SideScanning™ – Inside the Engine that Powers Orca Security

The cloud estate of an organization comprises 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.

Orca Security introduces a radically new approach that secures the entire cloud estate and helps meet compliance mandates, but without disrupting business operations in live environments.

Orca Security Ltd.
Exhibit 2067
Wiz v. Orca
IPR2024-00863, -00864, -00865, -01109, -01190, -01191
Ex. 2067-001

Table of Contents

Background

Traditional Scanning Methods

- Agent-based scanning
- Authenticated scanning
- Unauthenticated scanning

Cloud Security Posture Manager (CSPM)

Introduction to SideScanning™ - A Radical New Approach

How SideScanning™ Works

- Onboarding
- The SideScanning Process
  - The Control Plane Path
  - The Data Plane Path
  - Vulnerability Scanning
  - Configuration Scanning
  - Malware Scanning
  - Detecting Lateral Movement Risk, Exploitable Keys, and Weak Passwords
  - Sensitive Information Scanning
  - Container Scanning
  - Collector Teardown
- Combining Information, Analysis, and Reporting
  - Showing Alerts in Context
- Extending the map into containerized environments
  - Combining Low Severity Issues into Larger Alerts

In Summary

About Orca Security

Background

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.

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.

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.

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:

- Detect risks such as vulnerabilities, malware, misconfigurations, lateral movement risk, and unsecured sensitive data
- See cloud inventory at every layer: I/S, OS, applications, and data
- Discover and see previously missed assets

The engine that makes all this possible is called SideScanning™.

Traditional Scanning Methods

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 weren't reimagined for the cloud.

Agent-based scanning

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 don't want to be bothered with deploying

Ex. 2067-003

agents on VMs, in containers, and in serverless configurations—let alone dealing with their never-ending maintenance.

Authenticated scanning

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're 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.

Unauthenticated scanning

An unauthenticated scan can only examine publicly visible information and isn't able to provide detailed information about assets. It's 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 isn't linked from your main website. The site won't 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.

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's 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.

Orca's earliest customer engagements revealed that the average organization lacks security visibility into at least 50% of its cloud infrastructure footprint. This is mostly due to an organization's inability to keep up with the incredibly high TCO for agent deployment and maintenance.

Cloud Security Posture Manager (CSPM)

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

Ex. 2067-004

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.

Introduction to SideScanning™ - A Radical New Approach

Orca Security uses a novel patent-pending technology called SideScanning™. SideScanning is a radical approach because Orca doesn't 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 is able to provide deep and contextualized visibility of cloud environments. It covers 100% of an organization's assets with absolutely no agent or network scanner.

Orca Security requires 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. It does so while also reading into virtual machines, disks, databases, and datastores, as well as logs for all cloud assets. It then analyzes the data and builds a full-stack inventory. Next it automatically assesses the security state of every discovered asset throughout the technology stack, including all four cloud layers: I/S, OS, apps, and data.

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's 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.

Orca doesn't 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.

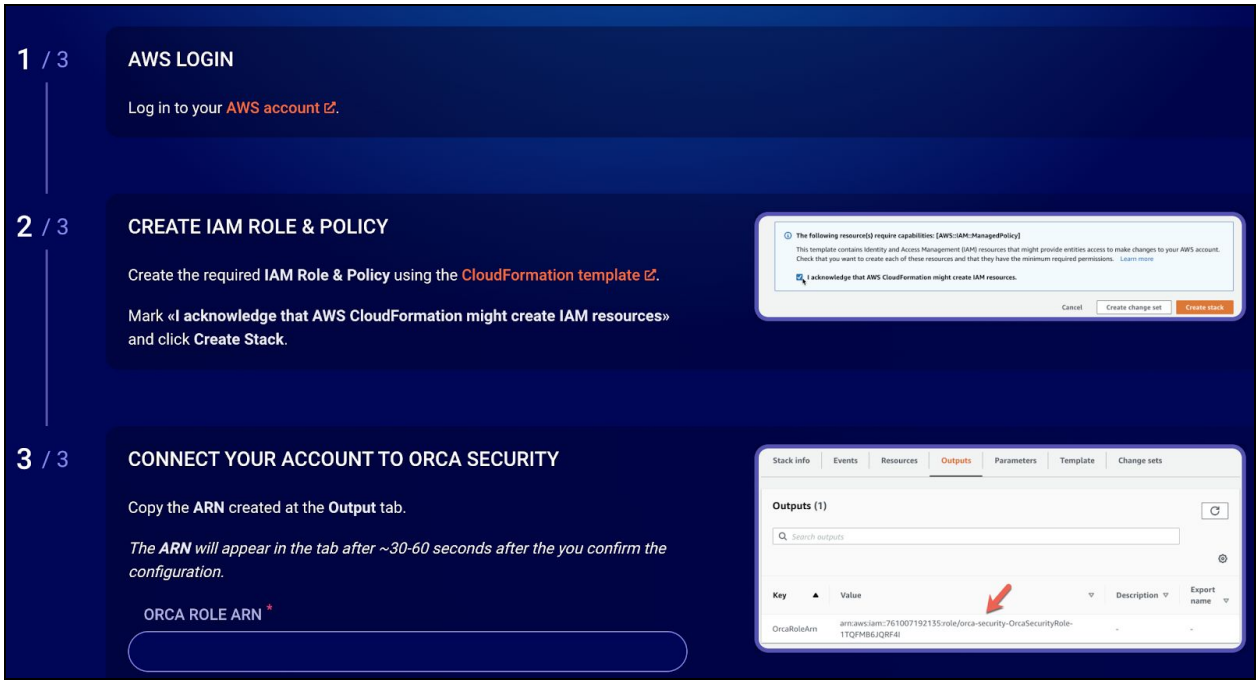
Ex. 2067-005

How SideScanning™ Works

Although this paper uses AWS terminology, readers can apply the same concepts to Azure and GCP.

Onboarding

Overview – Orca's onboarding process is simple and quick. You provide Orca with a role and establish trust between your account and Orca's production account. The role has a few permissions, the most important being read-only permissions and permissions to read the block storage layer. The entire process is encapsulated within a cloud formation template, which means that an administrator only has to 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. That's essentially it, and it takes but a few minutes.



Example: Three step installation on AWS

Permission Detail – Orca's read-only permission enables it to visualize and build a map of your 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.)

Ex. 2067-006

Orca requests permission to read encrypted key management service (KMS) volumes to open snapshots in that account. Orca doesn't copy the customer key, but rather uses it to re-encrypt the snapshots using its own key to continue the examination.

The entire process is quite fast. It's 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.

Ex. 2067-007

The SideScanning Process

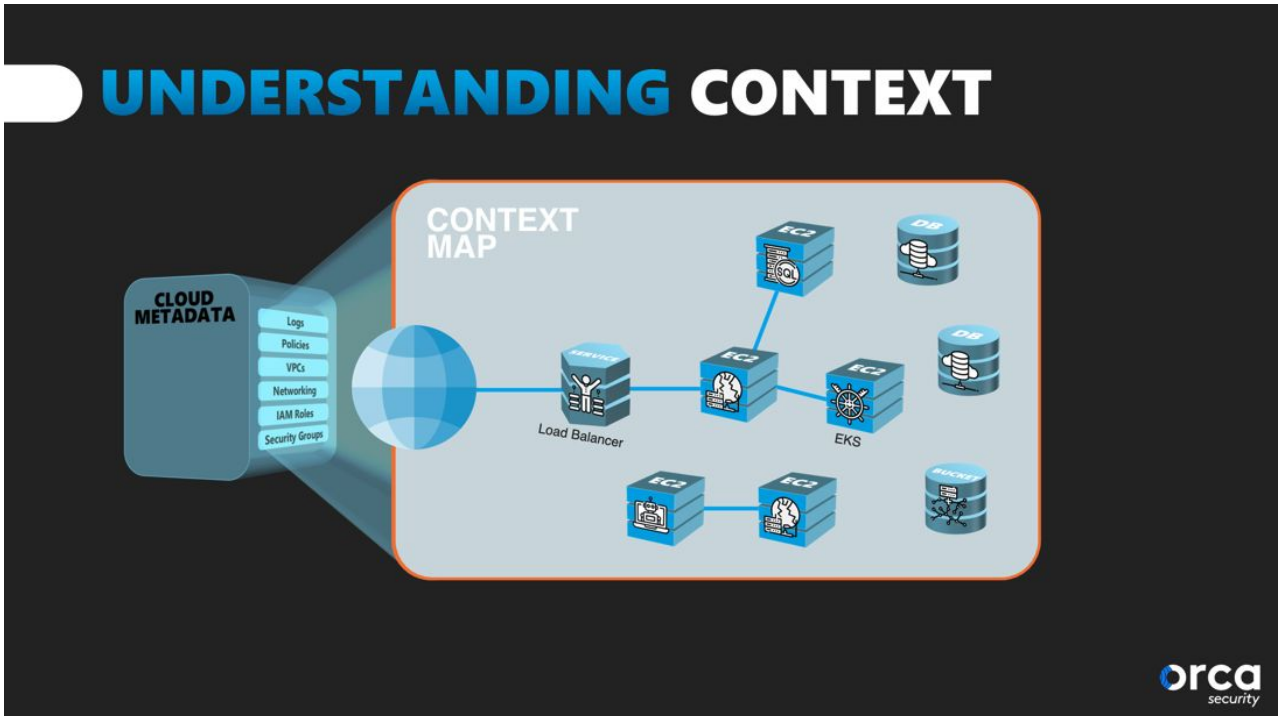
After you've added your account to Orca, the SideScanning™ process starts with building a map of your organization's entire estate. Every asset in the account is enumerated in all regions, including:

- 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

The Control Plane Path

After the cloud assets are enumerated, the SideScanning™ process splits 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 one of the most valuable features it provides, the map also enables Orca to contextualize assets and their associated risks discovered within the customer cloud account.

Ex. 2067-008



During this phase, Orca provides definitive alerts regarding:

- S3 buckets exposed to the world
- snapshots that have been published to the entire world
- other misconfigurations at the cloud control plane level

Orca's control plane path yields 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—a task Orca uniquely does in its second phase and is what constitutes its "secret sauce".

If an inspected machine is already infected by an advanced tool, the malware can't affect the scanning process because we never run the malicious application—we just look at it from a different machine. In this way, we are able to see rootkits that are invisible to host-based solutions.

Ex. 2067-009

The Data Plane Path

For each region in your cloud account, Orca enumerates all the possible compute assets, taking snapshots of their data volumes to share with the Orca production account[1]. 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 immediately deletes 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've done and runs several data collection steps:

- Vulnerability Scanning
- Configuration Scanning
- Malware Scanning
- Lateral Movement Risk
- Exploitable Keys and Weak Password Detection
- Sensitive Information Scanning
- Container Scanning

Orca doesn't run a clone of your workload. Rather, the collector rebuilds the correct configuration and mounts the volumes in their native file systems in Orca's collection machine. 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.

AWS and all other cloud providers allow Orca to create a snapshot of the disk state while the machine is running. The process can snapshot multiple volumes at the same point in time, resulting in deep and consistent visibility.

[1] This paper talks about Orca's default deployment mode - Full SaaS. Orca supports two additional deployment modes, which are described in this support article: https://orcasecurity.zendesk.com/hc/en-us/articles/360039718051-Deployment-Modes-for-Orca-

Ex. 2067-010

Vulnerability Scanning

In performing vulnerability assessment, Orca extracts all 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. Among others, this database includes aggregated vulnerability data from:

- NVD
- US-CERT
- OVAL – Red Hat, Oracle Linux, Debian, Ubuntu, SUSE
- JVN
- Alpine secdb
- Amazon ALAS
- Red Hat Security Advisories
- Debian Security Bug Tracker
- Exploit Database
- JPCERT
- WPVulnDB
- Node.js Security Working Group
- Ruby Advisory Database
- Safety DB(Python)
- PHP Security Advisories Database
- RustSec Advisory Database
- Microsoft MSRC, KB
- Kubernetes security announcements
- Drupal security advisories

Note: This list is current as of May 2020. More sources will be occasionally added.

Ex. 2067-011

Configuration Scanning

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 gathers 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 performs 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 runs the CIS Benchmarks on the workloads to check for misconfigurations.

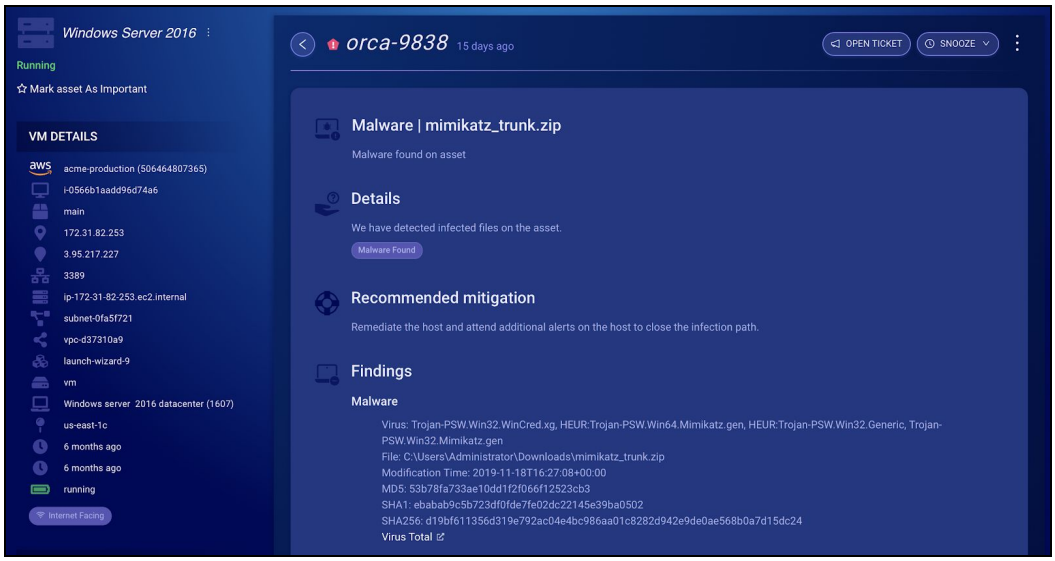
After taking snapshots, Orca is not accessing the customer's environment to derive any other security value. No customer resources are used at all—no disk, no RAM. Nothing.

Malware Scanning

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 won't detect polymorphic malware, but Orca does so with its deep scanning capabilities.

Malware scanning is performed on the Orca side, so it doesn't affect production workloads. Orca is free to devote significant compute resources to scanning because—unlike an agent—it isn't limited by the customer machine's CPU and available memory.

Another benefit of this approach is that the scanner can't be tampered by malware running on a machine, as the scanner runs on the Orca host. This enables Orca to detect advanced threats, such as rootkits that can easily circumvent other forms of detection.

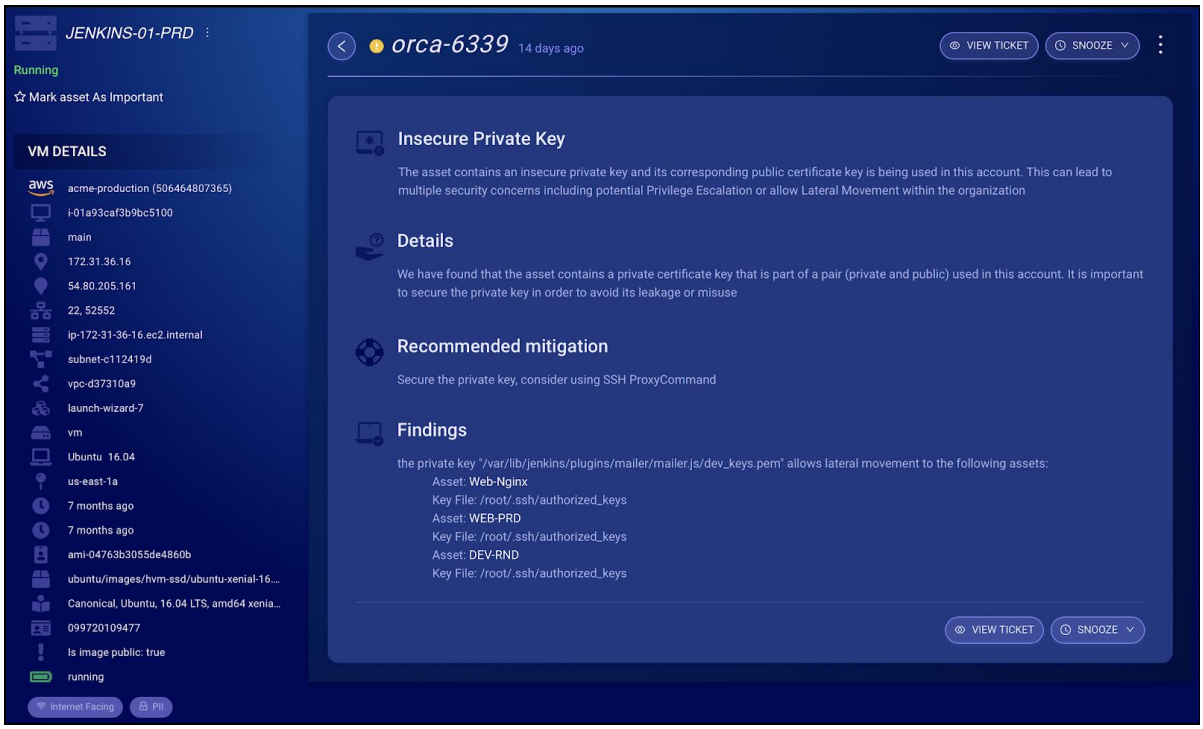


Example: Malware

Ex. 2067-012

Detecting Lateral Movement Risk, Exploitable Keys, and Weak Passwords

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.



Example: Insecure private key

For each scanned machine, Orca gathers all 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.

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. (Additional information on this topic is found in How Orca's Cloud Security Solution Detects Weak Passwords.) Orca makes note of the stored password along with a corresponding pointer to it, the password is not shared outside of its ephemeral collector.

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

Ex. 2067-013

private and public keys and to show where lateral movement is not only possible but quite trivial to accomplish.

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."

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.

Sensitive Information Scanning

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's 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.

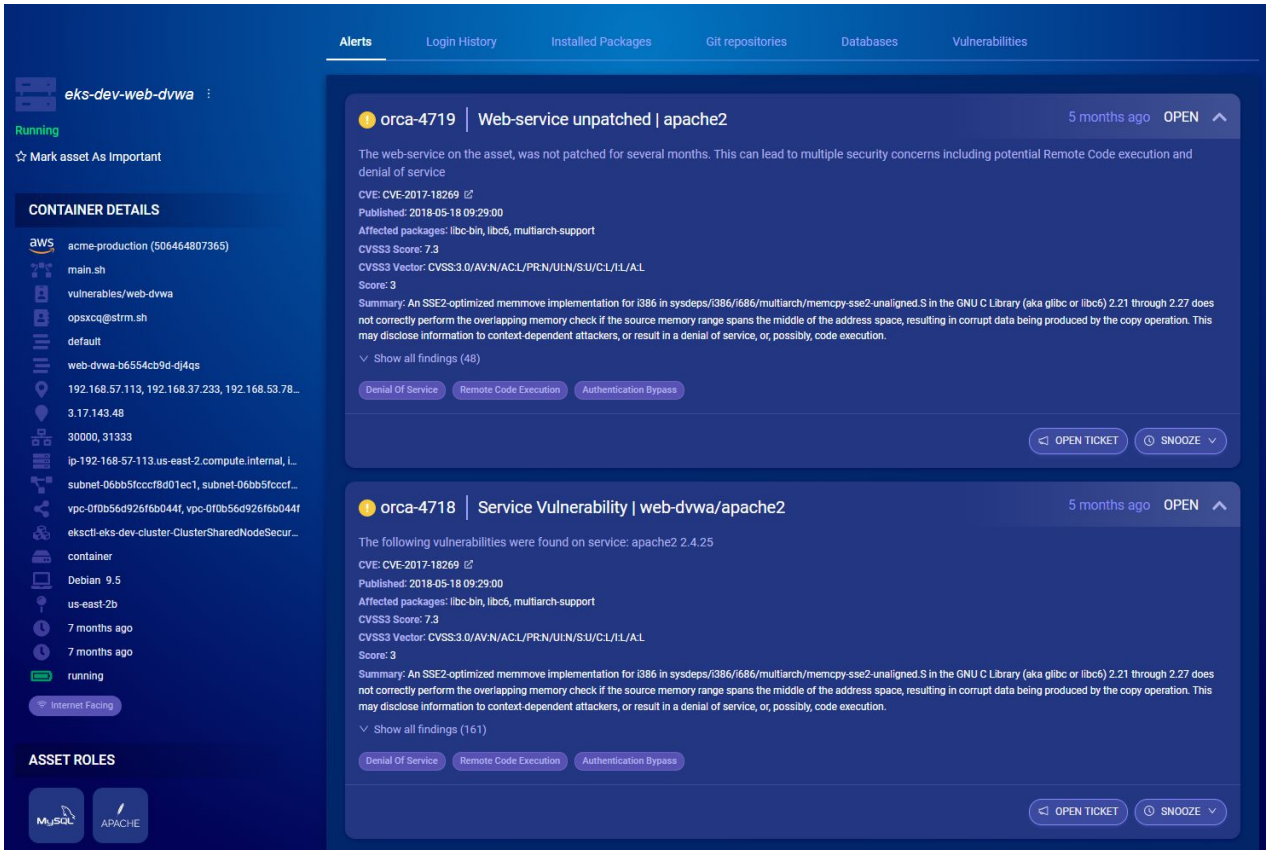
To be certain that such alerts don't constitute false positives, Orca performs statistical scans on the workload level. It's very likely for a random number to resemble a Social Security number, yet it's extremely unlikely for the majority of a file with thousands of numbers to contain one valid SSN by pure chance.

Container Scanning

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, and recursively runs the scanning processes mentioned above — vulnerabilities, configuration, malware, lateral movement risk, sensitive data, etc.

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

Ex. 2067-014



Example: Vulnerable container

There are several benefits to the approach of scanning containers via SideScanning vs competing approaches:

- Unlike CI/CD or registry scanning solutions, Orca scans **the runtime view** of the container, not the initial one. This means that it can detect deviations from baseline, including compromises of the environment.

- Orca is agnostic to the orchestration layer and covers all of the containers regardless of any integration to the orchestration layer (Kubernetes or other) that can be skipped.

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 aren't accessible in these types of deployments.

Collector Teardown

At this point, Orca tears down its collectors and securely sends the gathered information to its backend to begin its data analysis and reporting phase.

Ex. 2067-015

Combining Information, Analysis, and Reporting

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.

Showing Alerts in Context

Orca combines conclusions from different environmental perspectives 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.)

Consider a vulnerability in a web service. Orca scores it as:

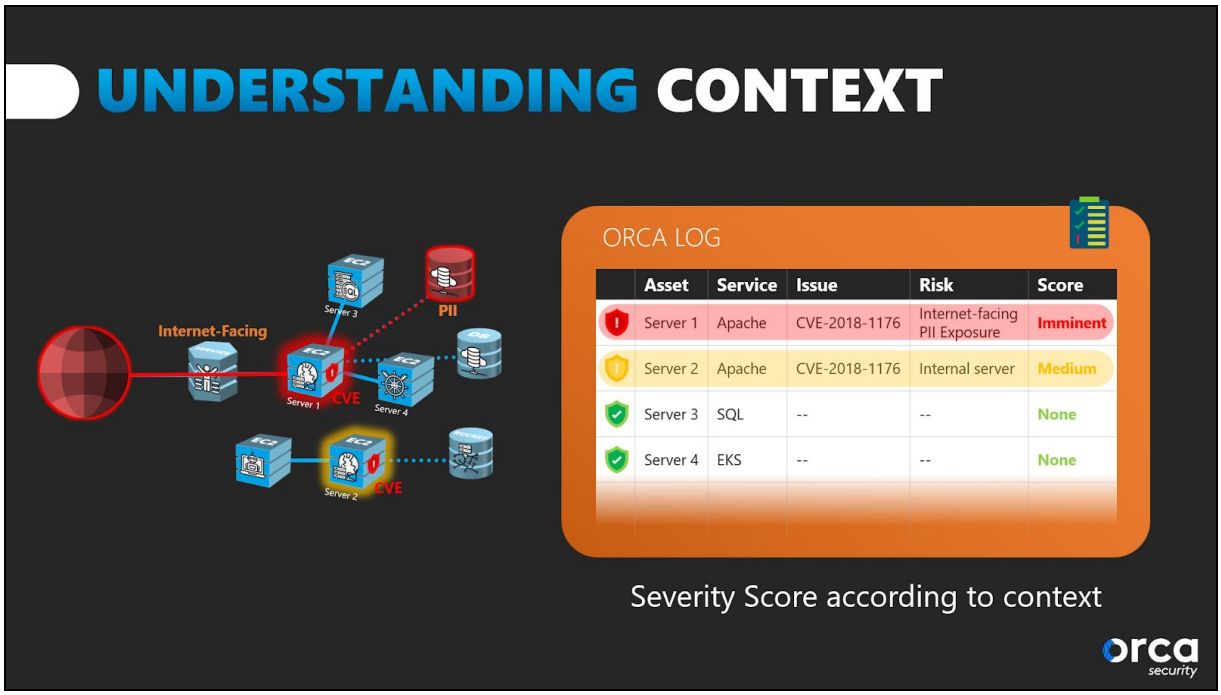
- High risk if it's connected to the internet, either directly or indirectly via a load balancer or reverse proxy
- Medium severity if it's only accessible internally
- Low severity if it's blocked by a security group configuration of the cloud provider

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.

Orca also evaluates network misconfigurations and their implications. 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.

One big advantage of Orca is that the "bird's eye view" of the control plane path and the "detailed view" of the data plane path are integrated and analyzed by a single vendor and all the data is in a single holistic database. Now, with Orca, all assets and their associated risks can be connected together and viewed in full context. Using Orca no further integration of disparate data sources is necessary.

Ex. 2067-016



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.

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.

Ex. 2067-017

Extending the map into containerized environments

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. 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.

Combining Low Severity Issues into Larger Alerts

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.

Ex. 2067-018

In Summary

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.

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.

About Orca Security

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.

Delivered as SaaS, Orca Security's patent-pending 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.

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.

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.

Connect your first cloud account in minutes and see for yourself. Visit https://orca.security

© Copyright Orca Security, 2020. All trademarks, service marks, and trade names referenced in this material are the property of their respective owners.

Ex. 2067-019