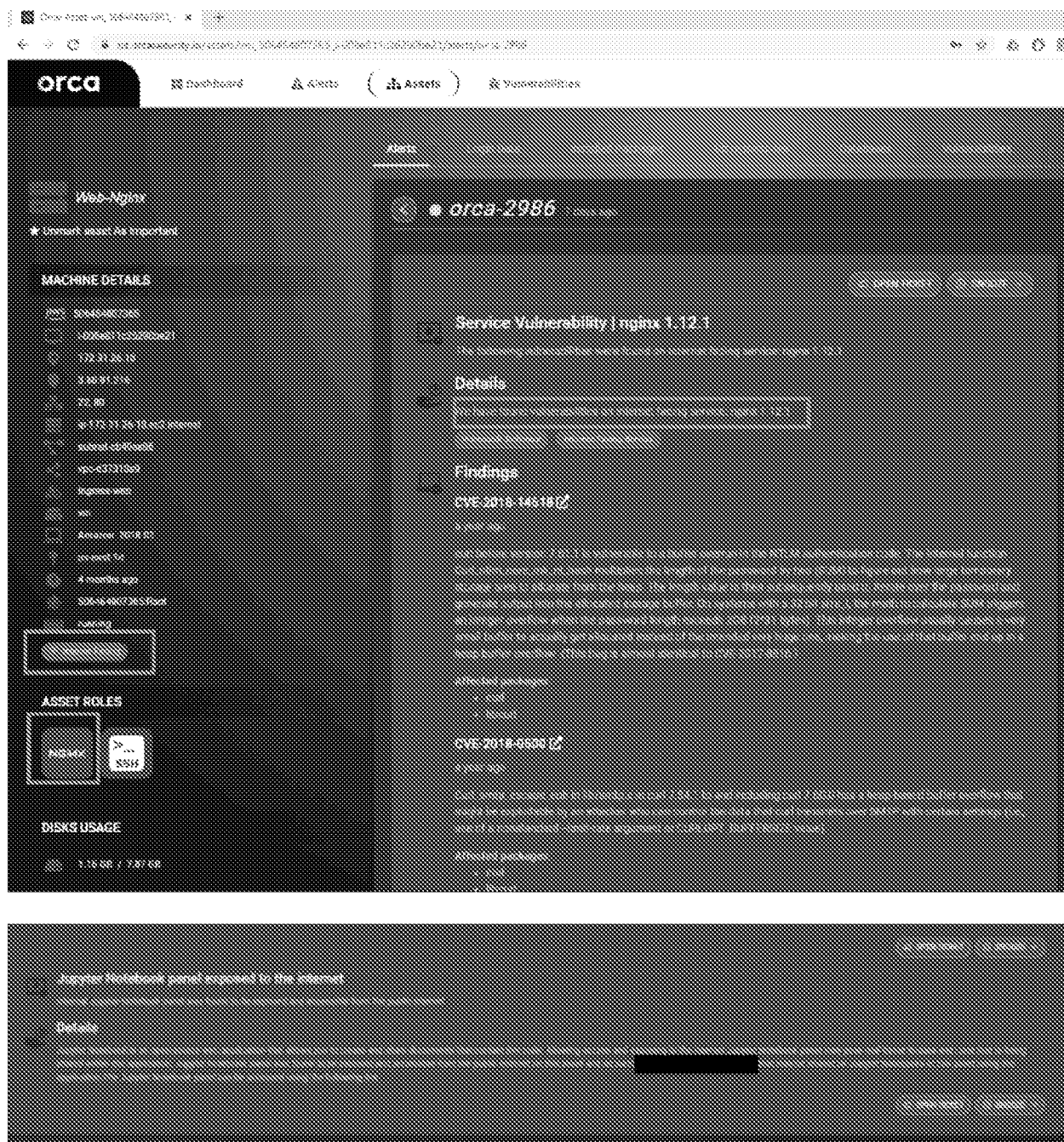
and breaches. And we go even further than alerting on configuration issues; we evaluate the cloud as a whole in order to prioritize potential harm.

[00115] How do we do this? First, we scan your infrastructure's configuration and network layout to build a diagram similar to a first-generation Cloud Security Posture Management (CSPM). Our SideScanning™ technology analyzes virtual machine disks, databases, datastores, and cloud logs to create an inventory of each asset across all four layers. We also scan resources including network services, proxies, load balancers, and Kubernetes clusters. Using this data, we perform a full assessment of the asset's security state to determine its vulnerabilities and risks, including whether the asset is publicly exposed.

[00116] By combining data collected from your infrastructure and control planes, Orca Security builds a unified view of your assets in order to detect and prioritize vulnerabilities. Orca operates independently of your assets, providing full-stack visibility even for stopped machines. And unlike other solutions such as agents, it does so without affecting performance.

[00117] Using Orca Security to Scan a Real-World Cloud Deployment. See, for example, this simple case of a web server at risk from a remote code execution vulnerability. In this case, the system automatically determines the alert is at an imminent compromise severity, as the machine is internet-facing and this vulnerability can be exploited at any minute. This same vulnerability on an internal-facing machine would get a lower 'hazard' scoring.

[00118] Traditional security scanners will give all machines with this vulnerability the same risk rating, regardless of their internet presence.



[00119] In the above example, the organization left a Jupyter notebook (a tool used for internal data science purposes) publicly accessible on the internet. This is a very bad practice, as this isn't a system that is designed to face internet borne attacks without a mediator. Nevertheless, the traditional approach of scanning 'inside' via agents won't find a thing – as the machine itself is configured correctly. Additionally, by

only looking at the ‘overall’ picture (as a CSPM does), no malicious activity will be detected as the CSPM is blind to what’s contained inside the workload. Only the combination of both deep and wide visibility can detect such a basic (yet critical) risk.

[00120] Risks must be taken within context; the same issue has dramatically different implications based on the environmental map. Any approach that depends on humans to prioritize alerts based on their knowledge of the organization will fail.

[00121] Orca provides in-depth scanning of your entire cloud infrastructure, immediately alerting you of the most important risks.

Unlocking the Key to the Cloud

[00122] Managing keys to assets and data in cloud deployments have become a key concern for many organizations. As explained by Kenneth Hui, a Solutions Architect at Rubrik, paraphrasing Kerckhoffs Principle, poor key management is “like having a state of the art anti-burglar system for your home but leaving the house keys and alarm codes under the welcome mat.”

[00123] Keys provide access to countless cloud resources, from SaaS services to SSH servers. Compromised keys can be used to access encrypted data, reissue digital certificates, or even steal cars. Keys provide a great deal of security and convenience, but can become extremely dangerous in the wrong hands. We’ll explore the risks of storing cryptographic/access keys insecurely, and how you can protect yourself with Orca. At the end of the day, an improperly saved key is usually the difference between a contained incident, should one machine be breached, to a full blown data leak which makes the morning news. The Keys to the Kingdom

[00124] At its core, a cryptographic key is a string of bytes used to decrypt data or authorize access to a resource. They allow you to secure and restrict access to cloud assets, even if those assets are accessible from a public network. However, keys commonly leak due to accidental exposures, misconfigurations, or other means. They must be stored securely so that they can be used without becoming vulnerable to attackers.

[00125] One of the top challenges in key management is verifying that in fact keys are stored ONLY where they should be. Keys can be found in several locations, including:

[00126] File systems: keys are often stored as plain text files on servers and cloud storage buckets.

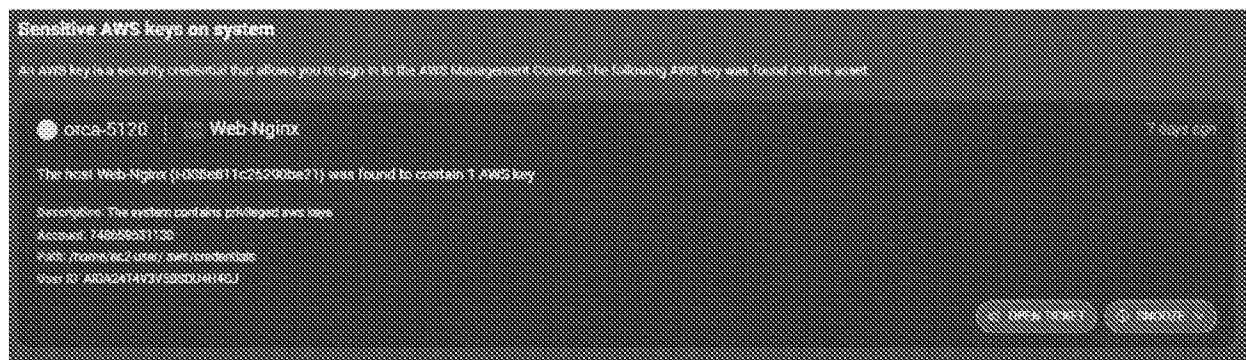
[00127] Key Management Services (KMS): vaults such as AWS KMS, Azure Key Vault, or Google KMS can be misconfigured, exposing keys beyond the original intent.

[00128] Source code repositories: developers may accidentally check keys into source control and push it to tools such as GitHub, GitLab, or BitBucket.

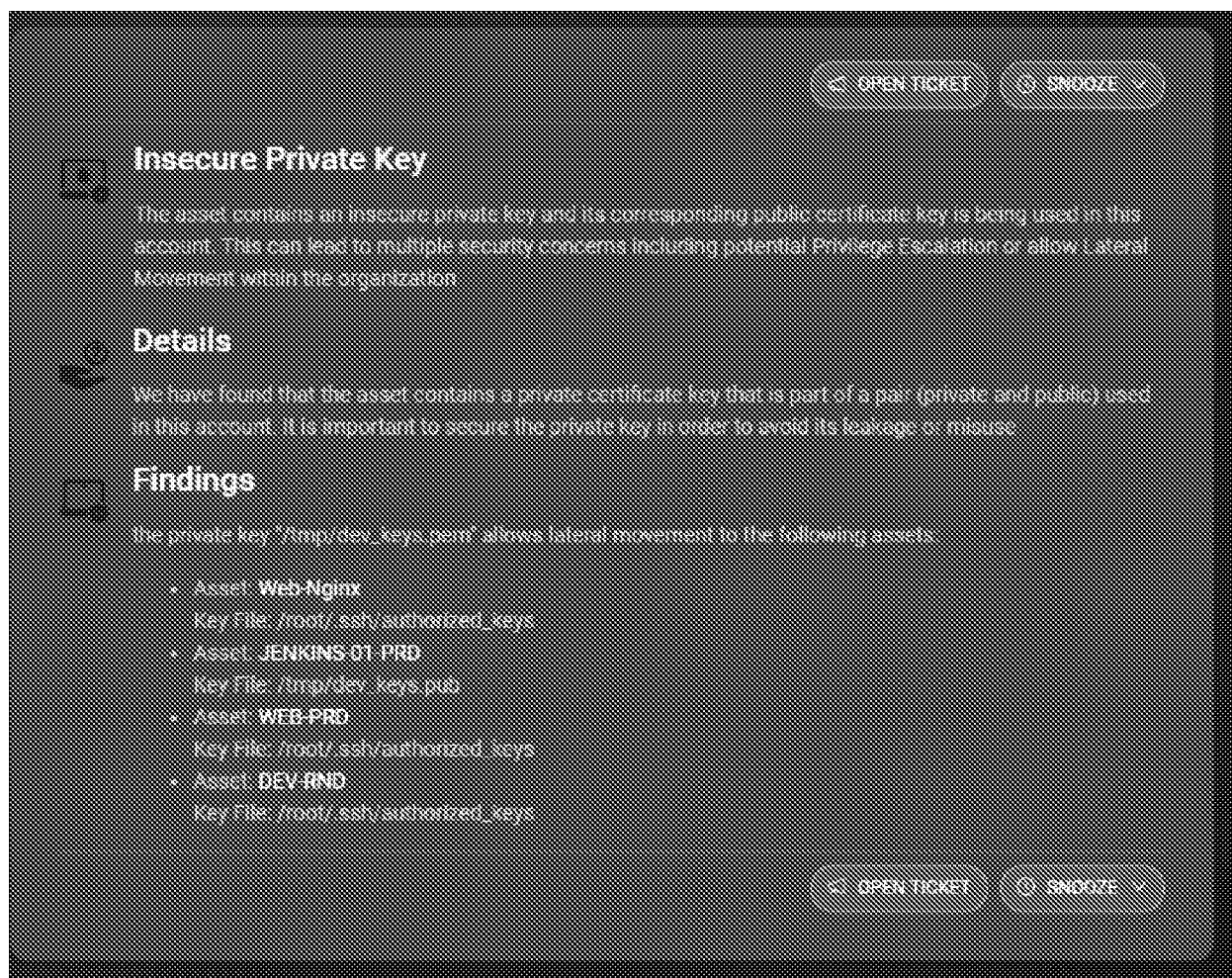
[00129] Developers also tend to leave keys in semi-hidden locations such as file archives. While these might mask keys from superficial scans, a more dedicated attacker can easily extract them and use them for lateral movement. In addition, keys have a variety of forms. A private SSH key is significantly different from a GitHub personal access token, but both could be used to wreak havoc on your applications and infrastructure.

[00130] One Small Key Can Cause Big Problems. It only takes one stolen key for an attacker to cause mayhem. For example, an exposed SSH key can give attackers

access to internal servers, potentially granting access to private data or the ability to move laterally to other resources. An exposed service account key for a cloud platform can allow attackers to provision new services, control infrastructure, steal data, or perform any number of malicious actions. Any proper security strategy must assume that some assets will be breached, and ensure that if this happens, the attacker will not be able to access all of the organization crown jewels.



[00131] To build a strong defense, you need to think like an attacker. Defenders tend to think in terms of checklists, while attackers think in graphs. They find an opening and leverage it to access connected insecure resources. All it takes is one small opening, a “weak backdoor”, and a swath of other systems are immediately at risk.



[00132] The most effective way to detect insecure keys is with a comprehensive security solution like Orca Security. Orca scans your entire environment for exposed keys and not only identifies where they're located, but which assets are put at risk as a result. We will automatically alert you when keys that can be used to perform malicious actions are detected, but haven't yet been revoked. Time is of the essence when it comes to insecure keys and our goal is to provide you with the information you need to prevent a breach.

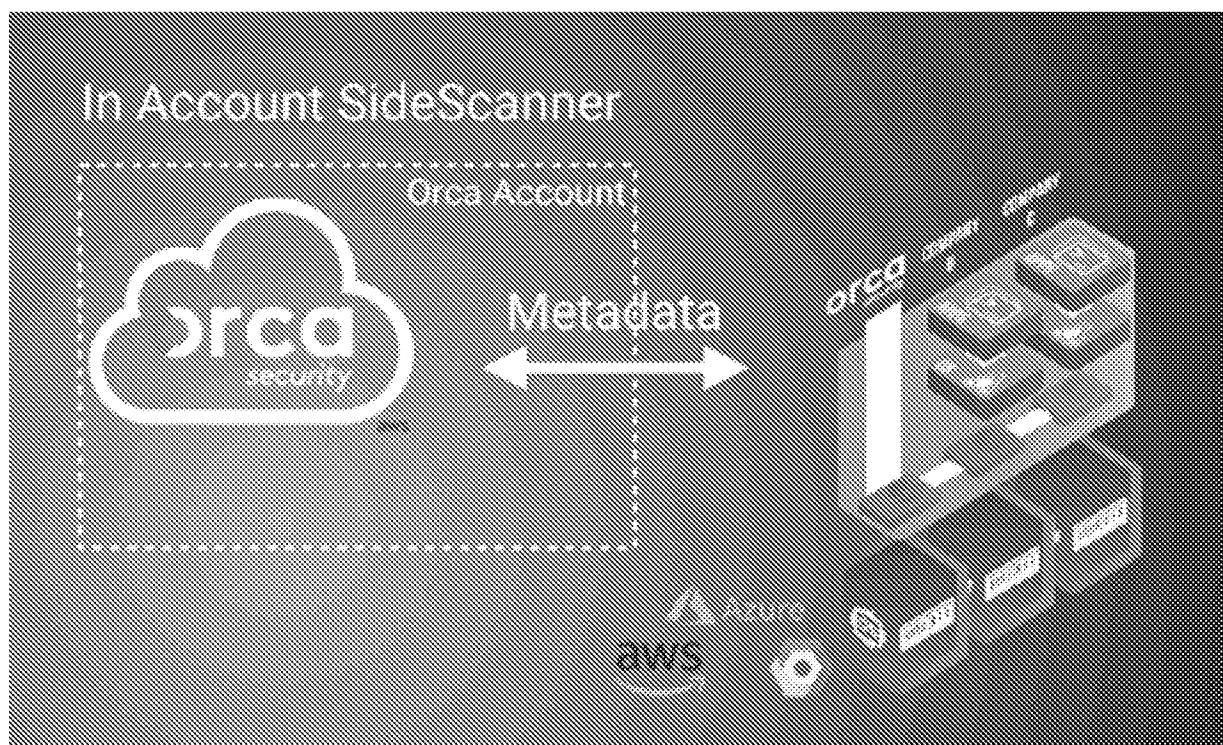
[00133] Securing keys is not an easy task. Like any other form of private data, keys must be managed and stored securely to prevent abuse and theft. Orca reduces the potential impact of poorly managed keys by performing in-depth scans of your

environment, immediately alerting you to exposed keys, and determining which resources are placed at risk.

OrcaPod

[00134] Orca Pod (In account scanning) deployment allows the Orca Sidescanner to run inside your AWS/Azure/GCP account (e.g., a primary account). In this mode, the Orca SAAS service will generate ephemeral scanners inside your account (VM's) that will be able to perform the same actions as the SAAS mode but logically hosted inside the account.

[00135] Data processing: In this mode, all of your raw data is processed in your account. The Orca SAAS scanner isn't granted access to raw data (VM's, buckets, DB's). The side scanner sends metadata only for analysis on the Orca Security platform backend, which analyzes the metadata, correlates it to assets and alerts, and produces security posture visibility.



[00136] 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.

[00137] 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.

[00138] 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.

While the foregoing description fully enables the following claims, the following claim concepts and description are to be considered part of the written description as well as reflecting additional subject matter possessed, enabled, and claimed.

What is Claimed is:

A. Concept # 1: Foundation of SideScanning

Some claimed technology uses an out-of-band process to reach cloud workloads through the runtime storage layer, combining this with information gathered from cloud provider APIs.

1. Exemplary Independent Claim

1. A system for inspecting data, the system comprising:

at least one processor configured to:

establish a trusted relationship between a source account in a cloud environment and a scanner account;

using the established trust relationship, utilize cloud provider APIs to identify workloads in the source account;

use cloud provider APIs to query a geographical location of at least one of the identified workloads;

receive an identification of the geographic location;

use the cloud provider APIs to access block storage volumes of the at least one workload;

determine a file-system of the at least one workload;

mount the block storage volumes on a scanner based on the determined file-system;

activate a scanner at the determined geographic location of the block storage volumes;

reconstruct from the block storage volumes a state of the workload; and

assess the reconstructed state of the workload to extract insights.

2. Exemplary Dependent Claim Concepts

- wherein the geographic location includes an identifier of a site
- wherein mounting includes selecting a driver corresponding to the determined file system.
- wherein the at least one processor is configured to, prior to activating, dynamically deploy a scanner at the geographical location
- wherein the identification of the geographic location includes a data center
- wherein the identification of the geographic location is a region
- wherein the identification includes at least one of a data center name, IP address, name of the cloud provider, a unique ID
- wherein a state of the workload includes at least two of an indication of installed applications, versions of the installed applications, operating system configurations, application configurations, profile configurations, logs, or database content
- wherein the environmental model includes an identification of installed applications, versions of the installed applications, operating system configurations, and file system content
- wherein the environmental model is an evolving model
- wherein the insights include vulnerabilities
- wherein the insights include composition of installed software

B. Concept #2: Vulnerability Management

Some claimed technology performs a comprehensive cyber security scan of the block storage volume. In performing a vulnerability assessment, it extracts the OS packages, libraries, and program language libraries such as Java archives, Python packages, Go modules, Ruby gems, PHP packages, and Node.js modules. It gathers library versions and other identifying characteristics, and in a later phase tries matching them to known vulnerabilities in its vulnerability database.

1. Exemplary Independent Claim

2. A cyber security system for a cloud environment, the system comprising:

at least one processor configured to:

using a cloud provider API, access a block storage volume of a workload maintained in a cloud storage environment;

identify an installed software application in the accessed block storage volume;

analyze the identified installed software application to determine an associated software version;

access a data structure of known software vulnerabilities for a plurality of versions of software applications;

perform a lookup of the identified installed software version in the data structure to identify known vulnerabilities;

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;
and

use the network accessibility information and the identified at least one port to identify vulnerabilities susceptible to attack from outside the workload.

2. Exemplary Dependent Claim Concepts.

- wherein querying the cloud provider API includes examining cloud-based data sources associated with the workload
- wherein network accessibility information includes data from an external data source, cloud provider information, and at least one network capture log, from which accessibility may be deduced
- wherein the at least one processor is further configured to extract OS packages, libraries, and program language libraries
- wherein installed software includes scripts
- wherein the data structure includes aggregated vulnerability data
 - wherein the aggregated vulnerability data includes data from third-party vendors
 - wherein some of the aggregated vulnerability data is collected by a scanner
 - wherein the aggregated vulnerability data includes at least one of advisories, exploits, security announcements and known bugs

C. Concept #3: Prioritizing Internet-Accessible Workloads for Cyber Security

Some claimed technology performs cyber risk prioritization analysis on workloads indirectly accessible from the internet. In doing so, it highlights bugs or other configuration risks that might only be exploitable from internal machines yet still facilitate an attacker's lateral movement.

1. Exemplary Independent Claim

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

at least one processor configured to:

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;

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

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

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

integrate the identified services into the graph;

traverse the graph to identify at least one vector originating in the internet and reaching the workload; and

output a risk notification associated with the workload.

2. Exemplary Dependent Claim Concepts

- wherein the plurality of entities includes a network appliance
- wherein the 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 a graph includes identifying individual entities as nodes, and interconnecting the nodes
- wherein the graph includes port numbers associated with the workload
- wherein the graph includes a path an attack path to the workload
- wherein risk notification is categorized based on the networking configuration
- wherein risk notification includes resolution recommendations

D. Concept #4: Malware Detection Using Secondary System

Rather than run malware detection on a primary system that itself may be susceptible to malware, Some claimed technology runs its malware detection on a secondary system, out of reach of any malware that may reside on the primary system.

1. Exemplary Independent Claim

4. A cyber security system for a cloud environment, the system comprising:

at least one processor configured to:

deploy a cloud provider API to access a block storage volume of a workload maintained on a target account in a primary system of a cloud storage environment;

dynamically deploy a scanner at a location of the block storage volume and on a secondary system other than the target system;

scan the block storage volume for malicious code, using the secondary system;

identify malicious code based on the scan; and

output from the secondary system, a notification of a presence of malicious code in the primary system.

2. Exemplary Dependent Claim Concepts

- wherein the location of the block storage volume includes at least one of: the target account, a secondary system account, cloud provider account, a third party account
- wherein scanning the block storage volume includes scanning disk-backed memory
- wherein the secondary system is a virtual machine, container or serverless function

- wherein the secondary system has an operating system different from an operating system of the target account
- wherein scanning the disk backed memory occurs using the secondary system
- wherein the disk-backed memory includes a page file
- wherein the malicious code is a root kit undetectable by the target system
- wherein the disk-back memory includes a cache file
- wherein dynamically deploying a scanner includes suspending operation of the scanner after the scan of the block storage volume
- wherein dynamically deploying a scanner includes revising a pre-deployed scanner

E. Concept #5: Orchestration Layer Agnostic Scanner for Containers

Some claimed technology utilizes the low-level access to scan containerized environments regardless of the orchestration mechanism. When reaching a workload that includes containers, a container scanning component reconstructs the container runtime-layered file system, and recursively runs the scanning processes.

1. Exemplary Independent Claim

5. A system for container security assessment, the system comprising:

at least one processor configured to:

establish a trusted relationship between a scanner and infrastructure running an instance;

based on the trusted relationship, gain access to a block storage volume of the instance via a cloud;

analyze the block storage volume to identify at least one LFS;

virtually reconstruct the LFS;

identify a plurality of containers in the reconstructed LFS by analyzing data and metadata associated with the reconstructed LFS;

scan the at least one container to identify cybersecurity risks;

using the data and metadata, categorize the plurality of identified containers into a plurality of groups, where each container in each group shares similar security risks; and

output a report identifying the plurality of groups in association with a respective security risk.

2. Exemplary Dependent Claim Concepts

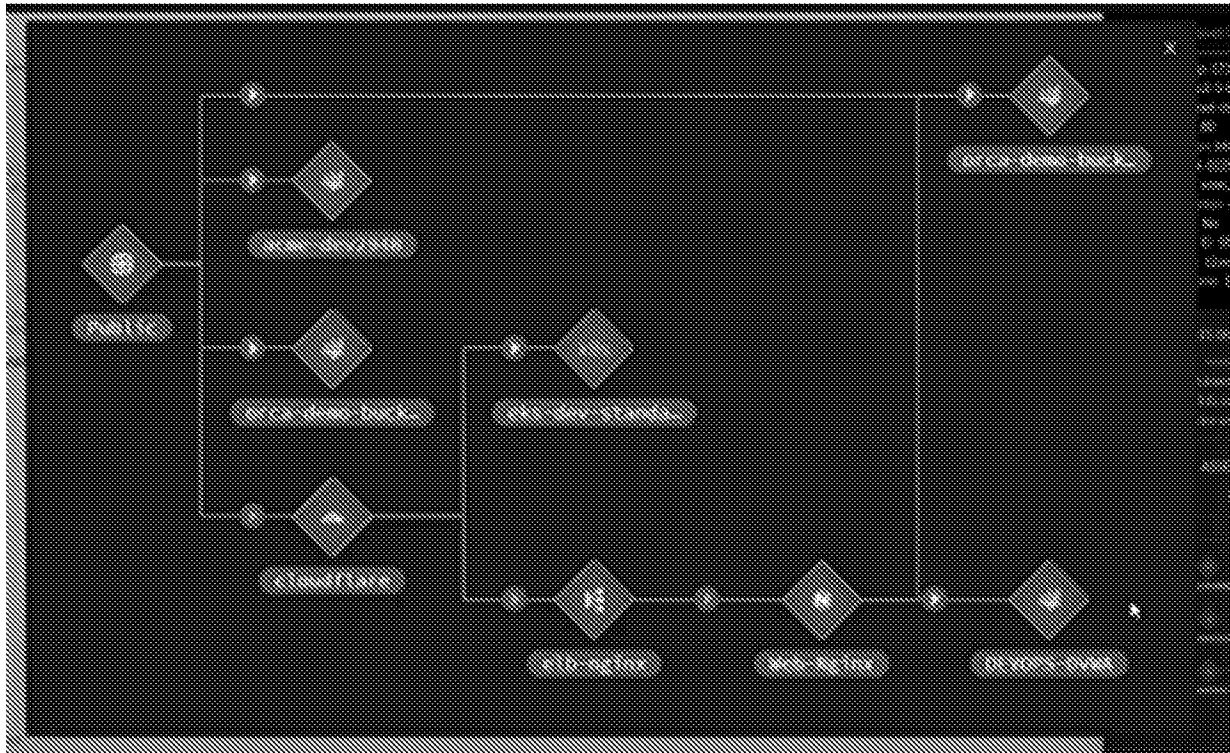
- Wherein the instance is a virtual machine
- Wherein the instance is a serverless compute instance
- Wherein the instance is a container

- wherein the at least one processor is further configured to detect deviations from baseline, including compromises of the environment
- wherein scanning occurs recursively
- wherein cybersecurity risks include at least one of: vulnerabilities, configuration, malware, lateral movement risk, or sensitive data

Note: LFS=Layered file system

F. Concept #6: Forward and Rearward Facing Attack Vector Visualization

Some claimed technology generates possible forward and backward attack vector visualizations for a cloud environment and generates a graphical user interface enabling users to visually identify both types of vulnerabilities.



1. Exemplary Independent Claim

6. 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 asset;

identify relationships between at least some of the assets, the relationships including at least one of trust, 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 identification to identify at least one possible internet-originating attack vector to the asset;

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; and

outputting signals to cause on a display a presentation of the forward and backward paths, thereby enabling visualization of the entry points and the lateral movement risks.

2. Exemplary Dependent Claim Concepts

- wherein assets in the cloud environment include at least one of: a virtual machine, a network appliance, a storage appliance, a compute instances, or engine instances
- wherein assets in the cloud environment include at least one of: identity and access management policies, organization policies, or access policies
- wherein the presentation of the forward and backward paths indicates alternative paths
 - wherein 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 the entry points at risk

G. Concept #7: Trigger-Based Incremental Cybersecurity Analysis

Some claimed technology monitors cloud change logs to act on any changes and contextualize associated risks discovered within a customer cloud account. It does so by examining differences between differing scans thereby reducing a load on computing resources while still providing a comprehensive scan.

1. Exemplary Independent Claim

7. A system for generating a composite cybersecurity risk assessment in a cloud environment, the system comprising:

at least one processor configured to:

access first scan information related to a primary asset group in a cloud environment to determine a first state;

monitor cloud change logs via a cloud provider API to determine at least one of 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 having communication privileges with the primary asset group, to determine at least one change in state;

upon detection of the at least one change in state, identify a delta between the first state and a second current state;

perform a supplementary scan on the delta to generate supplementary scan information; and

update the first scan information with the supplementary scan information to generate a comprehensive data set representing a composite cybersecurity analysis.

2. Exemplary Dependent Claim Concepts

- Wherein the first scan information includes at least one of vulnerabilities, configuration issues, malware location, sensitive data location, or lateral movement risks
- wherein monitoring cloud change logs includes performing incremental log analysis
- wherein the trust-related change in state includes at least one of a change in certificate authority, certificate expiration, a change in a certificate issuing authority, or a change in the certificate
- wherein the supplemental scan type is based on the determined change
- wherein cloud change logs are monitored continuously in real-time

H. Concept #8: Passive Key Identification

Some claimed technology gathers, for each of the scanned machines, the remote access keys installed there, as well as any keys that could provide access to other network resources. It looks for passwords and IT scripts containing passwords that could be used by attackers against the environment—as well as AWS keys, SSH keys, or other key types that provide unchecked access to important resources. In essence, it acts like a whitelisted attacker. Once it reaches a machine, it looks for what an attacker might search and enumerates that in a report.

1. Exemplary Independent Claim

8. A cybersecurity system for passively matching keys with compute resources, the system comprising:

at least one processor configured to:

analyze a cloud environment to identify a plurality of keys to the compute resources in the cloud environment;

passively perform cryptographic analysis on the plurality of identified keys to identify non-functional first fingerprints that uniquely identify each of the plurality of keys;

analyze trust configurations of the compute resources to passively identify a second set of fingerprints for each of the compute resources;
and

compare the first set of fingerprints with the second set of fingerprints to match keys with compute resources without using the keys to access the compute resources.

2. Exemplary Dependent Claim Concepts

- wherein the keys are stored in at least one of a workload or a data store
- wherein non-functional fingerprint characterizes the key without identifying the key
- wherein the keys include passwords or IT scripts containing passwords

- wherein the keys include at least one of a cloud key (e.g. AWS key) or an SSH key
- wherein the at least one processor is configured to test key validity to eliminate false positives
- wherein the at least one processor is configured to analyze multi-machine interaction using identified fingerprints
 - wherein analyzing multi-machine interaction includes comparing the first set of fingerprints with the second set of fingerprints
- wherein the at least one processor is configured to analyze multi-machine interaction using identified keys
 - wherein analyzing multi-machine interaction includes comparing keys identified on one machine with keys identified on another machine

I. Concept #9: Hybrid Deployment of Ephemeral Scanners

Some claimed deployment technology allows the cloud cybersecurity to run inside an AWS/Azure/GCP account. In this mode, the cloud cybersecurity service will generate ephemeral scanners inside the account that perform the same actions as the SAAS mode but are logically hosted inside the account.

1. Exemplary Independent Claim

9. A cybersecurity scanner deployment system, comprising:

at least one processor configured to:

access a primary account maintained in a cloud environment;

receive information defining a structure of the primary account, wherein the structure includes a plurality of assets, and wherein the information excludes primary account data;

deploy 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, the output excluding primary account data;

receive a transmission of the metadata from the ephemeral scanner, wherein the transmission excludes the primary account data;

analyze the received metadata to identify a plurality of cybersecurity vulnerabilities;

correlate each of the identified cybersecurity vulnerabilities with an associated one of the plurality of assets; and

produce a report correlating the cybersecurity vulnerabilities with the assets.

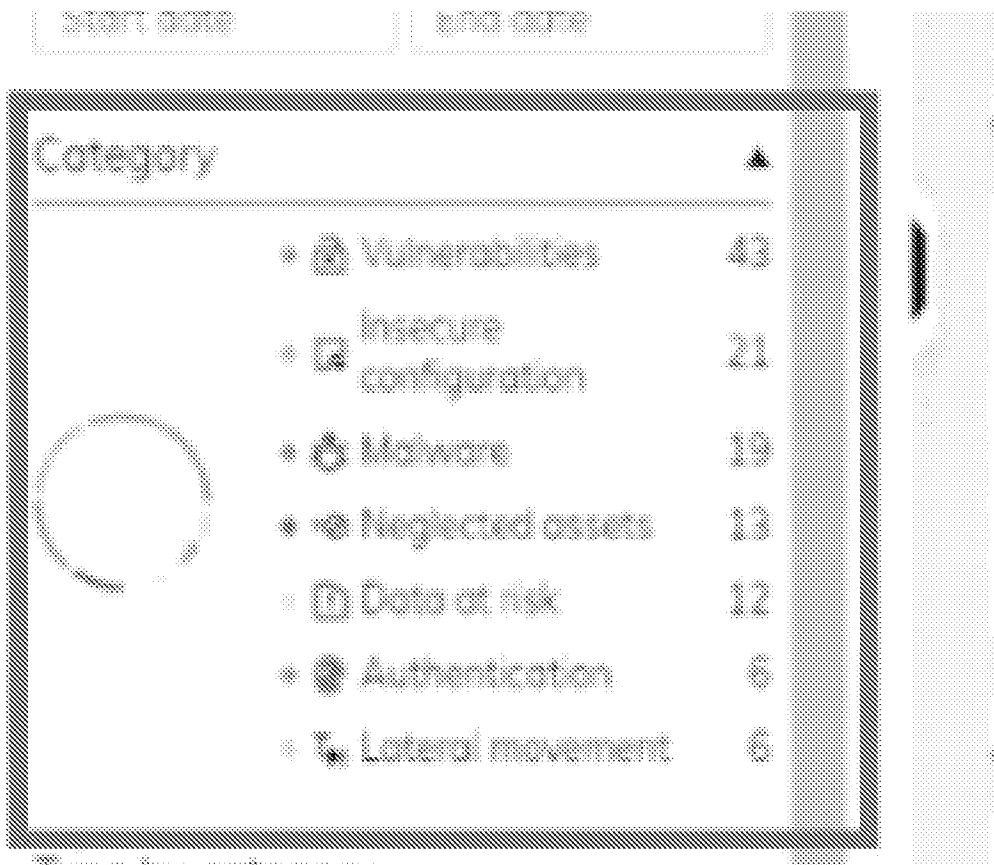
2. Exemplary Dependent Claim Concepts

- wherein the primary account is an account owned by a consumer

- wherein defining the at least on block storage volume includes presenting found risk data without sharing consumer data and data that was used to identify the risk data
- wherein the ephemeral scanner is configured to perform vulnerability scanning
- wherein the ephemeral scanner is configured to perform configuration scanning
- wherein the ephemeral scanner is configured to perform malware scanning
- wherein the ephemeral scanner is configured to perform lateral-movement risk analysis
- wherein the ephemeral scanner is configured to perform sensitive information scanning
- wherein the ephemeral scanner is configured to perform container scanning
- wherein the ephemeral scanner is configured to perform keys and password scanning

J. Concept #10: Cybersecurity Categorical Risk Level Analysis

Some claimed technology assesses whether malicious activity is actual or probable and assigns associated risk levels based on the assessment. Thereafter, it outputs a categorical risk level assessment.



1. Exemplary Independent Claim

10. A comprehensive cybersecurity risk analysis system, comprising:
- at least one processor configured to:
 - access scan information related to an asset group in a cloud environment;
 - analyze a state of the asset group to determine internet-originating vectors that extend to particular assets in the asset group;

receive scan-derived metadata associated with the particular assets, wherein the metadata identifies cybersecurity risks associated with the particular assets;

aggregate information regarding the internet-originating vectors and the scan-derived metadata to determine a risk level associated with each of the particular assets, wherein a first risk level corresponds to existing unauthorized access via an identified internet-originating vector, and wherein a second risk level corresponds to identification of a currently unused internet-originating vector to an asset previously identified as having a cybersecurity risk;

assign the first risk level to a first subset of the assets in the asset group;

assign the second risk level to a second subset of the assets in the asset group; and

present a categorical risk level analysis report identifying the first subset in association with the first risk level and the second subset in association with the second risk level.

2. Exemplary Dependent Claim Concepts

- wherein the at least one processor is further configured to assign to an identified cybersecurity risk a third risk level when the identified cybersecurity risk lacks an internet-originating vector
- wherein the at least one processor is further configured to assign to an identified cybersecurity risk a fourth risk level when the identified cybersecurity risk is informational in nature
- wherein the at least one processor is further configured to generate a user interface displaying a categorized risk analysis scheme

- wherein the user interface includes a filter menu based on at least one of category, asset type, risk type, region, or account
- wherein the user interface includes a description view for each identified risk
- wherein the cybersecurity risks include at least one of vulnerabilities, insecure configurations, malware, neglected assets, data at risk, lateral movement, or authentication
- wherein the user interface is configured to display a possible attack vector
- wherein the user interface is configured to provide a recommended mitigation tactic
- wherein the categorical risk level analysis includes at least one of risk level, risk type, or asset at risk
- wherein the categorical risk level analysis is configured to separate risks by category
 - wherein the categorical risk level analysis is further configured to identify at least one top risk in each category based on the assigned risk level

Notes:

- Possible Risk Level Examples:
 - Compromise - attacker already gained access,
 - Imminent Compromise - attack vector exists and can be used,
 - Hazardous - serious security implication, but no full vector exists,

- Informational - no action required, but need to know e.g. low space alert. Vector can be direct and indirect.

K. Concept #11: Cybersecurity Risk Timeline Visualization Interface

Some claimed technology provides a temporal interface for representing overall cybersecurity risk snapshots at different points in time. This enables users to interact with a timeline and quickly compare how overall risk changes over time. This permits management to assess the effectiveness measures taken.

1. Exemplary Independent Claim

11. A cybersecurity risk timeline visualization system, comprising:

at least one processor configured to:

collect scan-derived metadata related to an asset group in a cloud environment at a plurality of points in time;

collect data defining internet-originating vectors that extend to particular assets in the asset group at the plurality of points in time;

aggregate the scan-derived metadata and the data-defining internet originating vectors to generate a representation of the asset group for each of the plurality of points in time;

present an interactive interface control permitting navigation to specific varying points in time;

receive a selection of a particular point in time; and

display the representation of the asset group corresponding to the particular point in time.

2. Exemplary Dependent Claim Concepts

- wherein the interactive interface control is in a form of a slider
- wherein the representation includes an alert timeline
- wherein the interactive interface control is in a form of a calendar
- wherein the selection of the point in time causes the interactive interface control to visually emphasize the particular point in time.

- wherein the at least one processor is further configured to generate a report associated with the particular point in time
- wherein the at least one processor is further configured to cause a display of a delta between the particular point in time and an additional selected point in time
- wherein the at least one processor is further configured to collect inventory data

L. Concept #12: Dashboard Aggregates Disparate Risk Information of an Asset

Some claimed technology provides a dashboard that permits a user to select an asset, and to view on a common dashboard, the cybersecurity risks associated with that asset.

1. Exemplary Independent Claim

12. A graphical user interface system for providing comprehensive cloud environment risk inventory visualization, the system comprising:

at least one processor configured to:

present a plurality of asset categories;

receive a selection of a particular asset category;

present a list of assets in the selected category that have cybersecurity risks;

for each listed asset:

retrieve workload component cybersecurity risk information,
and

retrieve cloud component cybersecurity risk information;

present a common interface providing access to the workload component cybersecurity risk information and cloud component cybersecurity risk information; and

in the common interface, present an interconnection between the workload component cybersecurity risk information and the cloud component cybersecurity risk information.

2. Exemplary Dependent Claim Concepts

- wherein the plurality of asset categories include at least some of account, authentication, container, database, image, container, managed service, messaging service, monitoring, network, storage, users and access, VMs, serverless

- wherein the common interface is configured to display
 - asset type, risks, region, account
 - description for each identified asset
 - vulnerabilities, insecure configurations, malware, neglected assets, data at risk, lateral movement, authentication
 - possible attack vectors
 - provides recommended mitigation tactic
 - workload metrics
- wherein the at least one processor is configured to provide a cybersecurity report for each listed asset
- wherein the at least one processor is configured to identify a risk level distribution for the listed assets
- wherein the at least one processor is configured to receive a specific risk search query and to identify assets vulnerable to the specific risk.

M. Concept #13: Quality Control for Cybersecurity Risk Mitigation

A recurring problem in cybersecurity is that cybersecurity risk, although addressed, may not be fully mitigated, leaving open vulnerabilities thought to be resolved. Some claimed technology solves this problem by integrating with a ticketing system and initiating a scan on assets whose tickets are marked as being resolved. If the scan determines incomplete resolution, the ticket is returned for additional mitigation.

1. Exemplary Independent Claim

13. A bidirectional cybersecurity risk resolution ticketing integration system, comprising:

at least one processor configured to:

receive scan-derived metadata identifying at least one cybersecurity risk associated with an asset;

send to a ticketing tracker a request for resolution of the at least one cybersecurity risk associated with the asset;

receive from the ticketing tracker, metadata identifying a cyber security risk resolution status of the asset;

determine if the cyber security risk resolution status of the asset indicates at least partial resolution of the at least one cybersecurity risk;

upon the determination, initiate a post-resolution scan of the asset to assess mitigation of the at least one cybersecurity risk;

and

in an event post-resolution scan fails to confirm mitigation, update the ticketing tracker to reflect failed resolution.

2. Exemplary Dependent Claim Concepts

- wherein the at least one processor is configured to in the event of confirmed mitigation update the ticketing tracker to indicate a successful resolution of the at least one cybersecurity risk

- wherein the at least one processor is configured to communicate with the ticketing tracker via an API
- wherein the post-resolution scan corresponds to the identified at least one cybersecurity risk
- wherein the at least one processor is configured to introduce a waiting period if the post-resolution scan fails to confirm mitigation, and to re-run the scan after the waiting period
- wherein, when the post-resolution scan fails to confirm mitigation, identify in metadata a developer involved in the at least partial resolution, and to escalate a review request to a higher-level developer.
- wherein the at least one processor is configured to cause a display of an event timeline associated with the resolution.