IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ORCA SECURITY LTD.,	)	
	)	
Plaintiff,	)	
	)	
v.	)	C.A. No. _____________
	)	
WIZ, INC.,	)	**DEMAND FOR JURY TRIAL**
	)	
Defendant.	)	

COMPLAINT FOR PATENT INFRINGEMENT

INTRODUCTION AND SUMMARY OF THE ACTION

1. Plaintiff Orca Security Ltd. ("Orca") brings this action against Wiz, Inc. ("Wiz") to put an end to Wiz's flagrant, ongoing, and unauthorized use of Orca's patented technologies.

2. Wiz has built its business on a simple business plan: copy Orca. This copying is replete throughout Wiz's business and has manifest in myriad ways. In its marketing, Wiz copies Orca's imagery, its message, and even the coffee it uses at trade shows. In prosecuting patents, Wiz recruited away Orca's former patent attorney to copy Orca's intellectual property and even the figures from Orca's patents. And, most importantly for this action, in its products and services, Wiz has embedded a number of revolutionary inventions developed and patented by Orca, passed those inventions off falsely as Wiz innovations, and forced Orca to compete against its own technological breakthroughs in the marketplace. Wiz's conduct in this regard is illegal, unjust, and in violation of the United States patent laws. Orca thus brings this complaint to redress Wiz's willful and deliberate infringement of Orca's patents.

* * *

3. Modern cloud computing launched in 2006, and quickly evolved from an emerging fad to the predominant technology employed across the globe. By 2018, nearly half of all companies claimed that 31% to 60% of their IT systems were cloud-based.[1]

4. With this widespread and rapid adoption came inevitable security threats that, if left unchecked, could threaten the industry. What made the cloud so attractive—the ability to quickly spin-up or tear-down assets on demand and expand at an unprecedented pace—also made cloud computing environments exceptionally challenging to protect.

5. Before Orca, stale security approaches and conventional wisdom from legacy technologies were employed. Those entrenched in the field adapted traditional security tools designed for on-premise physical computers to the cloud environment, either checking all traffic going in or going out (network security) or attempting to install agents within each virtual asset within the system (endpoint security). Those tools—effective for discrete numbers of physical machines or services—were woefully inadequate to protect cloud-computing environments with enormous and dynamically changing numbers of virtual assets. This led to multiplying vulnerabilities and tremendous uncertainty in that large organizations had little insight into which services operate in their environment, who owns those services, who is obligated to maintain them, and what risks attend them.

6. Enter Avi Shua, an Israeli-born cybersecurity technologist with a life-long fascination with ways to protect—or break into—computer systems. Even as a teen, Mr. Shua led corporate IT security for his high school. Mr. Shua then spent 10 years in the Israel Defense Forces as part of Unit 8200, an elite division of the Israel Intelligence Corps responsible for collecting signal intelligence and code decryption, counterintelligence, cyberwarfare, military intelligence,

[1] https://www.comptia.org/content/research/2018-trends-in-cloud-computing

and surveillance. Following his military service, Mr. Shua joined Check Point Software, an early pioneer in the computer security industry. Mr. Shua quickly rose through the ranks during his decade at Check Point, ultimately serving as its Chief Technologist for four years.

7. After leaving Check Point, Mr. Shua turned his sights toward addressing the many shortcomings he had observed in cloud computing security. Among other things, Mr. Shua realized that the transient nature of workloads in a virtual environment made it effectively impossible for traditional endpoint and network security to continuously map onto those workloads. The result was a whack-a-mole approach that looked to secure workloads by adjusting endpoint security dynamically as vulnerabilities arose. This approach resulted in long periods with no security visibility, gaping holes in protection, and prohibitive costs to implement.

8. Dissatisfied, Mr. Shua looked to develop a new platform that could provide frictionless and comprehensive security coverage to a constantly evolving cloud environment. He realized that there was a better way—a more effective choke point—for analyzing cloud security within a virtual environment: the virtualization itself held the answer. In general terms, Mr. Shua conceived of a revolutionary approach that analyzed virtual cloud assets using read-only access with no impact on performance, and without deploying agents or network scanners. The result was vastly improved visibility into a cloud environment, deeper and better results, and improved speed. Mr. Shua's innovations also enabled the integration of data into unified data models, to view cloud security threats in a context that was not possible before, and so to prioritize risks that endanger the organization's most critical assets.

9. Mr. Shua and his co-founders founded Orca in 2019 to create a cloud security tool that brought Mr. Shua's inventions to market. The company took off like a rocket ship: the year after it was founded, Orca Security achieved more than 1,000% year-over-year growth. As noted

by customers, this success was due to the genius of Orca's Platform. As one customer noted, "Orca Security is unique in that it locates vulnerabilities with precision and delivers tangible, actionable results—without having to sift through all of the noise."[2] And another customer echoed the sentiment, stating: "Orca is unique in that it doesn't require the installation of cumbersome agents. This reduces integration costs, and eliminates the question we had always asked ourselves, 'are agents installed on all resources?'"[3]

10. In the four years since its founding, Orca has raised substantial investment funds and grown from fewer than a dozen to more than 400 employees today. Orca has been recognized as one of the most innovative companies in cloud security and, in 2022, was the recipient of Amazon Web Services Global Security Partner of the Year Award.[4] The U.S. Patent Office has awarded Orca several patents for Mr. Shua's inventions, including U.S. Patent Nos. 11,663,031 (the "'031 patent"), and 11,663,032 (the "'032 patent"), among others.

11. Now, Orca is threatened because the Defendant, Wiz, Inc., has taken Orca's revolutionary inventions and created a copycat cloud security platform, improperly trading off of Orca's inventions, including those claimed in the '031 and '032 patents, without authorization.

WIZ AND ITS WIDESPREAD COPYING OF ORCA

12. Wiz was founded in January 2020 by Assaf Rappaport, Ami Luttwak, Yinon Costica, and Roy Reznikthat, a team that previously led the Cloud Security Group at Microsoft,

[2] https://web.archive.org/web/20200930194127/https://orca.security/ (Aaron Brown, Senior Cloud Security Engineer, Sisense).

[3] https://web.archive.org/web/20200930194127/https://orca.security/ (Jonathan Jaffe, Head of Information Security, Legal Counsel, people.ai).

[4] https://finance.yahoo.com/news/orca-security-awarded-2022-regional-010000110.html

one of the top providers of cloud computing environments in the world.[5] According to those founders, it was their time at Microsoft that provided them the "insight" that current cloud security tools were too complicated, fragmented, and generate too many alerts.[6] Wiz was thus founded to "build a platform that lets teams scan their environments across compute types and cloud services for vulnerabilities and configuration, network, and identity issues without agents"; *i.e.,* to do exactly what Orca had already been doing for over a year.[7]

13. This was not a coincidence or a simultaneous stroke of genius. On the contrary, Wiz was birthed from the very beginning as a counterfeit copy of *Orca's* ideas—Mr. Shua had presented Orca's Platform to Wiz's founders at Microsoft in May 2019, and the so-called "insight" of which Wiz boasts was nothing more than the misappropriation of Mr. Shua's ideas and Orca's technology as presented to Wiz's founders before they formed Wiz and sought to launch a copycat competitor to Orca. It was at this 2019 meeting that Mr. Shua explained how cloud security would forever be changed by his novel agentless cloud security platform as implemented in Orca's cloud-native security platform. Within months, the Wiz founders left their lucrative careers at Microsoft to start Wiz, build a clone of Orca's technology, and compete directly with Orca.

14. Because of the massive head start it received from Orca and Mr. Shua, it took Wiz just months from the time the company was founded before it had a fully functioning "cloud visibility solution for enterprises that provides a complete view of security risks across clouds,

[5] https://www.darkreading.com/cloud/former-microsoft-cloud-security-leads-unveil-new-startup; https://www.forbes.com/sites/davidjeans/2020/12/09/wiz-sequoia-index-cybersecurity-100-million-former-microsoft-executives/?sh=4414df63254c ("At Microsoft, Rappaport says he became increasingly aware of a growing problem for large companies: managing cloud security threats was a fragmented process, with security teams becoming overwhelmed by alerts.").

[6] https://www.darkreading.com/cloud/former-microsoft-cloud-security-leads-unveil-new-startup

[7] *Id.*

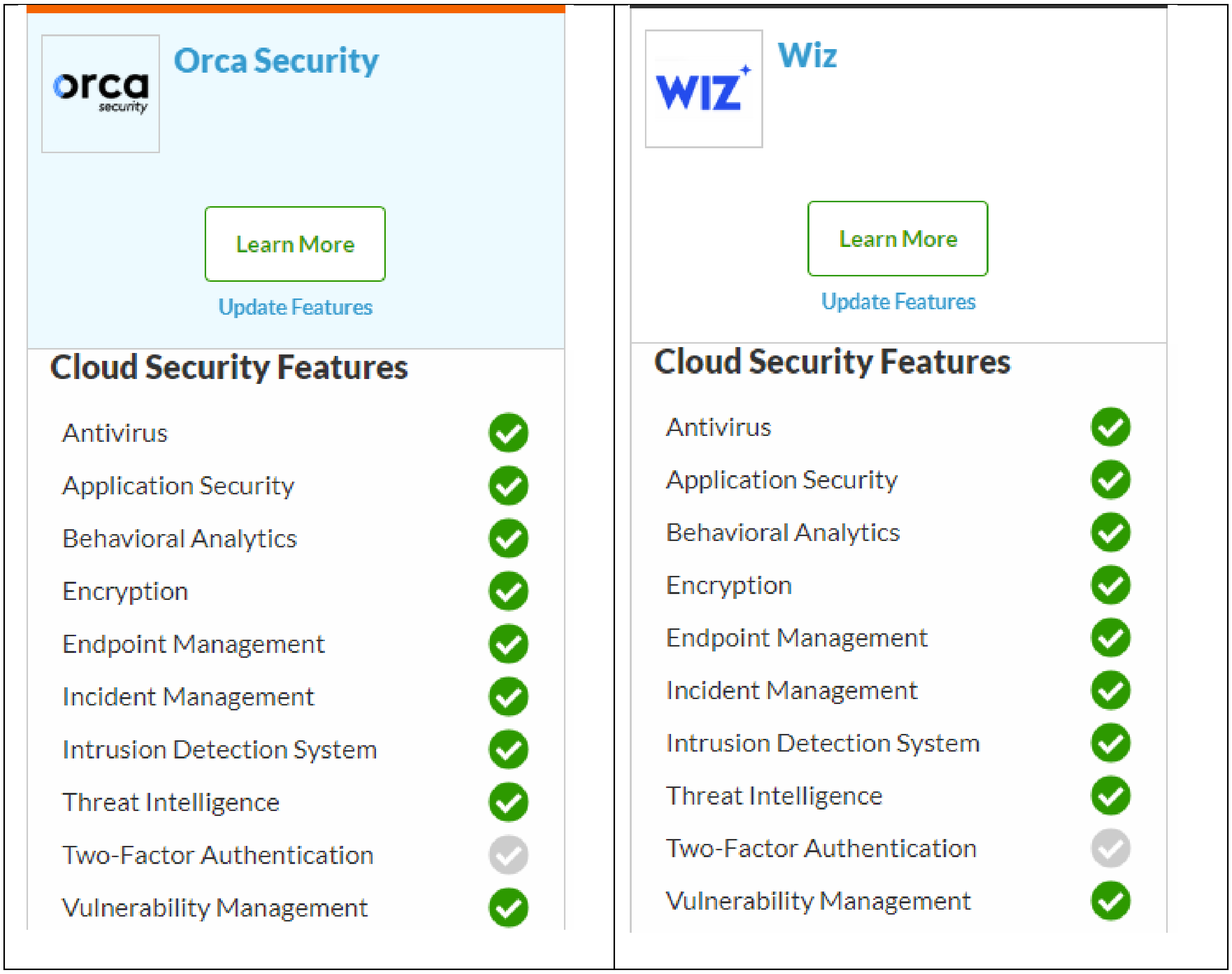
workloads and containers" that was "already used by Fortune 100 companies."[8] In August 2022, Wiz announced it had become the "fastest-growing software company ever" reaching "$100M ARR [annual recurring revenue] in 18 months."[9] And just eight months later in February 2023, Wiz raised $300 million and achieved a company valuation of $10 billion.[10]

15. Wiz's wholesale copying of Orca's technology has been observed by third party industry analysts. For example, SOURCEFORGE's comparison of Orca and Wiz lists identical "Cloud Security Features" for each platform:

[8] https://www.securityweek.com/cloud-security-firm-wiz-emerges-stealth-100m-funding/

[9] https://www.wiz.io/blog/100m-arr-in-18-months-wiz-becomes-the-fastest-growing-software-company-ever

[10] https://techcrunch.com/2023/02/27/cloud-security-startup-wiz-now-valued-at-10b-raises-300m/



Orca Security

Learn More

Update Features

Cloud Security Features

Antivirus	✓
Application Security	✓
Behavioral Analytics	✓
Encryption	✓
Endpoint Management	✓
Incident Management	✓
Intrusion Detection System	✓
Threat Intelligence	✓
Two-Factor Authentication	✓
Vulnerability Management	✓

Wiz

Learn More

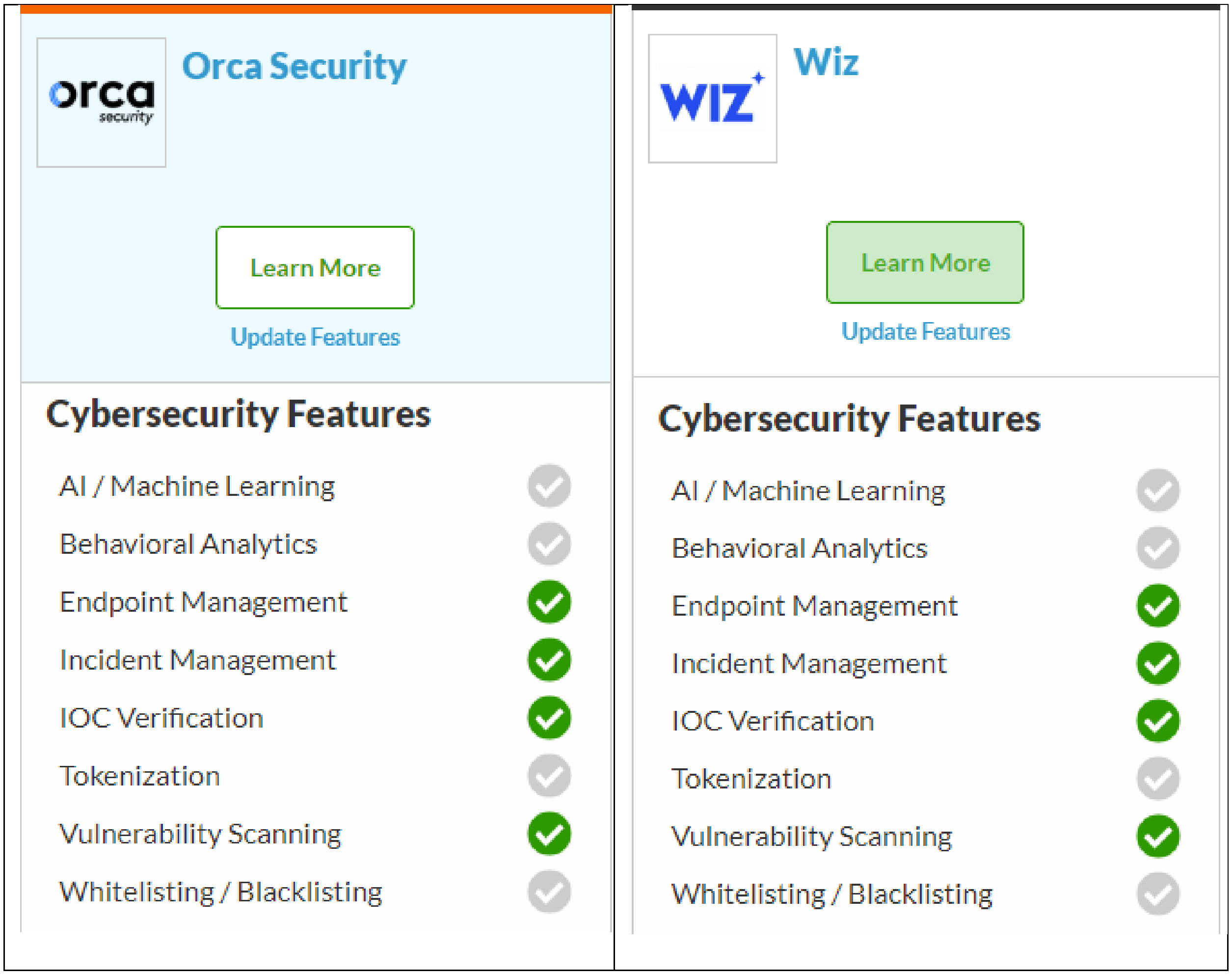
Update Features

Cloud Security Features

Antivirus	✓
Application Security	✓
Behavioral Analytics	✓
Encryption	✓
Endpoint Management	✓
Incident Management	✓
Intrusion Detection System	✓
Threat Intelligence	✓
Two-Factor Authentication	✓
Vulnerability Management	✓

https://sourceforge.net/software/compare/Orca-Security-vs-Wiz/.

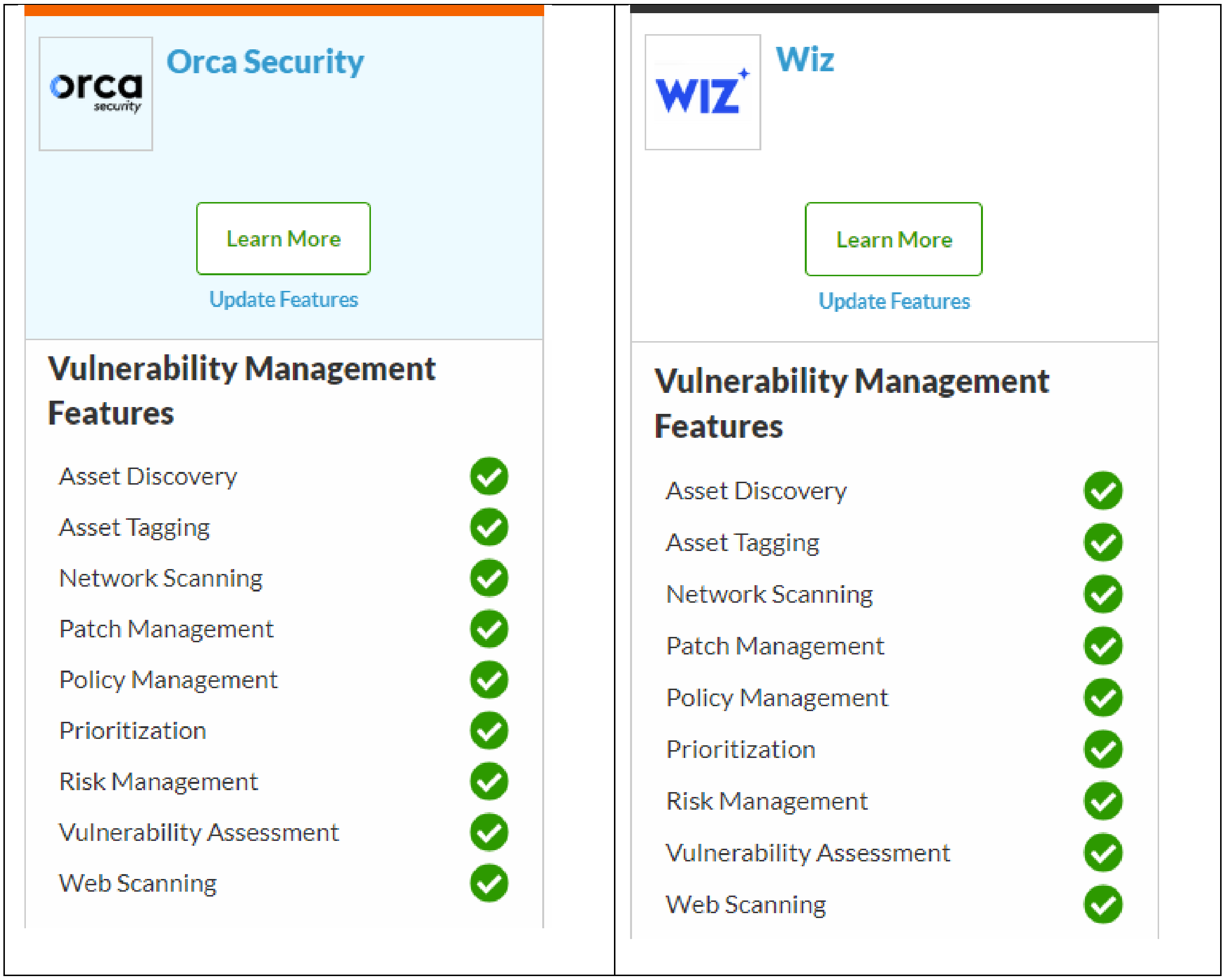
16. SOURCEFORGE also notes that Wiz has the same "Cybersecurity Features" as Orca:



Orca Security		Wiz	
Cybersecurity Features		**Cybersecurity Features**	
AI / Machine Learning	✓	AI / Machine Learning	✓
Behavioral Analytics	✓	Behavioral Analytics	✓
Endpoint Management	✅	Endpoint Management	✅
Incident Management	✅	Incident Management	✅
IOC Verification	✅	IOC Verification	✅
Tokenization	✓	Tokenization	✓
Vulnerability Scanning	✅	Vulnerability Scanning	✅
Whitelisting / Blacklisting	✓	Whitelisting / Blacklisting	✓

Id.

17. SOURCEFORGE further shows that Wiz has the same "Vulnerability Management Features" as Orca:



Vulnerability Management Features	Orca Security	Wiz
Asset Discovery	✓	✓
Asset Tagging	✓	✓
Network Scanning	✓	✓
Patch Management	✓	✓
Policy Management	✓	✓
Prioritization	✓	✓
Risk Management	✓	✓
Vulnerability Assessment	✓	✓
Web Scanning	✓	✓

Id.

18. Through all of its copying, Wiz has attributed none of its technology to Orca. In fact, Wiz has done the opposite. Wiz has claimed it was the "first cloud visibility solution"[11] and the "first full stack multi-cloud security platform."[12] But even its "full stack" descriptor was copied from Orca. It was Orca that first announced its "Unprecedented Full Stack Cloud Visibility" platform in June 2019, months before Wiz was even founded.[13] As another more recent example, Wiz announced in June 2022 that it had a "new vision for cloud security" with the "introduction

[11] https://web.archive.org/web/20210128014251/https://wiz.io/

[12] https://web.archive.org/web/20210422201202/https://www.wiz.io/product

[13] https://orca.security/resources/blog/orca-security-lands-6-5m-seed-round-to-deliver-it-security-teams-unprecedented-full-stack-cloud-visibility-securing-high-velocity-cloud-growth/

of attack path analysis."[14] But Wiz's "attack path analysis" was not new, and it wasn't Wiz's vision. It was Mr. Shua's from just two months earlier. On March 31, 2022, Mr. Shua blogged about Orca's new "Cloud Attack Path Analysis" dashboard, which Wiz copied.[15]

19. Wiz's copying of Orca did not stop with the technology, but pervades Wiz's business as a whole. For example, Orca realized early on that its cloud-native approach could be analogized to a medical MRI, providing a full model of the cloud environment without affecting it in any way. Early Orca marketing materials noted: "***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***." Exhibit 3 (Orca SideScanning Technical Brief (2020)) at 15. Wiz copied this message: "Instead of using an intrusive agent, Wiz leverages cloud-native tools to perform scans without interrupting or impacting production workloads. ***Just like an MRI performs a 3D scan of the body without affecting the body itself***, snapshot scanning achieves deep analysis of the workload without any impact or interruption to the live workload." Exhibit 4 (Wiz "Agentless Scanning" (Jan. 19, 2022)).

20. As another example, Orca promoted its technology as assuming the "heavy lifting" of contextualizing detected security threats and prioritizing those that matter most. Exhibit 3 at 15 ("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

[14] https://www.wiz.io/blog/uniting-builders-and-defenders-a-new-vision-for-cloud-security

[15] https://orca.security/resources/blog/cloud-attack-path-analysis/

intelligence possible."). Wiz copied this too beginning with its very first website in 2020: "***We do the heavy lifting*****,** you get total visibility."[16]

21. Wiz even copied the more mundane aspects of Orca's marketing. For example, at a multi-day security conference in London, Orca decided that it would break away from typical technology booths and instead sponsor a coffee booth. Wiz attended the same conference. On the first day, Wiz sponsored a typical technology booth. The following day, Wiz showed up with its own coffee machine. Just like Orca.

22. Wiz also has knowingly copied Orca's patents, its prosecution strategy, and even its prosecuting attorney. Orca's first patent applications were filed and prosecuted by a lawyer at a small boutique firm with less than 10 attorneys, with whom Mr. Shua worked directly and confidentially. That engagement was terminated in 2021 when Orca learned that Wiz had engaged the same lawyer to file patents for Wiz on overlapping technology. Wiz's patent applications now include figures and descriptions that are nearly identical to those found in Orca's '031 and '032 patents:

[16] https://web.archive.org/web/20201209145922/http://www.wiz.io/.

<table>
<tr><th>Orca</th><th>Wiz</th></tr>
<tr>
<td>

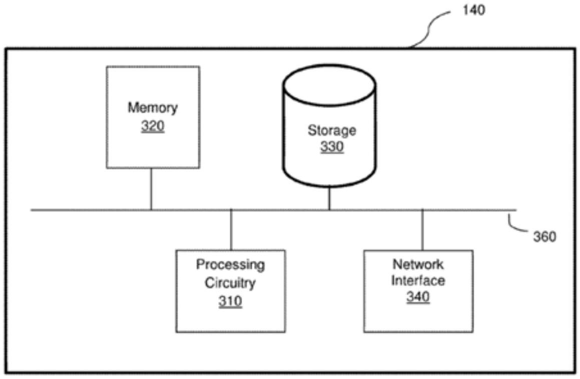


FIG. 3

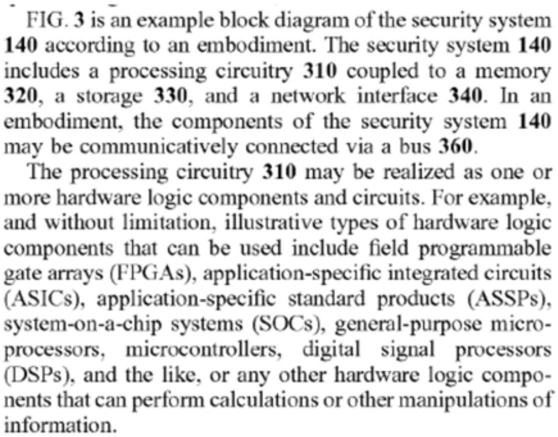


FIG. 3 is an example block diagram of the security system **140** according to an embodiment. The security system **140** includes a processing circuitry **310** coupled to a memory **320**, a storage **330**, and a network interface **340**. In an embodiment, the components of the security system **140** may be communicatively connected via a bus **360**.

The processing circuitry **310** may be realized as one or more hardware logic components and circuits. For example, and without limitation, illustrative types of hardware logic components that can be used include field programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), application-specific standard products (ASSPs), system-on-a-chip systems (SOCs), general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), and the like, or any other hardware logic components that can perform calculations or other manipulations of information.

’032 patent at Fig 3, 8:7-23; ’031 patent at Fig. 3, 9:15-31.

</td>
<td>

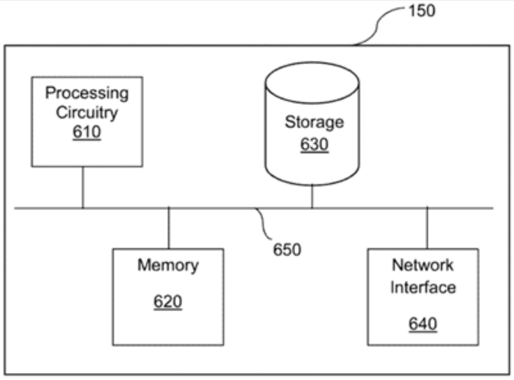


FIG. 6

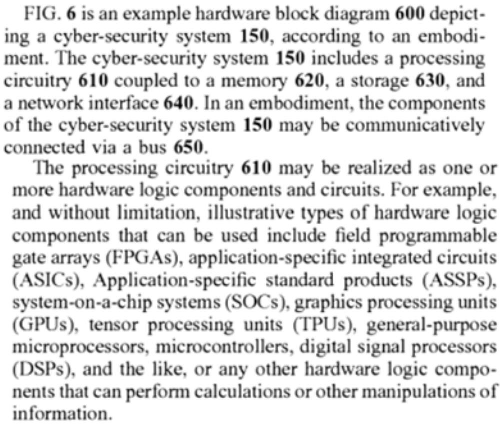


FIG. **6** is an example hardware block diagram **600** depicting a cyber-security system **150**, according to an embodiment. The cyber-security system **150** includes a processing circuitry **610** coupled to a memory **620**, a storage **630**, and a network interface **640**. In an embodiment, the components of the cyber-security system **150** may be communicatively connected via a bus **650**.

The processing circuitry **610** may be realized as one or more hardware logic components and circuits. For example, and without limitation, illustrative types of hardware logic components that can be used include field programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), Application-specific standard products (ASSPs), system-on-a-chip systems (SOCs), graphics processing units (GPUs), tensor processing units (TPUs), general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), and the like, or any other hardware logic components that can perform calculations or other manipulations of information.

Wiz’s U.S. Patent No. 11,374,982 at Fig. 6, 20:61-21:12.

</td>
</tr>
</table>

23. Again, this was no coincidence. On information and belief, Wiz knew that the lawyer it hired had prosecuted Orca’s patent applications and hired him to assist Wiz in its attempts to pass off Orca’s technology and intellectual property.

24. In furtherance of its scheme to copy Orca, Wiz also recruited Orca’s outside corporate counsel to work for Wiz. That lawyer attended Orca’s Board of Director meetings and, as a result, was exposed to Orca’s highly confidential technology and business plans. Orca replaced its outside corporate counsel in November 2020 after it learned that Wiz had engaged the very same lawyer as its own corporate counsel. On information and belief, Wiz knew that the

lawyer it hired was Orca's outside corporate counsel and Wiz hired him to assist Wiz in its attempts to copy Orca.

25. Beyond the foregoing examples, on information and belief, Wiz has hired former Orca employees and worked with third parties to acquire Orca's confidential information relating to current and future product plans, marketing, sales, prospective customers, and prospective employees, and has used that confidential information in furtherance of its efforts to copy and to compete unfairly with Orca.

26. This action seeks to put an end to, and obtain relief for, this pattern of copying and Wiz's willful infringement of the '031 patent and the '032 patent (collectively, the "Asserted Patents").

THE PARTIES

27. Plaintiff Orca Security Ltd. is an Israeli company with a principal place of business at 3 Tushia St., Tel Aviv, Israel 6721803.

28. On information and belief, Defendant Wiz, Inc. is a Delaware company with a principal place of business at One Manhattan West, 57th Floor, New York, New York.[17]

JURISDICTION AND VENUE

29. This action arises under the patent laws of the United States, 35 U.S.C. § 1 et seq. This Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

30. This Court has personal jurisdiction over Wiz because Wiz is subject to general and specific jurisdiction in the state of Delaware. Wiz is subject to personal jurisdiction at least because Wiz is a Delaware corporation and resides in this District. Wiz has made certain minimum

[17] https://www.wiz.io/contact (Locations)

contacts with Delaware such that the maintenance of this suit does not offend traditional notions of fair play and substantial justice.

31. The exercise of personal jurisdiction comports with Wiz's right to due process because, as described above, Wiz has purposefully availed itself of the privilege of Delaware corporate laws such that it should reasonably anticipate being haled into court here.

32. Venue is proper in this district pursuant to 28 U.S.C. §§ 1391 and 1400(b) at least because Wiz is incorporated in the State of Delaware and is subject to personal jurisdiction in this District.

COUNT I
(Infringement of the '031 Patent)

33. Orca incorporates all other allegations in this Complaint.

34. The '031 patent is entitled "Techniques for Securing Virtual Cloud Assets at Rest Against Cyber Threats" and was duly and legally issued on May 30, 2023. A true and correct copy of the '031 patent is attached hereto as Exhibit 1.

35. Orca is the owner of all rights, title, and interest in the '031 patent.

36. The '031 patent is valid and enforceable.

37. The inventions claimed in the '031 patent improved on prior art cloud security systems and methods by, *inter alia*, taking at least one snapshot or requesting taking of at least one snapshot of a virtual machine at rest, and analyzing the at least one snapshot to detect vulnerabilities. *See, e.g.,* '031 patent at cls. 1-16. This snapshot-based analysis for inactive assets was not well understood, routine, or conventional. It is an inventive concept that allows virtual assets in a cloud computing platform to be analyzed and scanned for embedded vulnerabilities, at a time when the machine is inactive, because, among other things, the analysis does not require any interaction and/or information from a running virtual asset like agent-based solutions. By

analyzing virtual cloud assets at rest, the ’031 patent provides greater context for detected vulnerabilities and more comprehensive security for a cloud computing platform, including protecting against assets that may have become unsafe after they were turned off due to newly disclosed vulnerabilities or infrastructure changes.

(a) Direct Infringement of the ’031 Patent

38. Wiz, without authorization, directly infringes one or more claims of the ’031 patent, literally and/or under the doctrine of equivalents. Wiz infringes under 35 U.S.C. § 271 including, without limitation, 35 U.S.C. § 271(a), by making, using, selling, offering to sell, and/or importing within the United States without authority, Wiz’s CSP and/or other similar products or services, which include (or are otherwise referred to) but are not limited to Wiz’s Cloud Native Application Protection Platform (“CNAPP”), Cloud Security Posture Management (“CSPM”), Cloud Infrastructure Entitlement Management (“CIEM”), Data Security Posture Management (“DSPM”), Infrastructure-as-code (“IaC”) scanning (https://www.wiz.io/solutions/iac), and Cloud Detection and Response (“CDR”) platforms and/or features. *See* https://www.wiz.io/ (listing CNAPP, CSPM, CIEM, DSPM, IaC scanning, and CDR as “Product[s]”); *see also* https://www.wiz.io/product (same). Wiz’s infringement includes infringement of, for example, claim 9 of the ’031 patent.

39. Claim 9 of the ’031 patent recites:

> 1. A computer-implemented method for inspecting data, the method comprising:
>
> establishing an interface between a client environment and security components;
>
> using the interface to utilize cloud computing platform APIs to identify virtual disks of a virtual machine in the client environment;

using the computing platform APIs to query a location of at least one of the identified virtual disks;

receiving an identification of the location of the virtual disks of the virtual machine;

emulating the virtual disks for the virtual machine;

performing at least one of: (i) taking at least one snapshot, and (ii) requesting taking at least one snapshot of the virtual machine at rest, wherein the at least one snapshot represents a copy of the virtual disks of the virtual machine at a point in time;

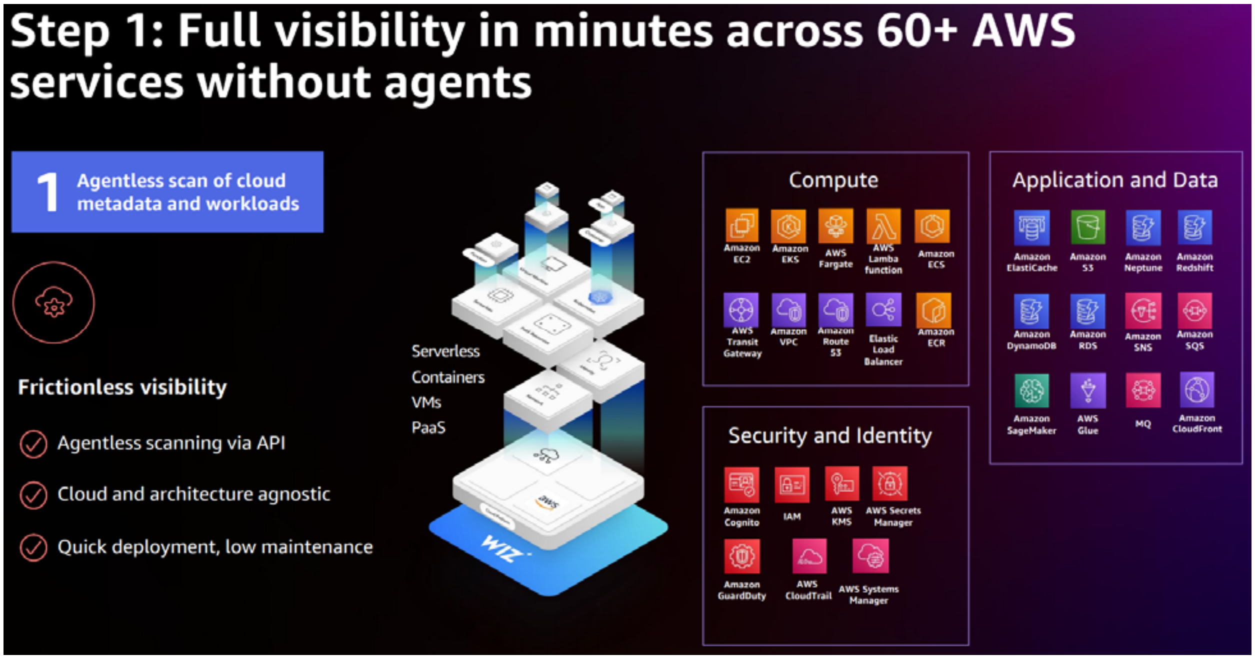
analyzing the at least one snapshot to detect vulnerabilities, wherein during the detection of the vulnerabilities by analyzing the at least one snapshot, the virtual machine is inactive; and

reporting the detected vulnerabilities as alerts.

40. On information and belief, Wiz practices each and every limitation of claim 9 of the ’031 patent by and through the use of Wiz’s CSP and/or other similar products or services for Wiz’s clients or customers.

41. The preamble of claim 9 recites “[a] computer-implemented method for inspecting data, the method comprising. . . .” To the extent the preamble is limiting, Wiz practices this step by, for example, using its computer-implemented CSP to inspect data in clients’ cloud computing environments, including inactive assets. *See, e.g.*, https://www.wiz.io/solutions/cnapp (“Wiz leverages unique technology to scan PaaS resources, Virtual Machines, Containers, Serverless Functions, . . . to identify the risks in each layer”); https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz (“Detect and prioritize CISA Known Exploited Vulnerabilities in the cloud with Wiz”).

42. Claim 9 further recites "establishing an interface between a client environment and security components . . . ." Wiz's public presentations and technical documentation confirm that Wiz practices this step by, for example, using Wiz's CSP to perform "[a]gentless scanning via API" provided by AWS, GCP, and Azure, among other cloud computing environments.



Step 1: Full visibility in minutes across 60+ AWS services without agents

1 Agentless scan of cloud metadata and workloads

Frictionless visibility

- Agentless scanning via API
- Cloud and architecture agnostic
- Quick deployment, low maintenance

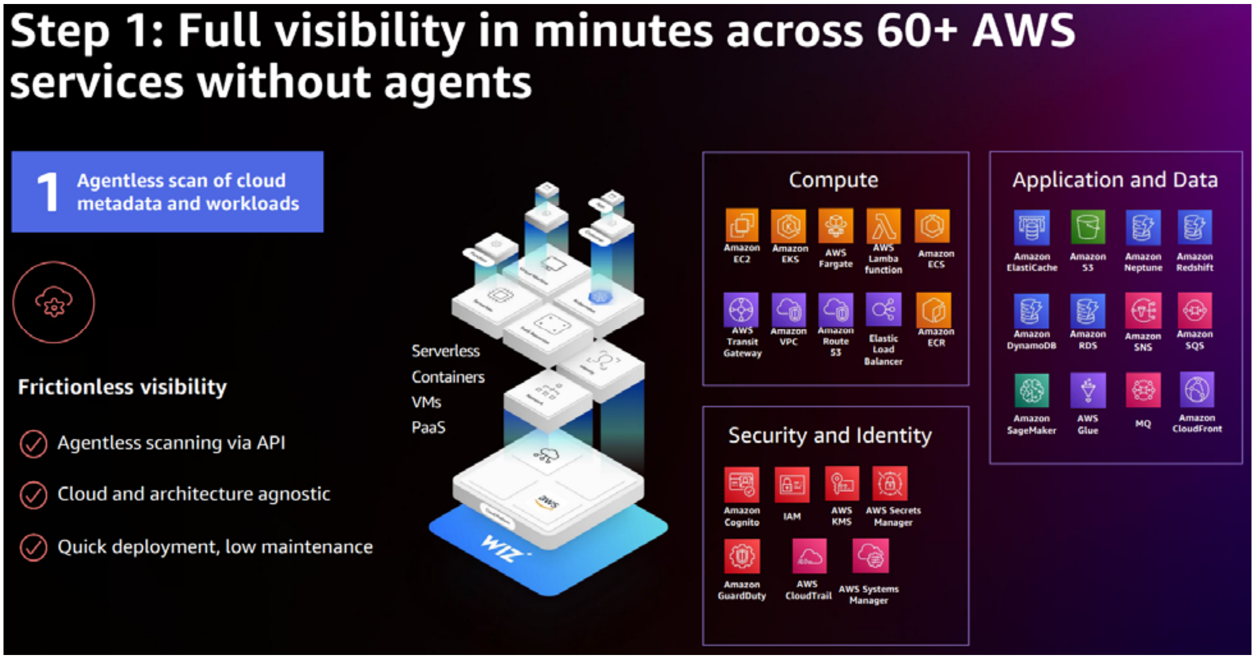
Compute

Security and Identity

Application and Data

See Exhibit 5 (AWS re:Invent - Context is Everything: Join the CNAPP Revolution to Secure Your AWS Deployments) at 13; Exhibit 6 (Wiz Cloud Security Platform Datasheet) (supported cloud computing platforms include AWS, Azure, and Google Cloud Platform (GCP)); https://support.wiz.io/hc/en-us/articles/5641497256860-Azure-Connector-Basics ("Wiz connects to your cloud environment via your cloud service provider's APIs in order to extract metadata and perform snapshot scans."); https://support.wiz.io/hc/en-us/articles/5449816387100-AWS-Connector-Basics (same); https://support.wiz.io/hc/en-us/articles/5642208092572-GCP-Connector-Basics (same); https://www.wiz.io/solutions/vulnerability-management ("Using a one-time cloud native API deployment, continuously assess workloads without deploying agents").

43. Claim 9 further recites "using the interface to utilize cloud computing platform APIs to identify virtual disks of a virtual machine in the client environment . . . ." Wiz practices this step by, for example, using Wiz's CSP to provide "[f]ull visibility" of virtual cloud assets in a client environment using an API provided by AWS, GCP, and Azure, among other cloud computing environments.



Step 1: Full visibility in minutes across 60+ AWS services without agents

1 Agentless scan of cloud metadata and workloads

Frictionless visibility

- Agentless scanning via API
- Cloud and architecture agnostic
- Quick deployment, low maintenance

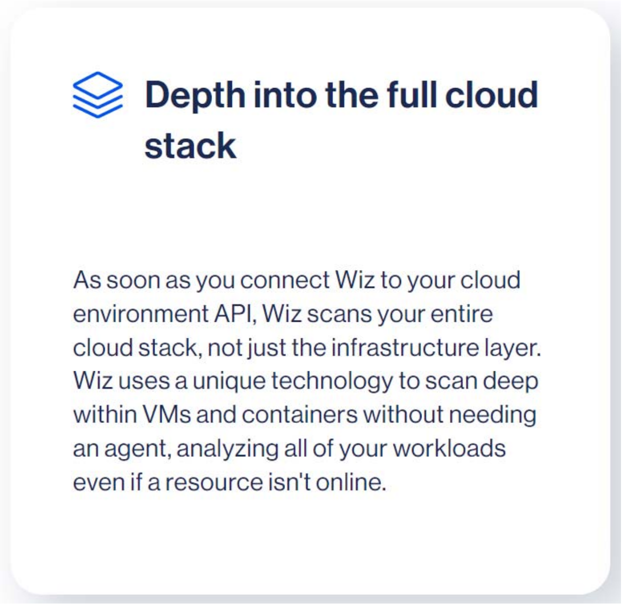
See Exhibit 5 at 13; Exhibit 6 (supported cloud computing platforms include AWS, Azure, and Google Cloud Platform (GCP)). Through the API, Wiz creates a graph of a client environment "with full context on the resource[s]," which includes identifying virtual disks of virtual machines. *See* https://www.wiz.io/blog/uniting-builders-and-defenders-a-new-vision-for-cloud-security; Exhibit 6 at 3 ("Wiz uses the full context of your cloud and combines this information into a single graph in order to correlate related issues"), 4 (Wiz "takes a snapshot of each VM system volume and analyzes its operating system, application layer, and data layer statically with no performance impact.").

44. Claim 9 further recites "using the computing platform APIs to query a location of at least one of the identified virtual disks . . . ." Wiz performs this step by, for example, using computing platform APIs to perform a query to locate virtual disks and other resources. *See* Exhibit 5 at 13 ("Agentless scanning via API"); https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz ("You can query and locate all the VMs, containers, and serverless functions in your cloud environment that are vulnerable to a specific CVE in the catalog with a simple query shortcut."); https://www.wiz.io/solutions/cnapp ("Scan buckets, data volumes, and databases and quickly classify the data to track wh[ere] data is located."); https://support.wiz.io/hc/en-us/articles/5643759466396-Security-Graph-Basics ("[C]heck out our guide for optimizing your Security Graph queries.").

45. Claim 9 further recites "receiving an identification of the location of the virtual disks of the virtual machine . . . ." Wiz practices this step by, for example, identifying virtual disks and other resources it locates when it performs a query. *See* https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz ("You can query and locate all the VMs, containers, and serverless functions in your cloud environment that are vulnerable to a specific CVE in the catalog with a simple query shortcut."). As another example, Wiz uses Wiz's CSP to create a graph showing the locations of virtual cloud assets, including virtual machines and virtual disks, within a client environment. *See* Exhibit 6 at 3 (Wiz "uses the full context of your cloud and combines this information into a single graph in order to correlate related issues"); *see also* Exhibit 5 at 13 ("Full visibility in minutes . . . without agents").

46. Claim 9 further recites "emulating the virtual disks for the virtual machine . . . ." On information and belief, Wiz practices this step by, for example, using Wiz's CSP to scan "all

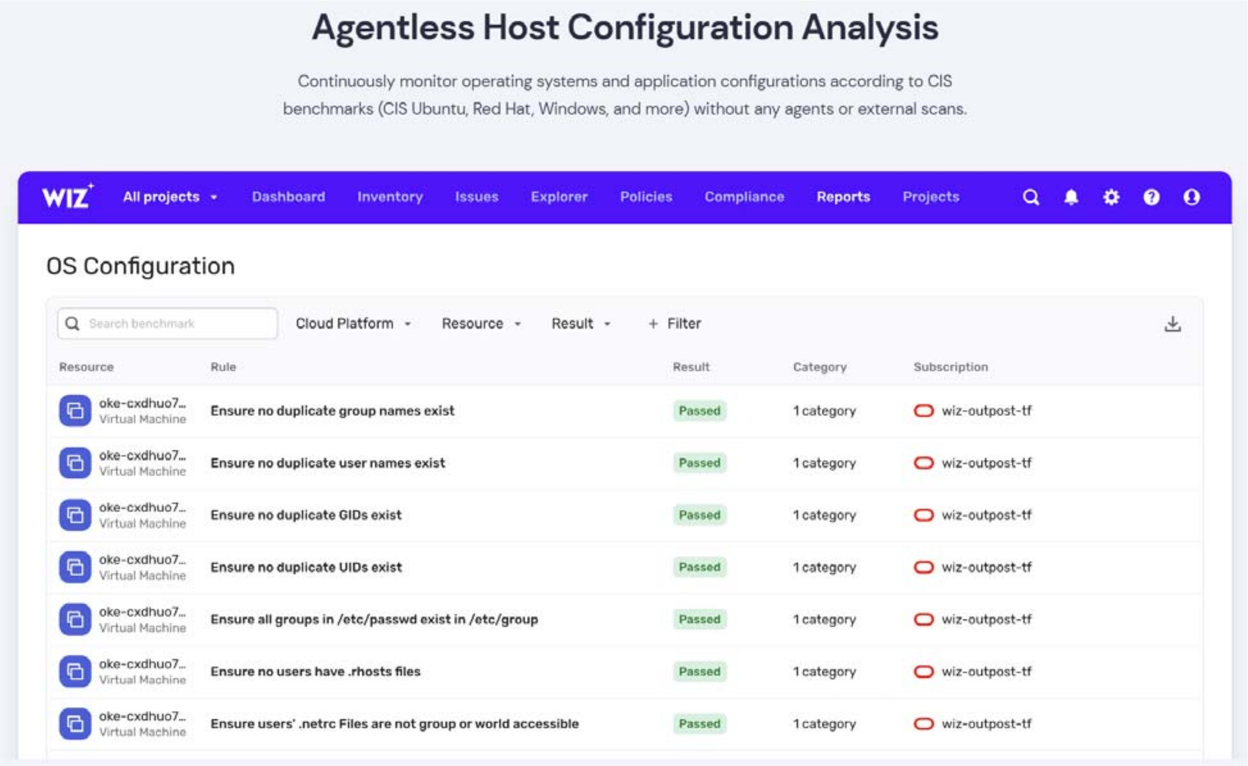
of [a customer's] workloads even if a resource isn't online" because an offline resource's virtual disks will need to be emulated before scanning.



Depth into the full cloud stack

As soon as you connect Wiz to your cloud environment API, Wiz scans your entire cloud stack, not just the infrastructure layer. Wiz uses a unique technology to scan deep within VMs and containers without needing an agent, analyzing all of your workloads even if a resource isn't online.

https://legacy.wiz.io/partners/google. Wiz's website also promotes its platform as using agentless "snapshot" scanning. *See* https://www.wiz.io/solutions/cnapp ("Wiz deployment leverages a single cloud role to scan your entire cloud environment: PaaS, Virtual Machines, Containers, Serverless functions, Buckets, Data volumes and Databases."); https://www.wiz.io/solutions/vulnerability-management. As Wiz's blog posts explain, "volume snapshot approach" where snapshots are scanned "out of band, do not rely on the cloud environment's compute resources to run." https://www.wiz.io/blog/agents-are-not-enough-why-cloud-security-needs-agentless-deep-scanning. Accordingly, on information and belief, Wiz uses its own separate compute resources to emulate virtual disks that it analyzes.



Agentless Host Configuration Analysis

Continuously monitor operating systems and application configurations according to CIS benchmarks (CIS Ubuntu, Red Hat, Windows, and more) without any agents or external scans.

OS Configuration

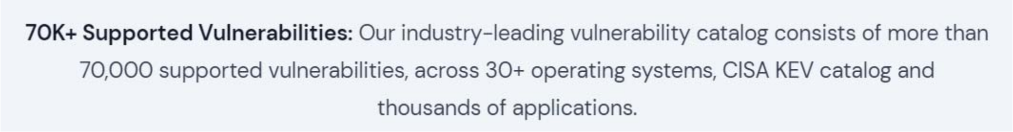
Resource	Rule	Result	Category	Subscription
oke-cxdhuo7... Virtual Machine	Ensure no duplicate group names exist	Passed	1 category	wiz-outpost-tf
oke-cxdhuo7... Virtual Machine	Ensure no duplicate user names exist	Passed	1 category	wiz-outpost-tf
oke-cxdhuo7... Virtual Machine	Ensure no duplicate GIDs exist	Passed	1 category	wiz-outpost-tf
oke-cxdhuo7... Virtual Machine	Ensure no duplicate UIDs exist	Passed	1 category	wiz-outpost-tf
oke-cxdhuo7... Virtual Machine	Ensure all groups in /etc/passwd exist in /etc/group	Passed	1 category	wiz-outpost-tf
oke-cxdhuo7... Virtual Machine	Ensure no users have .rhosts files	Passed	1 category	wiz-outpost-tf
oke-cxdhuo7... Virtual Machine	Ensure users' .netrc Files are not group or world accessible	Passed	1 category	wiz-outpost-tf

https://www.wiz.io/solutions/vulnerability-management.

47. Claim 9 further recites "performing at least one of: (i) taking at least one snapshot, and (ii) requesting taking at least one snapshot of the virtual machine at rest, wherein the at least one snapshot represents a copy of the virtual disks of the virtual machine at a point in time . . . ." Wiz performs this step by, for example, taking a snapshot of a virtual disk in order to "analyze[] [the] operating system, application layer, and data layer" of virtual machines in a client environment. *See* Exhibit 6 at 4, 3 (Wiz "[s]cans the workloads inside the container to determine . . . its vulnerabilities")*; see also* Exhibit 5 at 27. Wiz's technical documentation explains that "Wiz connects to [a] cloud environment via [a] cloud service provider's APIs in order to extract metadata and perform snapshot scans." https://support.wiz.io/hc/en-us/articles/5641497256860-Azure-Connector-Basics; https://support.wiz.io/hc/en-us/articles/5449816387100-AWS-Connector-Basics (same); https://support.wiz.io/hc/en-us/articles/5642208092572-GCP-Connector-Basics (same). On information and belief, Wiz also requests taking a snapshot of

virtual disks on a virtual machine when it is offline. https://legacy.wiz.io/partners/google ("Wiz uses a unique technology to scan deep within VMs and containers without needing an agent, analyzing all of your workloads even if a resource isn't online.").

48. Claim 9 further recites "analyzing the at least one snapshot to detect vulnerabilities, wherein during the detection of the vulnerabilities by analyzing the at least one snapshot, the virtual machine is inactive . . . ." Wiz performs this step by, for example, analyzing the snapshot of a virtual disk to determine cyber vulnerabilities affecting the virtual disk. For example, Wiz analyzes the snapshot of a virtual disk to identify potential vulnerabilities.

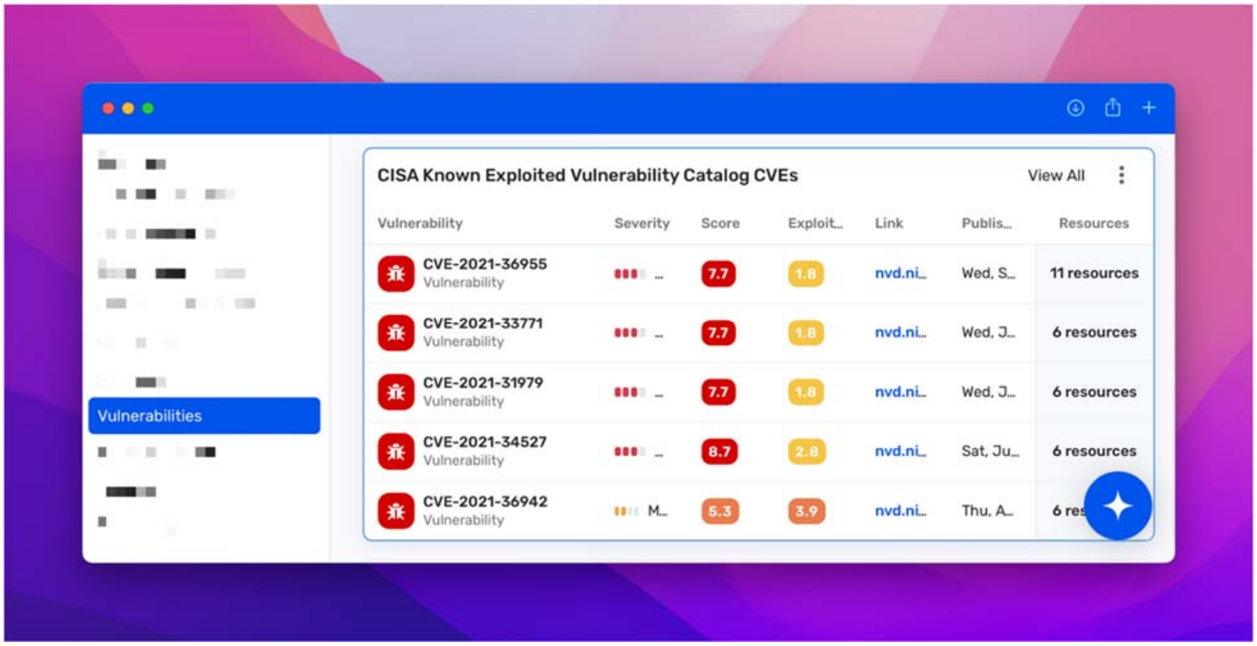


70K+ Supported Vulnerabilities: Our industry-leading vulnerability catalog consists of more than 70,000 supported vulnerabilities, across 30+ operating systems, CISA KEV catalog and thousands of applications.

https://www.wiz.io/solutions/vulnerability-management.

49. As another example, Wiz "analyzes [the] operating system, application layer, and data layer" of virtual machines to provide full visibility into vulnerabilities across the cloud computing environment. *See* Exhibit 6 at 4 (Wiz "[s]cans the workloads inside the container to determine . . . its vulnerabilities"); https://www.wiz.io/blog/uniting-builders-and-defenders-a-new-vision-for-cloud-security ("[D]efenders can now analyze activities and review timelines within the graph, with full context on the resource, roles, vulnerabilities, and potential impact."). Furthermore, Wiz analyzes snapshots of machines that are not online and/or "before deployment to the runtime environment." *See, e.g.*, https://legacy.wiz.io/partners/google ("Wiz uses a unique technology to scan deep within VMs and containers without needing an agent, analyzing all of your workloads even if a resource isn't online."); https://www.wiz.io/solutions/vulnerability-management; https://www.wiz.io/solutions/iac ("scan images continuously before deployment").

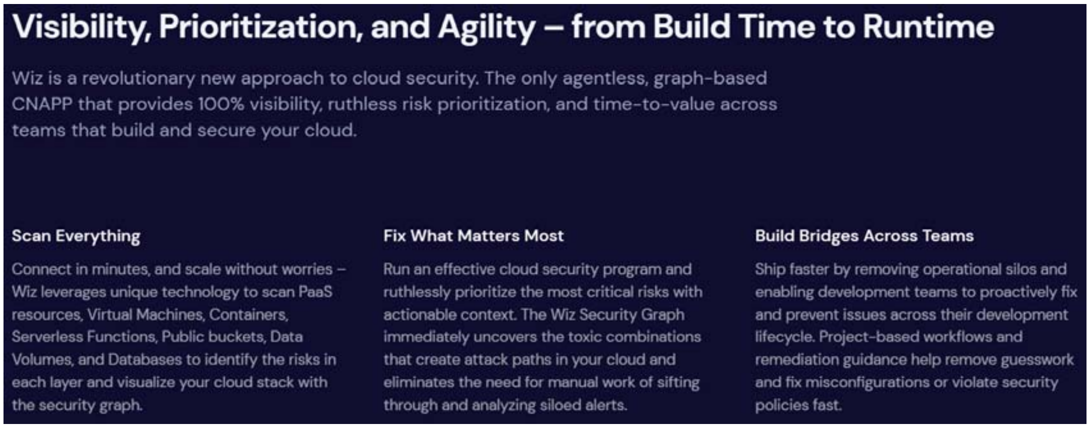
50. Claim 9 further recites "reporting the detected vulnerabilities as alerts." Wiz performs this step by, for example, reporting vulnerabilities in a client environment as alerts in Wiz's CSP.



CISA Known Exploited Vulnerability Catalog CVEs

Vulnerability	Severity	Score	Exploit...	Link	Publis...	Resources
CVE-2021-36955 Vulnerability	▮▮▮ ...	7.7	1.8	nvd.ni...	Wed, S...	11 resources
CVE-2021-33771 Vulnerability	▮▮▮ ...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-31979 Vulnerability	▮▮▮ ...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-34527 Vulnerability	▮▮▮ ...	8.7	2.8	nvd.ni...	Sat, Ju...	6 resources
CVE-2021-36942 Vulnerability	▮▮ M...	5.3	3.9	nvd.ni...	Thu, A...	6 res

https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz ("CISA Known Exploited Vulnerability Catalog CVEs dashboard in Wiz"); *see also* Exhibit 6 at 3 ("Scan for vulnerable and unpatched operating systems, installed software, and code libraries in your workloads prioritized by risk."). Wiz reports a "graph" to show "toxic combinations that create attack paths in [a] cloud."



Visibility, Prioritization, and Agility – from Build Time to Runtime

Wiz is a revolutionary new approach to cloud security. The only agentless, graph-based CNAPP that provides 100% visibility, ruthless risk prioritization, and time-to-value across teams that build and secure your cloud.

Scan Everything

Connect in minutes, and scale without worries – Wiz leverages unique technology to scan PaaS resources, Virtual Machines, Containers, Serverless Functions, Public buckets, Data Volumes, and Databases to identify the risks in each layer and visualize your cloud stack with the security graph.

Fix What Matters Most

Run an effective cloud security program and ruthlessly prioritize the most critical risks with actionable context. The Wiz Security Graph immediately uncovers the toxic combinations that create attack paths in your cloud and eliminates the need for manual work of sifting through and analyzing siloed alerts.

Build Bridges Across Teams

Ship faster by removing operational silos and enabling development teams to proactively fix and prevent issues across their development lifecycle. Project-based workflows and remediation guidance help remove guesswork and fix misconfigurations or violate security policies fast.

See, e.g., https://www.wiz.io/solutions/cnapp; https://www.wiz.io/blog/uniting-builders-and-defenders-a-new-vision-for-cloud-security ("[D]efenders can now analyze activities and review timelines within the graph, with full context on the resource, roles, vulnerabilities, and potential impact."); Exhibit 6 at 3 ("Wiz uses the full context of your cloud and combines this information into a single graph in order to correlate related issues"); https://www.wiz.io/solutions/vulnerability-management ("Use the Threat Center to immediately identify workload exposure to the latest vulnerabilities sourced from Wiz Research along with numerous third-party threat intelligence feeds.").

51. As described in the preceding paragraphs, Wiz infringes claim 9 of the '031 patent, either literally or under the doctrine of equivalents.

52. The above examples of how Wiz directly infringes claim 9 of the '031 patent are non-limiting and based on information currently available to Orca. In particular, additional or different aspects of Wiz's products or services may be identified that meet the limitations of claim 9 of the '031 patent, additional claims of the '031 patent may be determined to be infringed, and additional Wiz products or services may be identified as infringing once additional nonpublic information is provided through the course of discovery.

(b) Induced Infringement of the '031 Patent

53. On information and belief, in providing Wiz's CSP to its customers, Wiz has induced, and continues to induce, direct infringement of one or more claims of the '031 patent, including at least claim 9, literally and/or under the doctrine of equivalents pursuant to 35 U.S.C. § 271(b).

54. On information and belief, Wiz monitors Orca's patent portfolio and was aware of the '031 patent and its infringement thereof when the '031 patent issued or soon thereafter at least

as a result of its efforts to copy Orca's technology and its patents. For example, Wiz by and through its patent prosecution counsel had knowledge of the '031 patent's parent application, U.S. Patent Application No. 16/750,556, and its provisional application, U.S. Provisional Application No. 62/797,718, because Wiz's patent prosecution counsel is the same lawyer that filed those applications on behalf of Orca. As described above in Paragraph 22, Wiz's patents also include nearly identical figures and descriptions as those found in the '031 patent. In any event, Wiz has had knowledge of the '031 patent and its infringement thereof since at least as early as the filing of this Complaint.

55. On information and belief, Wiz possesses a specific intent to induce infringement by, at a minimum, providing user guides, instructions, sales-related material, and/or other supporting documentation, and by way of advertising, solicitation, and provision of product instruction materials, that instruct its customers on the normal operation of Wiz's CSP in a manner that infringes one or more claims of the '031 patent, including at least claim 9 of the '031 patent, or Wiz believed there was a high probability that the acts of its customers would infringe one or more claims of the '031 patent, including at least claim 9, and took deliberate steps to avoid learning of that infringement.

(c) Contributory Infringement of the '031 Patent

56. On information and belief, Wiz monitors Orca's patent portfolio and was aware of the '031 patent and its infringement thereof when the '031 patent issued or soon thereafter at least as a result of its efforts to copy Orca's technology and its patents. For example, Wiz by and through its patent prosecution counsel had knowledge of the '031 patent's parent application, U.S. Patent Application No. 16/750,556, and its provisional application, U.S. Provisional Application No. 62/797,718, because Wiz's patent prosecution counsel is the same lawyer that filed those

applications on behalf of Orca. As described above in Paragraph 22, Wiz's patents also include nearly identical figures and descriptions as those found in the '031 patent. In any event, Wiz has had knowledge of the '031 patent and its infringement thereof since at least as early as the filing of this Complaint.

57. On information and belief, by providing Wiz's CSP to its customers, Wiz has in the past contributed, and continues to contribute, to the direct infringement of one or more claims of the '031 patent, literally and/or under the doctrine of equivalents, in violation of 35 U.S.C. § 271(c), including at least claim 9 of the '031 patent. Wiz has contributorily infringed and continues to contribute to the infringement of one or more claims of the '031 patent by offering to sell or selling Wiz's CSP, which is a patented component, constituting a material part of the invention, knowing the same to be especially made or especially adapted for use in an infringement and not a staple article or commodity of commerce suitable for substantial non-infringing use.

(d) Willful Infringement of the '031 Patent

58. On information and belief, Wiz monitors Orca's patent portfolio and was aware of the '031 patent and its infringement thereof when the '031 patent issued or soon thereafter at least as a result of its efforts to copy Orca's technology and its patents. For example, Wiz by and through its patent prosecution counsel had knowledge of the '031 patent's parent application, U.S. Patent Application No. 16/750,556, and its provisional application, U.S. Provisional Application No. 62/797,718, because Wiz's patent prosecution counsel is the same lawyer that filed those applications on behalf of Orca. As described above in Paragraph 22, Wiz's patents also include nearly identical figures and descriptions as those found in the '031 patent. In any event, Wiz has had knowledge of the '031 patent and its infringement thereof since at least as early as the filing of this Complaint.

59. Wiz’s infringement has been and continues to be intentional and deliberate, entitling Orca to enhanced damages under 35 U.S.C. § 284 and a finding that this case is exceptional, entitling Orca to an award of reasonable attorneys’ fees under 35 U.S.C. § 285.

60. On information and belief, Wiz has profited from and will continue to profit from its infringing activities. Orca has been and will continue to be damaged and irreparably harmed by Wiz’s infringing activities. As a result, Orca is entitled to injunctive relief and damages adequate to compensate it for such infringement, in no event less than a reasonable royalty, in accordance with 35 U.S.C. §§ 271, 281, 283, and 284. The amount of monetary damages Wiz’s acts of infringement have caused to Orca cannot be determined without an accounting.

61. The harm to Orca from Wiz’s ongoing infringing activity is irreparable, continuing, and not fully compensable by money damages, and will continue unless Wiz’s infringing activities are enjoined.

COUNT II
(Infringement of the ’032 Patent)

62. Orca incorporates all other allegations in this Complaint.

63. The ’032 patent is entitled “Techniques for Securing Virtual Machines by Application Use Analysis,” and was duly and legally issued on May 30, 2023. A true and correct copy of the ’032 patent is attached hereto as Exhibit 2.

64. Orca is the current owner of all rights, title, and interest in the ’032 patent.

65. The ’032 patent is valid and enforceable.

66. The inventions claimed in the ’032 patent improve on prior art systems by, *inter alia*, accessing the snapshot of at least one virtual disk of a protected virtual cloud asset, analyzing the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications, and determining, based on the matching, an existence of

potential cyber vulnerabilities of the protected virtual cloud asset. *See, e.g.*, '032 patent at cls. 1-25. This novel analysis of snapshots for potential cyber vulnerabilities was not well understood, routine, or conventional. It is an inventive concept that allows, for example, practical implementations of vulnerability detection for virtual cloud assets in large data centers because it does not require the cumbersome installation of agents. This reduces the costs of licensing, deployment, integration, training, and support for a cloud security platform. Additionally, analyzing snapshots as provided in the claims of the '032 patent achieved unconventional performance, including (1) the ability to scan virtual cloud assets across an entire cloud environment in a matter of minutes compared to months-long installations of agent-based solutions, and (2) achieving comprehensive coverage and features that are not possible using agent-based approaches due to the tradeoff between performance and impact to the environment.

67. The claims of the '032 patent also recite inventive concepts for prioritizing potential cyber vulnerabilities based on use determinations and reporting prioritized alerts according to the use determinations. *See, e.g., id.* Prioritizing based on use determinations was not well understood, routine, or conventional, and improves on prior art techniques by putting potential cyber vulnerabilities in context. The novel limitations of the '032 patent invention, including analyzing snapshots and prioritizing potential cyber vulnerabilities based on use determinations, improve the implementation of a security system for cloud environments because the gathered information can be analyzed to produce actionable, context-based alerts and reports without relying on agents or network scanners.

(a) Direct Infringement of the '032 Patent

68. Wiz, without authorization, directly infringes one or more claims of the '032 patent, literally and/or under the doctrine of equivalents. Wiz infringes under 35 U.S.C. § 271 including,

without limitation, 35 U.S.C. § 271(a), by making, using, selling, offering to sell, and/or importing within the United States without authority, Wiz's CSP and other similar products or services, which includes (or is otherwise referred to) but is not limited to Wiz's CNAPP, CSPM, CIEM, DSPM, IaC scanning, and CDR platforms and/or features. *See* https://www.wiz.io/; *see also* https://www.wiz.io/product. Wiz's infringement includes infringement of, for example, claim 1 of the '032 patent,

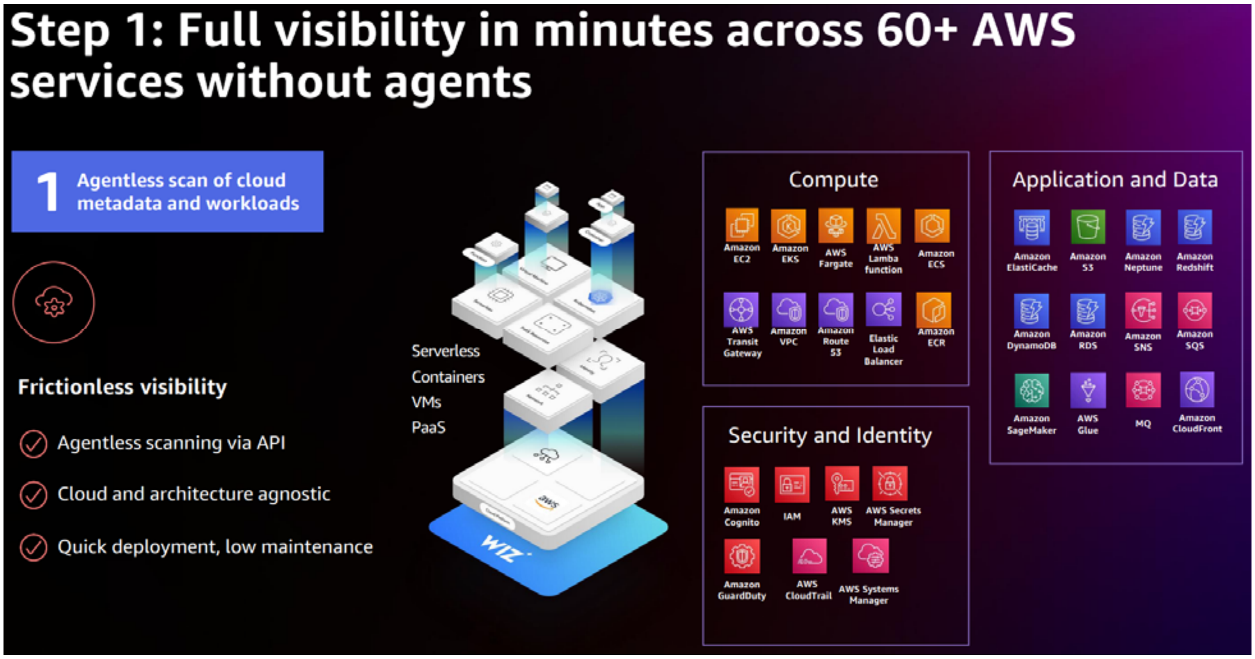
69. Claim 1 of the '032 patent recites:

> 1. A method for securing virtual cloud assets against cyber vulnerabilities in a cloud computing environment, the method comprising:
>
> determining, using an API or service provided by the cloud computing environment, a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the protected virtual cloud asset is instantiated in the cloud computing environment;
>
> accessing, based on the determined location and using an API or service provided by the cloud computing environment, the snapshot of the at least one virtual disk;
>
> analyzing the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications;
>
> determining, based on the matching, an existence of potential cyber vulnerabilities of the protected virtual cloud asset;
>
> determining whether the matching installed applications are used by the protected virtual cloud asset;
>
> prioritizing the potential cyber vulnerabilities based on the use determinations; and
>
> reporting the determined potential cyber vulnerabilities, as prioritized alerts according to the use determinations.

70. On information and belief, Wiz practices each and every limitation of claim 1 of the '032 patent by and through the use of Wiz's CSP and/or other similar products or services for Wiz's clients or customers.

71. The preamble of claim 1 recites "[a] method for securing virtual cloud assets against cyber vulnerabilities in a cloud computing environment, the method comprising . . . ." To the extent the preamble is limiting, Wiz practices this step by, for example, using Wiz's CSP to detect cyber vulnerabilities in cloud computing environments and secure virtual cloud assets within those environments against said vulnerabilities. *See, e.g.*, https://www.wiz.io/solutions/cnapp (advertising that Wiz "identif[ies] and remediate[s] risks and respond[s] to threats in [] cloud environments"); https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz ("Detect and prioritize CISA Known Exploited Vulnerabilities in the cloud with Wiz").

72. Claim 1 further recites "determining, using an API or service provided by the cloud computing environment, a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the protected virtual cloud asset is instantiated in the cloud computing environment . . . ." Wiz's public presentations and technical documentation confirm that Wiz practices this step by, for example, using Wiz's CSP to perform "[a]gentless scanning via API" provided by AWS, GCP, and Azure, among other cloud computing environments.



Step 1: Full visibility in minutes across 60+ AWS services without agents

1 Agentless scan of cloud metadata and workloads

Frictionless visibility

- ✓ Agentless scanning via API
- ✓ Cloud and architecture agnostic
- ✓ Quick deployment, low maintenance

See Exhibit 5 at 13; Exhibit 6 (supported cloud computing platforms include AWS, Azure, and Google Cloud Platform (GCP)). Wiz's technical documentation confirms that its agentless scanning includes "snapshot scanning" of instantiated virtual cloud assets, wherein Wiz "takes a snapshot of each VM system volume and analyzes its operating system, application layer, and data layer statically with no performance impact." Exhibit 6 at 4, 2 ("Wiz scans all the resources and workloads in your cloud environment using a unique snapshot technology that covers more than an agent can."); https://support.wiz.io/hc/en-us/articles/5641497256860-Azure-Connector-Basics ("Wiz connects to your cloud environment via your cloud service provider's APIs in order to extract metadata and perform snapshot scans."); https://support.wiz.io/hc/en-us/articles/5449816387100-AWS-Connector-Basics (same); https://support.wiz.io/hc/en-us/articles/5642208092572-GCP-Connector-Basics (same); https://www.wiz.io/solutions/vulnerability-management ("Using a one-time cloud native API deployment, continuously assess workloads without deploying agents").

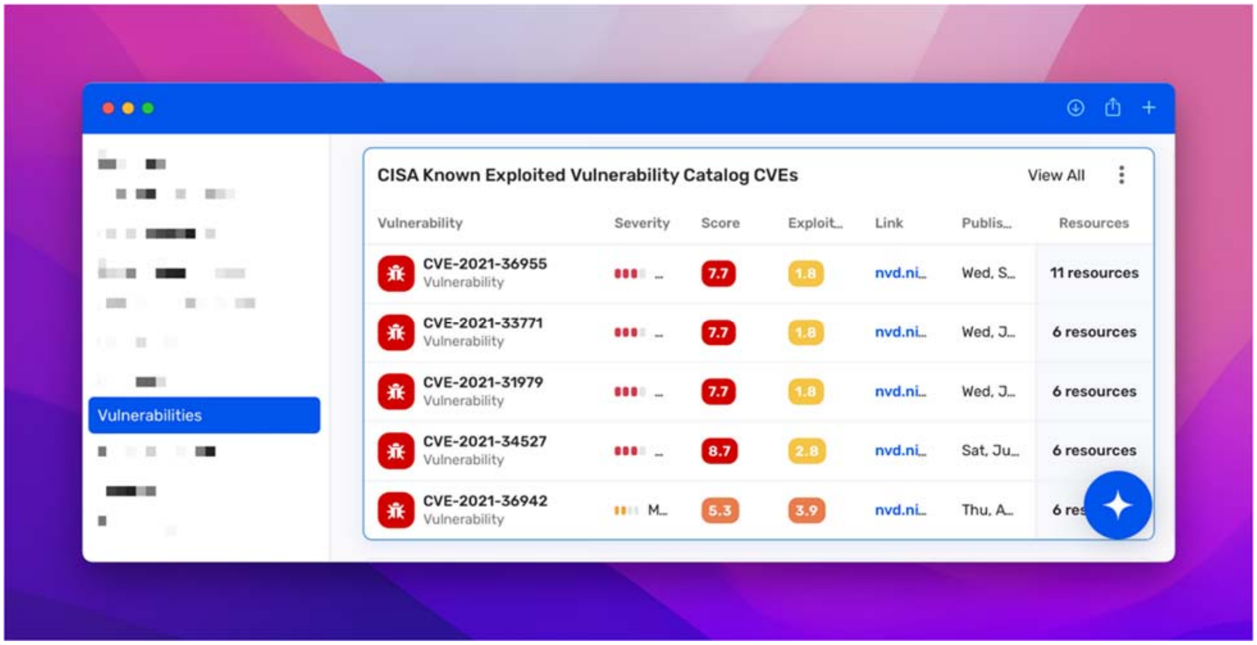
73. Claim 1 further recites "accessing, based on the determined location and using an API or service provided by the cloud computing environment, the snapshot of the at least one virtual disk . . . ." Wiz performs this step by, for example, accessing the snapshot of a virtual disk in order to "analyze[] [the] operating system, application layer, and data layer" of virtual cloud assets. *See* Exhibit 6 at 4, 3 (Wiz "[s]cans the workloads inside the container to determine . . . its vulnerabilities"). Wiz's technical documentation explains that "Wiz connects to [a] cloud environment via [a] cloud service provider's APIs in order to extract metadata and perform snapshot scans." https://support.wiz.io/hc/en-us/articles/5641497256860-Azure-Connector-Basics; https://support.wiz.io/hc/en-us/articles/5449816387100-AWS-Connector-Basics (same); https://support.wiz.io/hc/en-us/articles/5642208092572-GCP-Connector-Basics (same).

74. Claim 1 further recites "analyzing the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications . . . ." Wiz practices this step by, for example, analyzing the snapshot of a virtual disk by matching installed applications to a known list of vulnerabilities in the "CISA Known Exploited Vulnerability (KEV) Catalog," which is "a catalog of known exploited vulnerabilities that carry significant risk," including "vulnerabilities in . . . proprietary applications." *See* https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz. Wiz employs "agentless scanning" to "identify [] toxic combinations" between applications installed on a virtual disk and known vulnerable applications in Wiz's "vulnerability catalog consist[ing] of more than 70,000 supported vulnerabilities, across 30+ operating systems, CISA KEV catalog and thousands of applications."

70K+ Supported Vulnerabilities: Our industry-leading vulnerability catalog consists of more than 70,000 supported vulnerabilities, across 30+ operating systems, CISA KEV catalog and thousands of applications.

https://www.wiz.io/solutions/vulnerability-management; *see also id.*

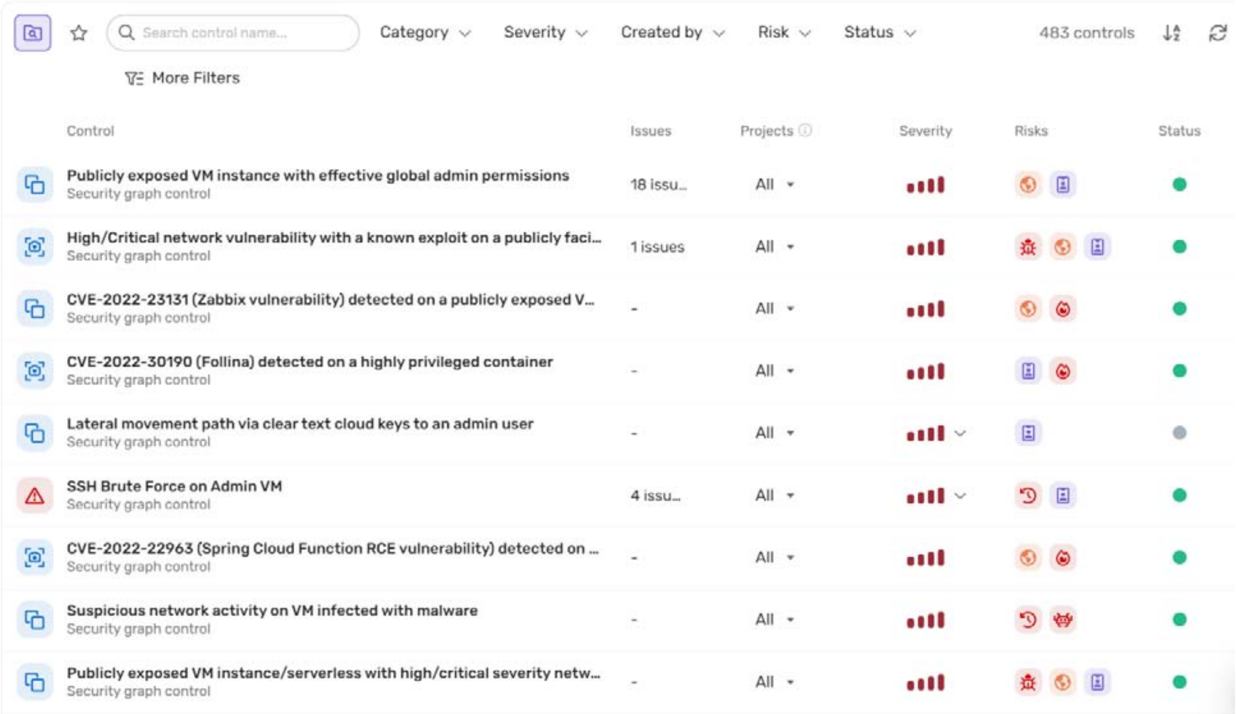
75. Claim 1 further recites "determining, based on the matching, an existence of potential cyber vulnerabilities of the protected virtual cloud asset . . . ." Wiz practices this step because, for example, it uses results of its agentless scanning to "list[] all the resources . . . that are currently vulnerable to one or more vulnerabilities in the catalog." *See* https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz.



CISA Known Exploited Vulnerability Catalog CVEs

Vulnerability	Severity	Score	Exploit...	Link	Publis...	Resources
CVE-2021-36955 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, S...	11 resources
CVE-2021-33771 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-31979 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-34527 Vulnerability	...	8.7	2.8	nvd.ni...	Sat, Ju...	6 resources
CVE-2021-36942 Vulnerability	M...	5.3	3.9	nvd.ni...	Thu, A...	6 res

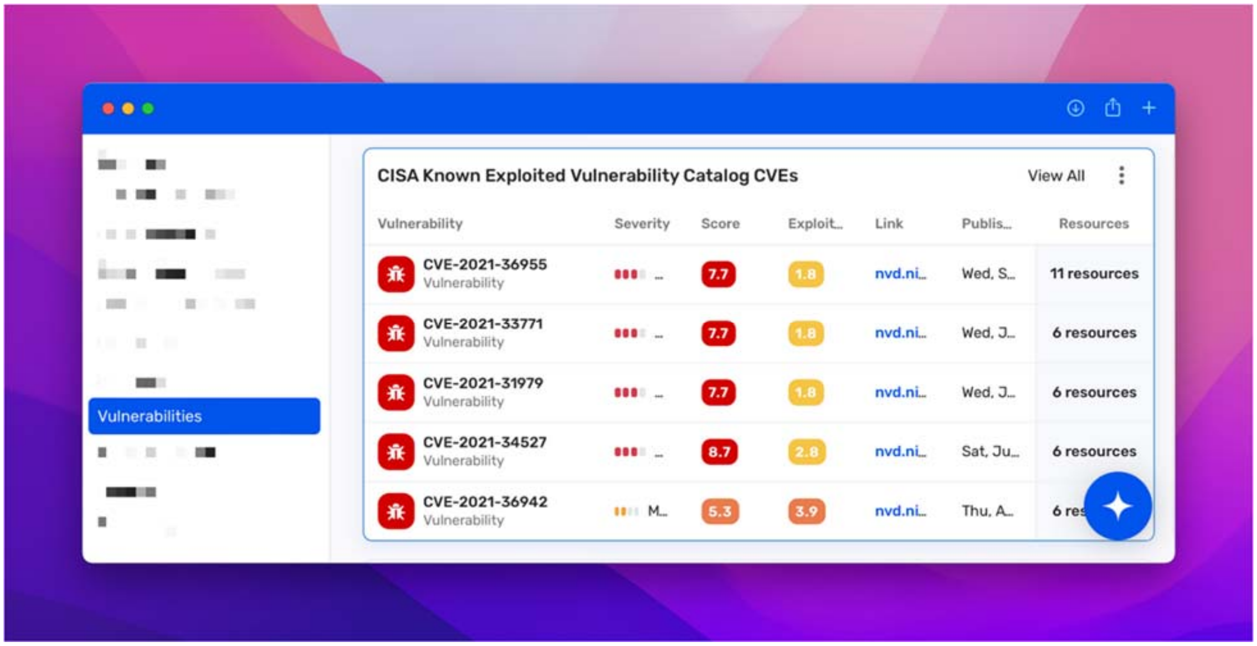
See id.



Control	Issues	Projects	Severity	Risks	Status
Publicly exposed VM instance with effective global admin permissions Security graph control	18 issu...	All	▮▮▮▮		●
High/Critical network vulnerability with a known exploit on a publicly faci... Security graph control	1 issues	All	▮▮▮▮		●
CVE-2022-23131 (Zabbix vulnerability) detected on a publicly exposed V... Security graph control	-	All	▮▮▮▮		●
CVE-2022-30190 (Follina) detected on a highly privileged container Security graph control	-	All	▮▮▮▮		●
Lateral movement path via clear text cloud keys to an admin user Security graph control	-	All	▮▮▮▮		●
SSH Brute Force on Admin VM Security graph control	4 issu...	All	▮▮▮▮		●
CVE-2022-22963 (Spring Cloud Function RCE vulnerability) detected on ... Security graph control	-	All	▮▮▮▮		●
Suspicious network activity on VM infected with malware Security graph control	-	All	▮▮▮▮		●
Publicly exposed VM instance/serverless with high/critical severity netw... Security graph control	-	All	▮▮▮▮		●

See also Exhibit 5 at 27 (listing "CVE" vulnerabilities, such as "CVE-2022-23131 (Zabbix Vulnerability)").

76. Claim 1 further recites "determining whether the matching installed applications are used by the protected virtual cloud asset . . . ." Wiz practices this step by, for example, determining whether applications in Wiz's vulnerability catalog are used by virtual cloud assets to determine what vulnerabilities "pose the highest risk to [a] cloud environment." *See* https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz.

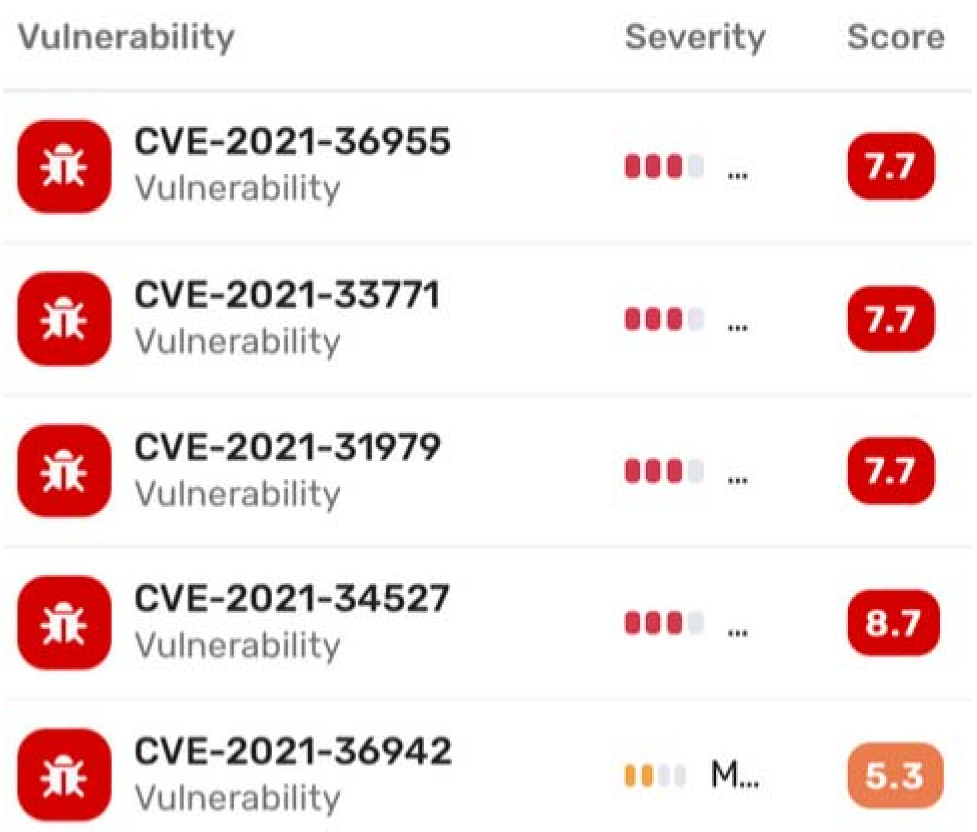


CISA Known Exploited Vulnerability Catalog CVEs

Vulnerability	Severity	Score	Exploit...	Link	Publis...	Resources
CVE-2021-36955 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, S...	11 resources
CVE-2021-33771 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-31979 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-34527 Vulnerability	...	8.7	2.8	nvd.ni...	Sat, Ju...	6 resources
CVE-2021-36942 Vulnerability	M...	5.3	3.9	nvd.ni...	Thu, A...	6 res

Id.; *see also* https://www.wiz.io/solutions/dspm ("Automatically correlate your sensitive data with underlying cloud context, including . . . how data assets are configured and used"); https://www.wiz.io/blog/monitor-detect-and-respond-to-cloud-data-risks-faster-with-built-in-security-controls ("[Y]ou can easily identify data resources with sensitive data that has traffic from an unrecommended IP."); https://www.wiz.io/blog/uncover-what-is-deployed-in-your-environment-with-enhanced-wiz-inventory ("The Wiz inventory already gives customers deep visibility into what cloud resources, applications, operating systems, and packages exist in their environment in minutes."); Exhibit 6 at 3 ("Wiz uses the full context of your cloud and combines this information into a single graph in order to correlate related issues").

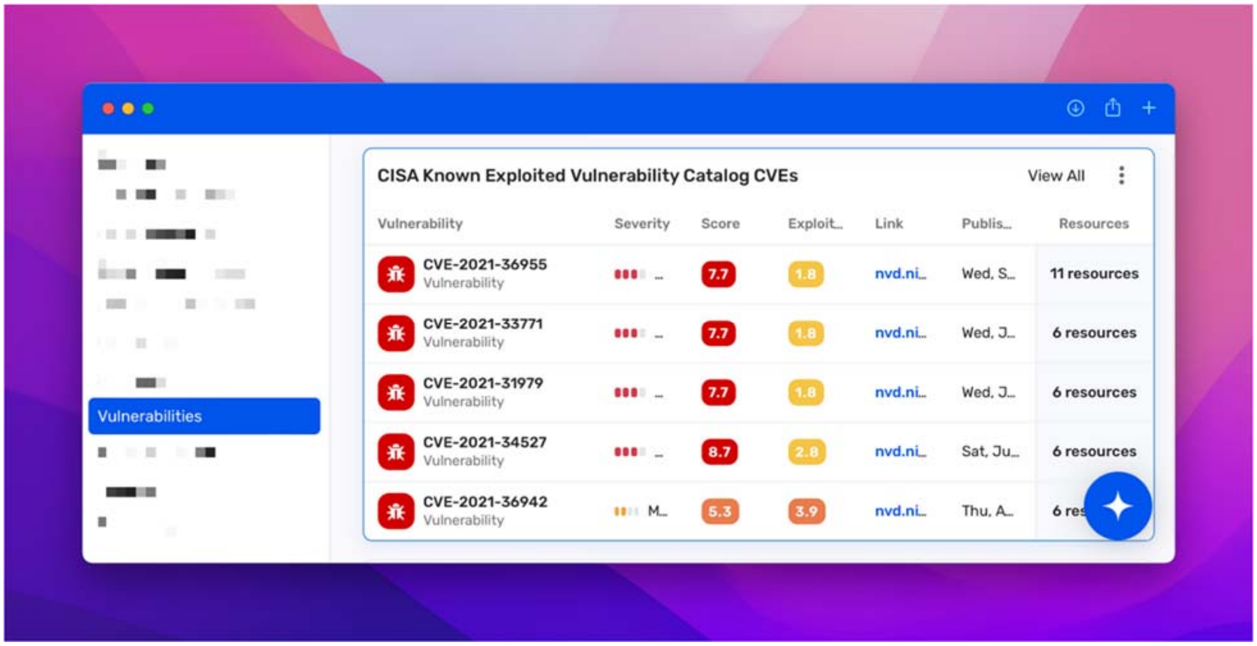
77. Claim 1 further recites "prioritizing the potential cyber vulnerabilities based on the use determinations . . . ." Wiz performs this step by, for example, using its vulnerability "catalog input . . . to better prioritize and mitigate the critical risks." *See* https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz. Wiz also prioritizes cyber vulnerabilities based on one or more of "Severity," "Score," and "exploitability" ratings.



Vulnerability	Severity	Score
CVE-2021-36955 Vulnerability	●●●○ ...	7.7
CVE-2021-33771 Vulnerability	●●●○ ...	7.7
CVE-2021-31979 Vulnerability	●●●○ ...	7.7
CVE-2021-34527 Vulnerability	●●●○ ...	8.7
CVE-2021-36942 Vulnerability	●●○○ M...	5.3

See id.; *see also* Exhibit 5 at 27 (same).

78. Claim 1 further recites "reporting the determined potential cyber vulnerabilities, as prioritized alerts according to the use determinations." Wiz performs this step by, for example, reporting "Vulnerabilit[ies]," prioritized according to "Severity," "Score," and/or "exploitability" through its "CISA Known Exploited Vulnerability Catalog CVEs dashboard."



CISA Known Exploited Vulnerability Catalog CVEs

Vulnerability	Severity	Score	Exploit...	Link	Publis...	Resources
CVE-2021-36955 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, S...	11 resources
CVE-2021-33771 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-31979 Vulnerability	...	7.7	1.8	nvd.ni...	Wed, J...	6 resources
CVE-2021-34527 Vulnerability	...	8.7	2.8	nvd.ni...	Sat, Ju...	6 resources
CVE-2021-36942 Vulnerability	M...	5.3	3.9	nvd.ni...	Thu, A...	6 res

See https://www.wiz.io/blog/detect-and-prioritize-cisa-known-exploited-vulnerabilities-kev-with-wiz; *see also* Exhibit 5 at 27 (prioritizing vulnerabilities according to "Severity"); https://www.wiz.io/blog/monitor-detect-and-respond-to-cloud-data-risks-faster-with-built-in-security-controls (Wiz "detect[s] and alert[s] on suspicious events and threats using rules continuously updated by Wiz Research."); https://www.wiz.io/solutions/dspm ("Automatically correlate your sensitive data with underlying cloud context, including . . . how data assets are configured and used").

79. As described in the preceding paragraphs, Wiz practices each limitation of claim 1 of the '032 patent, either literally or under the doctrine of equivalents.

80. The above examples of how Wiz directly infringes claim 1 of the '032 patent are non-limiting and based on information currently available to Orca. In particular, additional or different aspects of Wiz's products or services may be identified that meet the limitations of claim 1 of the '032 patent, additional claims of the '032 patent may be determined to be infringed, and

additional Wiz products or services may be identified as infringing once additional nonpublic information is provided through the course of discovery.

(b) Induced Infringement of the ’032 Patent

81. On information and belief, in providing Wiz’s CSP to its customers, Wiz has induced, and continues to induce, direct infringement of one or more claims of the ’032 patent, including at least claim 1, literally and/or under the doctrine of equivalents pursuant to 35 U.S.C. § 271(b).

82. On information and belief, Wiz monitors Orca’s patent portfolio and was aware of the ’032 patent and its infringement thereof when the ’032 patent issued or soon thereafter at least as a result of its efforts to copy Orca’s technology and its patents. Additionally, Wiz by and through its patent prosecution counsel had knowledge of the ’032 patent’s parent application, U.S. Patent Application No. 17/330,998, and its provisional application, U.S. Provisional Application No. 62/797,718, because Wiz’s patent prosecution counsel is the same lawyer that filed those applications on behalf of Orca. As described above in Paragraph 22, Wiz’s patents also include nearly identical figures and descriptions as those found in the ’032 patent. In any event, Wiz has had knowledge of the ’032 patent and its infringement thereof since at least as early as the filing of this Complaint.

83. On information and belief, Wiz possesses a specific intent to induce infringement by, at a minimum, providing user guides, instructions, sales-related material, and/or other supporting documentation, and by way of advertising, solicitation, and provision of product instruction materials, that instruct its customers on the normal operation of Wiz’s CSP in a manner that infringes one or more claims of the ’032 patent, including at least claim 1 of the ’032 patent, or, in the alternative, Wiz believed there was a high probability that the acts of its customers would

infringe one or more claims of the ’032 patent, including at least claim 1, and took deliberate steps to avoid learning of that infringement.

(c) Contributory Infringement of the ’032 Patent

84. On information and belief, Wiz monitors Orca’s patent portfolio and was aware of the ’032 patent and its infringement thereof when the ’032 patent issued or soon thereafter at least as a result of its efforts to copy Orca’s technology and its patents. Additionally, Wiz by and through its patent prosecution counsel had knowledge of the ’032 patent’s parent application, U.S. Patent Application No. 17/330,998, and its provisional application, U.S. Provisional Application No. 62/797,718, because Wiz’s patent prosecution counsel is the same lawyer that filed those applications on behalf of Orca. As described above in Paragraph 22, Wiz’s patents also include nearly identical figures and descriptions as those found in the ’032 patent. In any event, Wiz has had knowledge of the ’032 patent and its infringement thereof since at least as early as the filing of this Complaint.

85. By providing Wiz’s CSP to its customers, Wiz has in the past contributed, and continues to contribute, to the direct infringement of one or more claims of the ’032 patent, literally and/or under the doctrine of equivalents, in violation of 35 U.S.C. § 271(c), including at least claim 1 of the ’032 patent. Wiz has contributorily infringed and continues to contribute to the infringement of one or more claims of the ’032 patent by offering to sell or selling Wiz’s CSP, which is a patented component, constituting a material part of the invention, knowing the same to be especially made or especially adapted for use in an infringement and not a staple article or commodity of commerce suitable for substantial non-infringing use.

(d) Willful Infringement of the ’032 Patent

86. On information and belief, Wiz monitors Orca's patent portfolio and was aware of the '032 patent and its infringement thereof when the '032 patent issued or soon thereafter at least as a result of its efforts to copy Orca's technology and its patents. Additionally, Wiz by and through its patent prosecution counsel had knowledge of the '032 patent's parent application, U.S. Patent Application No. 17/330,998, and its provisional application, U.S. Provisional Application No. 62/797,718, because Wiz's patent prosecution counsel is the same lawyer that filed those applications on behalf of Orca. As described above in Paragraph 22, Wiz's patents also include nearly identical figures and descriptions as those found in the '032 patent. In any event, Wiz has had knowledge of the '032 patent and its infringement thereof since at least as early as the filing of this Complaint.

87. Wiz's infringement has been and continues to be intentional and deliberate, entitling Orca to enhanced damages under 35 U.S.C. § 284 and a finding that this case is exceptional, entitling Orca to an award of reasonable attorneys' fees under 35 U.S.C. § 285.

88. On information and belief, Wiz has profited from and will continue to profit from its infringing activities. Orca has been and will continue to be damaged and irreparably harmed by Wiz's infringing activities. As a result, Orca is entitled to injunctive relief and damages adequate to compensate it for such infringement, in no event less than a reasonable royalty, in accordance with 35 U.S.C. §§ 271, 281, 283, and 284. The full amount of monetary damages Wiz's acts of infringement have caused to Orca cannot be determined without an accounting.

89. The harm to Orca from Wiz's ongoing infringing activity is irreparable, continuing, and not fully compensable by money damages, and will continue unless Wiz's infringing activities are enjoined.

PRAYER FOR RELIEF

WHEREFORE, Orca respectfully asks that the Court enter judgment against Wiz and in favor of Orca as follows:

90. A judgment that Wiz has infringed and continues to infringe (either literally or under the doctrine of equivalents) one or more claims of the Asserted Patents under at least 35 U.S.C. § 271(a);

91. A judgment that Wiz has induced and continues to induce others to infringe one or more claims of the Asserted Patents under at least 35 U.S.C. § 271(b);

92. A judgment that Wiz has contributorily infringed and continues to contribute to the infringement of one or more claims of the Asserted Patents under at least 35 U.S.C. § 271(c);

93. A judgment that Wiz's infringement of the Asserted Patents has been and continues to be willful;

94. An award of monetary damages sufficient to compensate Orca for Wiz's patent infringement, with interest, pursuant to at least 35 U.S.C. § 284;

95. A preliminary and permanent injunction prohibiting Wiz and its officers, agents, representatives, assigns, licenses, distributors, servants, employees, related entities, attorneys, and all those acting in concert, privity, or participation with them, from:

A. infringing or inducing the infringement of any claim of the Asserted Patents; and

B. soliciting any new business or new customers using any information or materials that Orca derived from its infringement of the Asserted Patents;

96. An award of enhanced damages of three times the amount found or assessed for Wiz's willful patent infringement, pursuant to at least 35 U.S.C. § 284, including interest on such damages;

97. An order finding this case exceptional and awarding Orca its attorneys' fees, to be obtained from any and all of Wiz's assets, pursuant to 35 U.S.C. § 285, including prejudgment interest on such fees;

98. An accounting and supplemental damages for all damages occurring after the period for which discovery is taken, and after discovery closes, through the Court's decision regarding the imposition of a permanent injunction;

99. An award of Orca's costs and expenses of this suit as the prevailing party; and

100. Any and all other relief that the Court deems just and proper.

JURY DEMAND

Orca hereby demands a trial by jury on all issues so triable.

MORRIS, NICHOLS, ARSHT &TUNNELL LLP

/s/ Jack B. Blumenfeld

Jack B. Blumenfeld (#1014)
Rodger D. Smith II (#3778)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899-1347
(302) 658-9200
jblumenfeld@morrisnichols.com
rsmith@morrisnichols.com

Attorneys for Plaintiff Orca Security Ltd.

OF COUNSEL:

Douglas E. Lumish
Lucas Lonergan
LATHAM & WATKINS LLP
140 Scott Drive
Menlo Park, CA 94025
(650) 328-4600

Blake R. Davis
LATHAM & WATKINS LLP
505 Montgomery Street, Suite 2000
San Francisco, CA 94111
(415) 391-0600

July 12, 2023

EXHIBIT 1

US011663031B2

(12) **United States Patent**
Shua

(10) **Patent No.: US 11,663,031 B2**
(45) **Date of Patent: May 30, 2023**

(54) **TECHNIQUES FOR SECURING VIRTUAL CLOUD ASSETS AT REST AGAINST CYBER THREATS**

(71) Applicant: **Orca Security LTD.**, Tel Aviv (IL)

(72) Inventor: **Avi Shua**, Tel Aviv (IL)

(73) Assignee: **ORCA SECURITY LTD.**, Tel Aviv (IL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **17/400,364**

(22) Filed: **Aug. 12, 2021**

(65) **Prior Publication Data**

US 2021/0377287 A1 Dec. 2, 2021

Related U.S. Application Data

(63) Continuation of application No. 16/750,556, filed on Jan. 23, 2020.

(60) Provisional application No. 62/797,718, filed on Jan. 28, 2019.

(51) **Int. Cl.**

H04L 9/40	(2022.01)
G06F 9/455	(2018.01)
G06F 16/11	(2019.01)
G06F 11/14	(2006.01)

(52) **U.S. Cl.**

CPC ***H04L 63/1416*** (2013.01); ***G06F 9/45558*** (2013.01); ***G06F 11/1464*** (2013.01); ***G06F 16/128*** (2019.01); ***H04L 63/1433*** (2013.01); ***H04L 63/1441*** (2013.01); *G06F 2009/45562* (2013.01); *G06F 2009/45583* (2013.01); *G06F 2009/45587* (2013.01); *G06F 2009/45591* (2013.01); *G06F 2009/45595* (2013.01); *G06F 2201/84* (2013.01)

(58) **Field of Classification Search**

CPC H04L 63/1416; H04L 63/1433; H04L 63/1441; G06F 9/45558; G06F 2009/45562; G06F 2009/45591; G06F 2009/45587; G06F 2201/84

USPC 726/25

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

9,092,625	B1 *	7/2015	Kashyap	G06F 21/52
9,177,145	B2	11/2015	Todorović	
9,519,781	B2	12/2016	Golshan et al.	
9,563,777	B2	2/2017	Deng et al.	
9,798,885	B2	10/2017	Deng et al.	

(Continued)

OTHER PUBLICATIONS

NPL Search Terms (Year: 2021).*

(Continued)

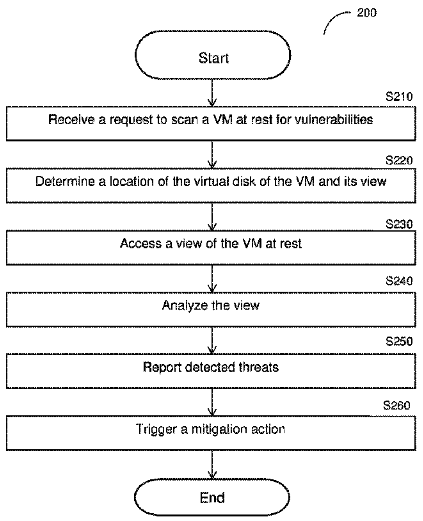
Primary Examiner — Syed A Zaidi

(74) *Attorney, Agent, or Firm* — Finnegan, Henderson, Farabow, Garrett & Dunner, LLP

(57) **ABSTRACT**

A method and system for securing virtual cloud assets at rest against cyber threats. The method comprises determining a location of a view of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is at rest and, when activated, instantiated in the cloud computing environment; accessing the view of the virtual disk based on the determined location; analyzing the view of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset, wherein the virtual cloud asset is inactive during the analysis; and alerting detected potential cyber threats based on a determined priority.

16 Claims, 4 Drawing Sheets


200
Start
S210
Receive a request to scan a VM at rest for vulnerabilities
S220
Determine a location of the virtual disk of the VM and its view
S230
Access a view of the VM at rest
S240
Analyze the view
S250
Report detected threats
S260
Trigger a mitigation action
End

(56) References Cited

U.S. PATENT DOCUMENTS

10,339,011	B1 *	7/2019	Bansal	G06F 16/188
10,412,109	B2	9/2019	Loureiro et al.	
10,536,471	B1 *	1/2020	Derbeko	G06F 21/53
10,552,610	B1 *	2/2020	Vashisht	G06F 3/0619
2007/0266433	A1	11/2007	Moore	
2008/0155223	A1 *	6/2008	Hiltgen	G06F 21/6218 718/1
2008/0263658	A1 *	10/2008	Michael	G06F 21/562 726/22
2009/0007100	A1	1/2009	Field	
2010/0070726	A1 *	3/2010	Ngo	G06F 11/1469 711/162
2013/0262801	A1 *	10/2013	Sancheti	H04L 67/1095 711/162
2013/0268763	A1 *	10/2013	Sweet	G06F 21/56 713/176
2014/0137190	A1	5/2014	Carey et al.	
2014/0173723	A1	6/2014	Singla	
2015/0052520	A1	2/2015	Crowell et al.	
2015/0161151	A1 *	6/2015	Koryakina	G06F 11/1451 711/114
2017/0011138	A1	1/2017	Venkatesh et al.	
2017/0076092	A1 *	3/2017	Kashyap	G06F 21/316
2018/0255080	A1	9/2018	Paine	
2018/0293374	A1	10/2018	Chen	

OTHER PUBLICATIONS

Pandey, Anjali, and Shashank Srivastava. "An approach for virtual machine image security." 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014). IEEE, 2014. (Year: 2014).*
NPL Search Terms (Year: 2022).*
U.S. Patent and Trademark Office, Non-final Office Action, dated Jul. 21, 2022, 24 pp. for U.S. Appl. No. 16/750,556 (filing date Jan. 23, 2020).

* cited by examiner

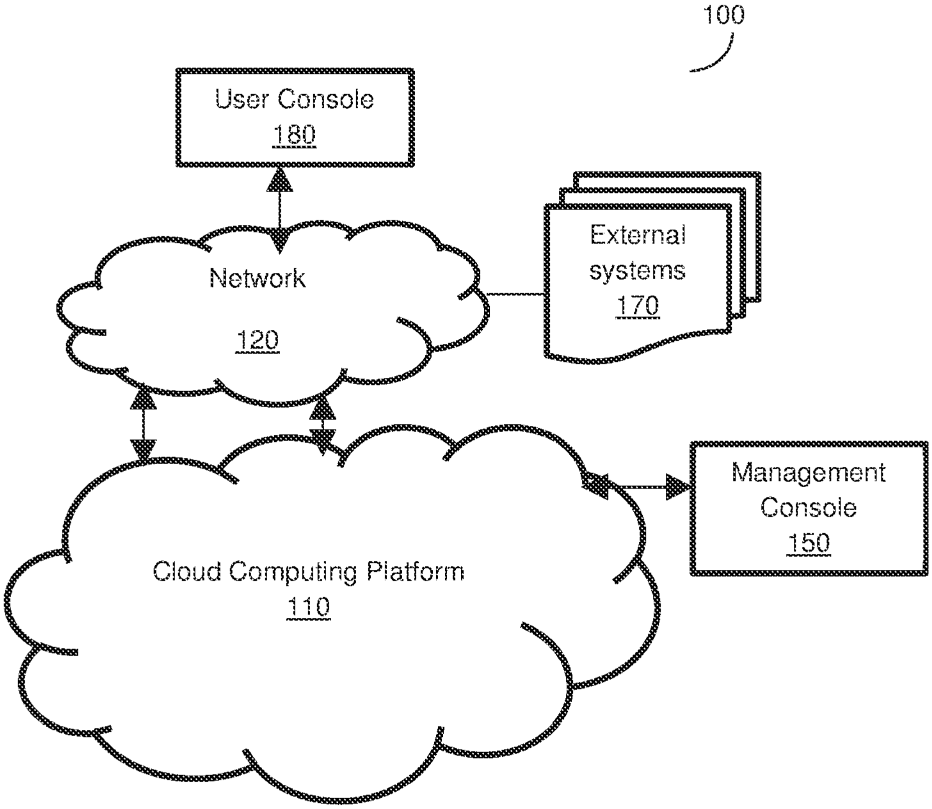

100
User Console
180
Network
120
External
systems
170
Cloud Computing Platform
110
Management
Console
150

FIG. 1A

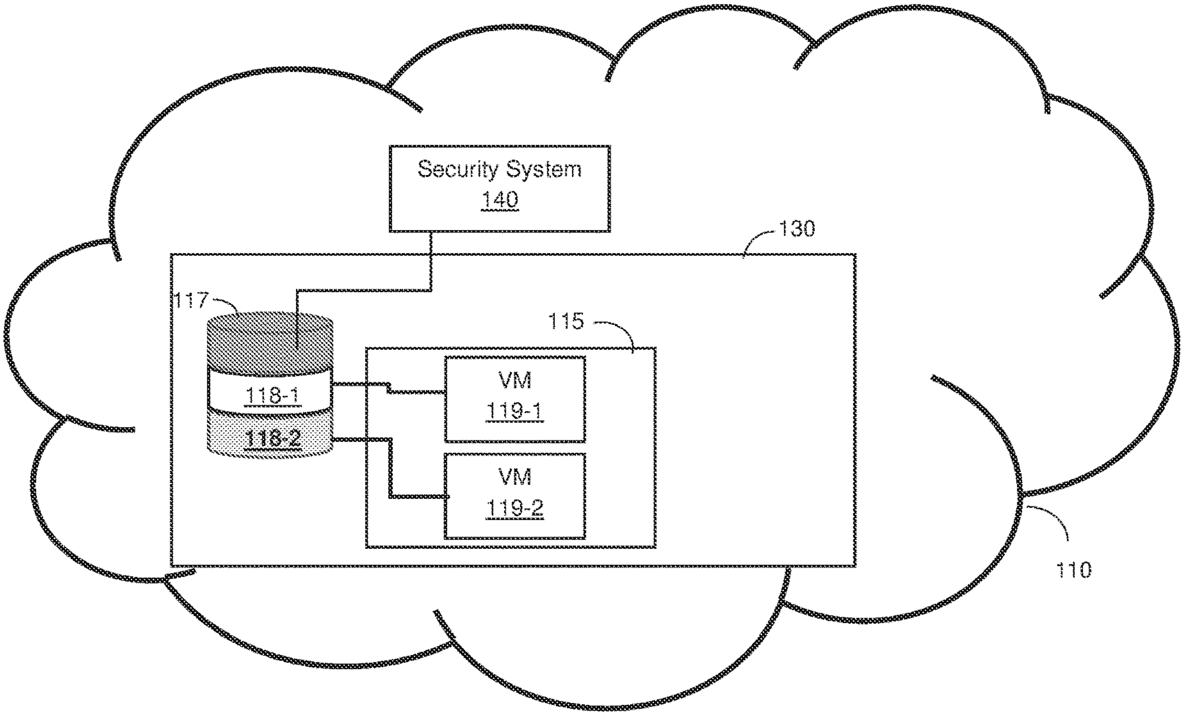


FIG. 1B

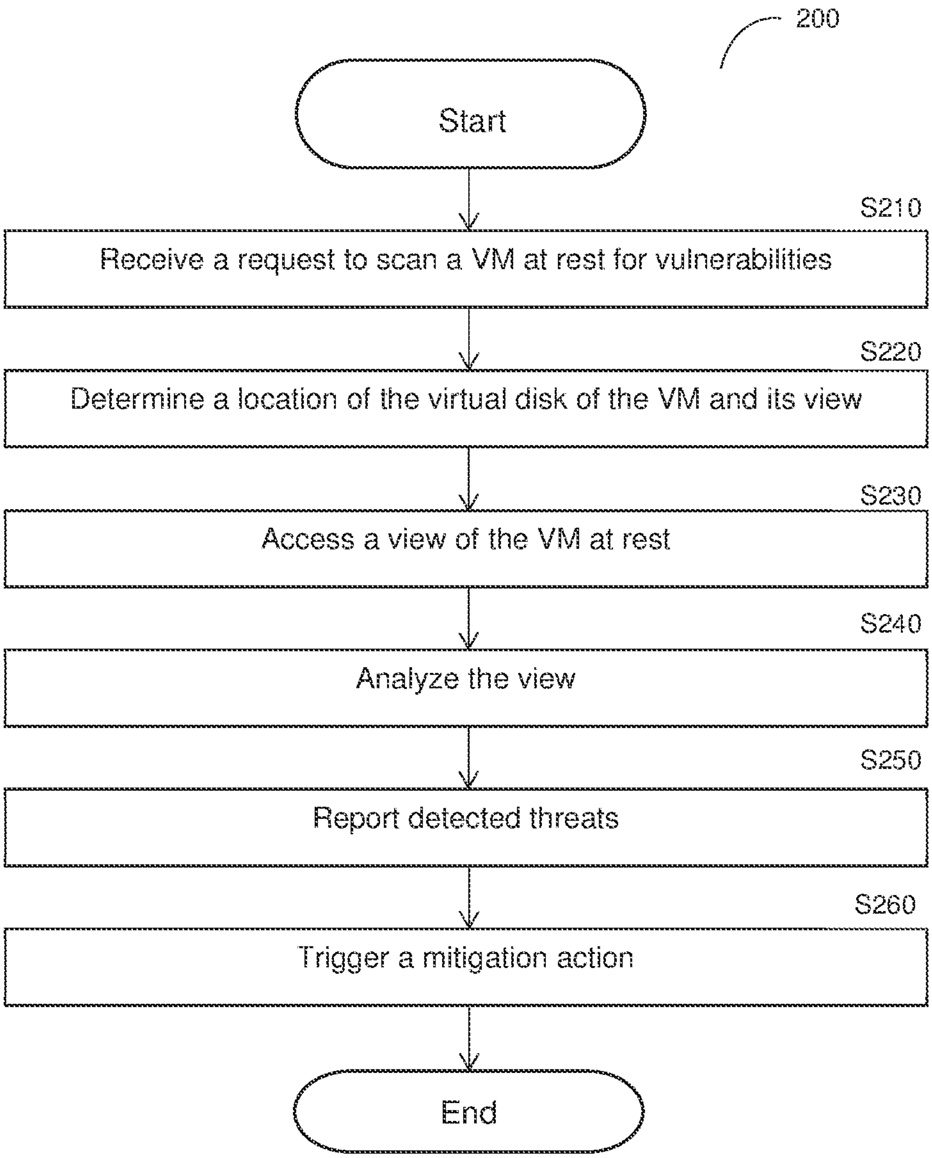

200
Start
S210
Receive a request to scan a VM at rest for vulnerabilities
S220
Determine a location of the virtual disk of the VM and its view
S230
Access a view of the VM at rest
S240
Analyze the view
S250
Report detected threats
S260
Trigger a mitigation action
End

FIG. 2

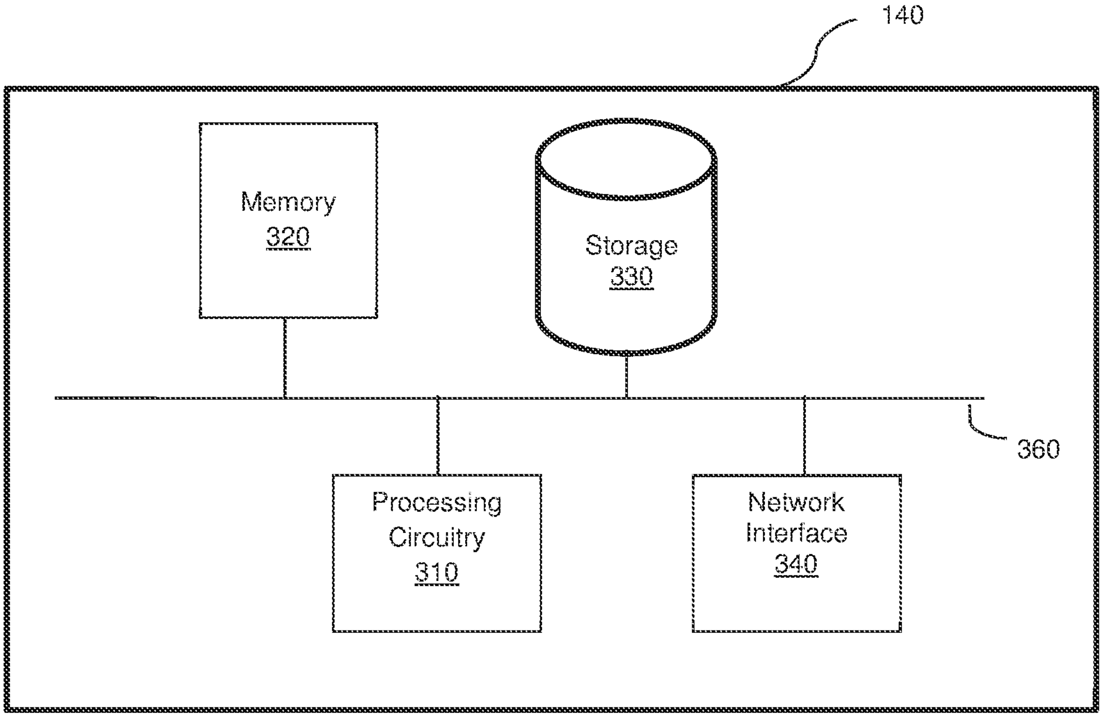

140
Memory
320
Storage
330
360
Processing
Circuitry
310
Network
Interface
340

FIG. 3

TECHNIQUES FOR SECURING VIRTUAL CLOUD ASSETS AT REST AGAINST CYBER THREATS

This application is a continuation of U.S. application Ser. No. 16/750,556, filed Jan. 23, 2020, now pending, which claims the benefit of U.S. Provisional Application No. 62/797,718 filed on Jan. 28, 2019. Each of the above referenced applications are incorporated herein by reference in its entirety.

TECHNICAL FIELD

This disclosure relates generally to cyber-security systems and, more specifically, to techniques for securing virtual machines.

BACKGROUND

Organizations have increasingly adapted their applications to be run from multiple cloud computing platforms. Some leading public cloud service providers include Amazon®, Microsoft®, Google®, and the like.

Virtualization plays a key role in a cloud computing, allowing multiple applications and users to share the same cloud computing infrastructure. For example, a cloud storage service can maintain data of multiple different users.

In one instance, virtualization can be achieved by means of virtual machines. A virtual machine emulates a number of “computers” or instances, all within a single physical device. In more detail, virtual machines provide the ability to emulate a separate operating system (OS), also referred to as a guest OS, and, therefore, a separate computer, from an existing OS (the host). This independent instance is typically isolated as a completely standalone environment.

Modern virtualization technologies are also adapted by cloud computing platforms. Examples for such technologies include virtual machines, software containers, and serverless functions. With their computing advantages, applications and virtual machines running on top of virtualization technologies are also vulnerable to some cyber threats. For example, virtual machines can execute vulnerable software applications or infected operating systems.

Protection of a cloud computing infrastructure, and, particularly, of virtual machines, can be achieved via inspection of traffic. Traditionally, traffic inspection is performed by a network device connected between a client and a server (deployed in a cloud computing platform or a data center) hosting virtual machines. Traffic inspection may not provide an accurate indication of the security status of the server due to inherent limitations, such as encryption and whether the necessary data is exposed in the communication.

Furthermore, inspection of computing infrastructure may be performed by a network scanner deployed out of path. The scanner queries the server to determine if the server executes an application that possess a security threat, such as vulnerability in the application. The disadvantage of such a scanner is that the server may not respond to all queries by the scanner or that the server may not expose the necessary data in the response. Further, the network scanner usually communicates with the server, and the network configuration may prevent such communication. In addition, some types of queries may require credentials to access the server. Such credentials may not be available to the scanner.

Traffic inspection may also be performed by a traffic monitor that listens to traffic flows between clients and the server. The traffic monitor can detect some cyber threats,

e.g., based on the volume of traffic. However, the monitor can detect threats only based on the monitored traffic. For example, misconfiguration of the server may not be detected by the traffic monitor. As such, traffic monitoring would not allow for detection of vulnerabilities in software executed by the server.

To overcome the limitations of traffic inspection solutions, some cyber-security solutions, such as vulnerability management and security assessment solutions, are based on agents installed in each server in a cloud computing platform or data center. Using agents is a cumbersome solution for a number of reasons, including IT resource management, governance, and performance. For example, installing agents in a large data center may take months.

Further, traffic monitoring does not allow detection of vulnerabilities in data at rest. Data at rest, in information technology, means inactive data that is stored physically in any digital form. Data at rest may include data, services, and/or services that are inactive but can be accessed or executed as needed. Similarly, in cloud computing, some machines (e.g., virtual machines) may also be at rest. Some machines are configured with applications or services which are infrequently executed. For example, such a machine may be utilized during one month of the year and remain inactive for the rest in the year. While at rest, the machines are powered off, and are not inspected for vulnerabilities, simply because scanners and/or installed monitoring agents cannot operate on a powered-off machine.

Another attempt would be to scan a machine at rest when the machine is powered on and preserving a log of its latest status. However, this would require keeping an updated log of the machine’s configurations and all its applications. Further, as threats constantly evolve, scanning based on past information may not be relevant. As such, when data or a machine at rest becomes active, undetected vulnerabilities can pose cyber threats.

It would therefore be advantageous to provide a security solution that would overcome the deficiencies noted above.

SUMMARY

A summary of several example embodiments of the disclosure follows. This summary is provided for the convenience of the reader to provide a basic understanding of such 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 concepts of one or more embodiments in a simplified form as a prelude to the more detailed description that is presented later. For convenience, the term “some embodiments” or “certain embodiments” may be used herein to refer to a single embodiment or multiple embodiments of the disclosure.

Certain embodiments disclosed herein include a method for securing virtual cloud assets at rest against cyber threats. The method comprises determining a location of a view of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is at rest and, when activated, instantiated in the cloud computing environment; accessing the view of the virtual disk based on the determined location; analyzing the view of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset, wherein the virtual cloud asset is inactive during the analysis; and alerting detected potential cyber threats based on a determined priority.

Certain embodiments disclosed herein also include a system for securing virtual cloud assets at rest against cyber threats, comprising: a processing circuitry; and a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to: determine a location of a view of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is at rest and, when activated, instantiated in a cloud computing environment; access the view of the virtual disk based on the determined location; analyze the view of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset, wherein the virtual cloud asset is inactive during the analysis; and alert detected potential cyber threats based on a determined priority.

BRIEF DESCRIPTION OF THE DRAWINGS

The subject matter disclosed herein is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the disclosed embodiments will be apparent from the following detailed description taken in conjunction with the accompanying drawings.

FIGS. **1**A and **1**B are network diagrams utilized to describe the various embodiments.

FIG. **2** is a flowchart illustrating a method detecting cyber threats, including potential vulnerabilities in virtual machines executed in a cloud computing platform according to some embodiments.

FIG. **3** is an example block diagram of the security system according to an embodiment.

DETAILED DESCRIPTION

It is important to note that the embodiments disclosed herein are only examples of the many advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily limit any of the various claimed embodiments. Moreover, some statements may apply to some inventive features but not to others. In general, unless otherwise indicated, singular elements may be in plural and vice versa with no loss of generality. In the drawings, like numerals refer to like parts through several views.

Various techniques disclosed herein include techniques for securing data at rest or machines at rest (collectively referred to as "machines at rest"). Data at rest may include inactive data that is stored physically in any digital form. Machines at rest may include a virtual machine configured service(s) and/or application(s) that are inactive but can be accessed or executed as needed. The applications and/or services in such machines at rest are infrequently executed. The disclosed techniques are utilized to scan for embedded vulnerabilities in machines at rest, when the machine is powered off. For example, a machine at rest may be utilized during one month of the year and remain inactive for the rest in the year. According to the disclosed embodiments, the machine is scanned for vulnerabilities when it is in its inactive step.

FIGS. **1**A and **1**B show an example network diagram **100** utilized to describe the various embodiments. A cloud computing platform **110** is communicably connected to a network **120**. Examples of the cloud computing platform **110** may include a public cloud, a private cloud, a hybrid cloud, and the like. Examples of a public cloud include, but are not limited to, AWS® by Amazon®, Microsoft Azure®, Google Cloud®, and the like. In some configurations, the disclosed embodiments may be operable in on-premises virtual machine environments. The network **120** may be the Internet, the world-wide-web (WWW), a local area network (LAN), a wide area network (WAN), and other networks.

The arrangement of the example cloud computing platform **110** is shown in FIG. **1**B. As illustrated, the platform **110** includes a server **115** and a storage **117**, serving as the storage space for the server **115**. The server **115** is a physical device hosting one or more virtual machines (VMs). In the example FIG. **1**B, two VMs **119-1** and **119-2** are shown, and both are protected entities. It should be noted that such a protected entity may be any virtual cloud asset including, but not limited to, a software container, a micro-service, a serverless function, and the like. For the sake of the discussion and without limiting the scope of the disclosed embodiments, VM-**119-1** is an active machine and VM **119-2** is a machine at rest. That is, VM **119-2** is mostly in an inactive state (e.g., being execute a day in a month, a month in a year, and remains inactive otherwise.

The storage **117** emulates virtual discs for the VMs **119-1** and **119-2** executed in by the server **115**. The storage **117** is typically connected to the server **115** through a high-speed connection, such as optical fiber, allowing fast retrieval of data. In other configurations, the storage **117** may be part of the server **115**. In this example, illustrated in FIG. **1**B, a virtual disk **118-1** is allocated for the VM **119-1** and the virtual disk **118-2** is allocated for the VM **119-2**. The server **115**, and, hence, the VMs **119-1** and **119-2**, may be executed in a client environment **130** within the platform **110**.

The client environment **130** is an environment within the cloud computing platform **110** utilized to execute cloud-hosted applications of the client. A client may belong to a specific tenant. In some example embodiments, the client environment **130** may be part of a virtualized environment or on-premises virtualization environment, such as a VMware® based solution.

Also deployed in the cloud computing platform **110** is a security system **140** configured to perform the various disclosed embodiments. In some embodiments, the system **140** may be part of the client environment **130**. In an embodiment, the security system **140** may be realized as a physical machine configured to execute a plurality of virtual instances, such as, but not limited to virtual machines executed by a host server. In yet another embodiment, the security system **140** may be realized as a virtual machine executed by a host server. Such a host server is a physical machine (device) and may be either the server **115**, a dedicated server, a different shared server, or another virtualization-based computing entity, such as a serverless function.

In an embodiment, the interface between the client environment **130** and the security system **140** can be realized using APIs or services provided by the cloud computing platform **110**. For example, in AWS, a cross account policy service can be utilized to allow interfacing the client environment **130** with the security system **140**.

In the deployment, illustrated in FIGS. **1**A and **1**B, the configuration of resources of the cloud computing platform **110** is performed by means of the management console **150**. As such, the management console **150** may be queried on the current deployment and settings of resources in the cloud computing platform **110**. Specifically, the management console **150** may be queried, by the security system **140**, about the location (e.g., virtual address) of the virtual disk **118-1** in the storage **117**. The system **140** is configured to interface with the management console **150** through, for example, an API.

In some example embodiments, the security system **140** may further interface with the cloud computing platform **110** and external systems **170**. The external systems may include intelligence systems, security information and event management (STEM) systems, and mitigation tools. The external intelligence systems may include common vulnerabilities and exposures (CVE®) databases, reputation services, security systems (providing feeds on discovered threats), and so on. The information provided by the intelligence systems may detect certain known vulnerabilities identified in, for example, a CVE database.

In an embodiment, the security system **140** is configured to detect vulnerabilities and other cyber threats related to the execution VM **119-1**. The detection is performed while the VM **119-1** is live, without using any agent installed in the server **115** or the VM **119-1**, and without relying on cooperation from the guest OS of the VM **119-1**.

According to another embodiment, the security system **140** is configured to detect vulnerabilities and other cyber threats related to the execution VM **119-2**. i.e., the machine at rest. The detection is performed while the VM **119-2** is powered off.

In both embodiments, the security system **140** can scan and detect vulnerable software, non-secure configurations, exploitation attempts, compromised assets, data leaks, data mining, and so on. The security system **140** may be further utilized to provide security services, such as incident response, anti-ransomware, and cyber-insurance, by accessing the security posture.

In some embodiments, the security system **140** is configured to query the cloud management console **150** for the address of the virtual disks **118-1** and **118-2**, respectively serving the VM **119-1**, VM **119-2**, and a location of the snapshot. A VM's snapshot is a copy of the machine's virtual disk (or disk file) at a given point in time. Snapshots provide a change log for the virtual disk and are used to restore a VM to a particular point in time when a failure error occurs. Typically, any data that was writable on a VM becomes read-only when the snapshot is taken. Multiple snapshots of a VM can be created at multiple possible point-in-time restore points. When a VM reverts to a snapshot, current disk and memory states are deleted and the snapshot becomes the new parent snapshot for that VM.

In an embodiment, a view, or a materialized view, of the virtual disk **118-2** associated with the VM **119-2** is accessed. A view is a stored query that consumes limited-to-no space, consuming only the space required to store the text of the query in the data dictionary. A materialized view is a both a stored query and a segment. That is, a stored query is executed, and the results are materialized into the segment. For the sake of simplicity, but without limiting the scope of the disclosed embodiments, the inspection of VM (VM **119-2**) is based on a view stored in the virtual disk **118-2**, while the inspection of the active (VM **119-1**) is based on a snapshot stored in the virtual disk **118-1**.

The snapshot of the VM **119-1** is located and may be saved from the virtual disk **118-1** for access by the security system **140**. In an embodiment, the VM's **119-1** snapshot may be copied to the system **140**. If such a snapshot does not exist, the system **140** may take a new snapshot or request such an action. The snapshots may be taken on a predefined schedule or upon predefined events (e.g., a network event or abnormal event). Further, the snapshots may be accessed or copied on a predefined schedule or upon predefined events. It should be noted that when the snapshot is taken or copied, the VM **119** still runs.

The view of the VM **119-2** is located and may be saved from the virtual disk **118-2** for access by the system **140**. In an embodiment, the VM's **119-2** view may be copied to the system **140**. If such a view does not exist, the system **140** may generate a query to create a new VM **119-2**. The view may be taken when the VM **119-2** is about to transition into an inactive state or when the same VM **119-2** is at rest. It should be noted that when the view is taken or copied, the VM **119-2** may be at rest (i.e., inactive and powered off).

It should be noted that the snapshots and/or views of the virtual disk **118-1** and/or **118-2** may not necessarily be stored in the storage **117**, but, for ease of discussion, it is assumed that the snapshot is saved in the storage **117**. It should be further noted that the snapshots and/or views are accessed without cooperation of the guest, virtual OS of the virtual machine.

The snapshot is parsed and analyzed by the security system **140** to detect vulnerabilities. This analysis of the snapshot does not require any interaction and/or information from the VM **119-1**. As further demonstrated herein, the analysis of the snapshot by the system **140** does not require any agent installed on the server **115** or VM **119-1**.

Further, the view is parsed and analyzed by the security system **140** to detect vulnerabilities. This analysis of the views does not require any interaction and/or information from the VM **119-2**. In fact, the VM **119-2** is in its inactive state (at rest) during the analysis. As further demonstrated herein, the analysis of the view by the system **140** does not require any agent installed on the server **115** or VM **119-2**.

Various techniques can be utilized to analyze the views and snapshots, depending on the type of vulnerability and cyber threats to be detected. Following are some example embodiments for techniques that may be implemented by the security system **140**.

In an embodiment, the security system **140** is configured to detect whether there is vulnerable code executed by the VMs **119-1** and **119-2**. In an embodiment, the VM **119-2** being analyzed is shut down, being, therefore, at rest. The VM **119-1** may be running or paused. In an embodiment, to detect vulnerabilities existing in the VM **119-2**, the security system **140** is configured to match installed application lists, with their respective versions, to a known list of vulnerable applications. Further, the security system **140** may be configured to match the application files, either directly, using binary comparison, or by computing a cryptographic hash against database of files in vulnerable applications. The matching may be also on sub-modules of an application. Alternatively, the security system **140** may read installation logs of package managers used to install the packages of the application.

In yet another embodiment, the security system **140** is configured to verify whether the vulnerability is relevant to the VM **119-2**. For example, if there is a vulnerable version or module not in use, the priority of that issue is reduced dramatically.

To this end, the security system **140** may be configured to check the configuration files of the applications and operating system of the VM **119-2** to verify access times to files by the operating system and/or to analyze the application and/or system logs in order to deduce what applications and modules are running.

In yet another embodiment, the security system **140** may instantiate a copy of the VM **119-2** and/or a subset of applications of the VM **119-2** on the server **115** or a separate server and monitor all activity performed by the instance of the VM. The execution of the instance of the VM is an isolated sandbox, which can be a full VM or subset of it, such as a software container (e.g., Docker® container) or another virtualized instance. The monitored activity may be further analyzed to determine abnormality. Such analysis may include monitoring of API activity, process creation, file activity, network communication, registry changes, and active probing of said subset in order to assess its security posture. This may include, but is not limited to, actively communicating with the VM **119-2** and using either legitimate communication and/or attack attempts to assess posture and, by that, deriving the security posture of the entire VM **119-2**.

In order to determine if the vulnerability is relevant to the VM **119-2**, the security system **140** is configured to analyze the machine memory, as reflected in the page file. The page file is saved in the snapshot and extends how much system-committed memory (also known as "virtual memory") a system can back. In an embodiment, analyzing the page file allows deduction of running applications and modules by the VM **119-2**. It should be noted that analyzing pages would be available only when VM **119-2** hibernates.

In yet another embodiment, the security system **140** is configured to detect cyber threats that do not represent vulnerabilities. For example, the security system **140** may detect and alert on sensitive data not being encrypted on the logical disk, private keys found on the disks, system credentials stored clearly on the disk, risky application features (e.g., support of weak cipher suites or authentication methods), weak passwords, weak encryption schemes, a disabled address space layout randomization (ASLR) feature, suspicious manipulation to a boot record, suspicious PATH, LD_LIBRARY_PATH, or LD_PRELOAD definitions, services running on startup, and the like.

In an embodiment, the security system **140** may further monitor changes in sensitive machine areas, and alert on unexpected changes, such as added or changed application files without installation. In an example embodiment, this can be achieved by computing a cryptographic hash of the sensitive areas in the virtual disk and checking for differences over time.

In some embodiments, the detected cyber threats (including vulnerabilities) are reported to a user console **180** and/or a security information and event management (SEM) system (not shown). The reported cyber threats may be filtered or prioritized based, in part, on their determined risk. Further, the reported cyber threats may be filtered or prioritized based, in part, on the risk level of the machine. This also reduces the number of alerts reported to the user.

In an embodiment, any detected cyber threats related to sensitive data, including personally identifiable information, or PII, is reported at a higher priority. In an embodiment, such data is determined by searching for the PII, analyzing the application logs to determine whether the machine accessed PH/PH-containing servers, or whether the logs themselves contain PII, and searching the machine memory, as reflected in the page file, for PII.

In an embodiment, the security system **140** may determine the risk of the VM **119** based on communication with an untrusted network. This can be achieved by analyzing the VM's **119-2** logs as saved in the virtual disk, and can be derived from the view.

In an example embodiment, the security system **140** may cause an execution of one or more mitigation actions. Examples for such actions may include disabling the VM **119-2** from execution, updating the VM **119-2** with recent patches, and so on.

The above examples for detecting vulnerabilities may be applicable also for a VM **119-1** and may be performed when the VM **119-1** will be started later on. For the active VM **119-1** the mitigation actions may include blocking traffic from untrusted networks, halting the operation of the VM, quarantining an infected VM, and the like. The mitigation actions may be performed by a mitigation tool and not the system **140**.

It should be noted that the example implementation shown in FIG. **1** is described with respect to a single cloud computing platform **110** hosting two VMs **119-1** and **119-2** in a single server **115**, merely for simplicity purposes and without limitation on the disclosed embodiments. Typically, virtual machines are deployed and executed in a single cloud computing platform, a virtualized environment, or data center and can be protected without departing from the scope of the disclosure. It should be further noted that the disclosed embodiments can operate using multiple security systems **140**, each of which may operate in a different client environment.

FIG. **2** shows an example flowchart **200** illustrating a method for detecting cyber threats including potential vulnerabilities in virtual machines at rest, according to some embodiments. The method may be performed by the security system **140**.

At S**210**, a request, for example, to scan a VM, at rest, for vulnerabilities, is received. A VM at rest is a machine that is currently powered off, i.e., not in an operational state. A VM at rest is executed at predefined time period but remains inactive (powered off) when not executed. The request, received at S**210**, may be received, or otherwise triggered, at every predefined time interval or upon detection of an external event. An external event may be a preconfigured event, such as a network event or abnormal event including, without limitation, requests to run the VM **119-2** not according to a schedule, access by an authorized user, and the like. The request may at least designate an identifier of the VM to be scanned.

At S**220**, a location of a view of the VM, at rest, to be scanned is determined. In an embodiment, S**220** may include determining the virtual disk allocated for the VM, prior to determining the location of the view. As noted above, this can be achieved by querying a cloud management console. In an embodiment, a snapshot of the VM, at rest, is accessed. At S**230**, a snapshot of the virtual disk is accessed, or otherwise copied.

At S**240**, the view is analyzed to detect cyber threats and potential vulnerabilities. S**240** may also include detecting cyber threats that do not represent vulnerabilities, Examples for cyber threats and vulnerabilities are provided above.

In an embodiment, S**240** may include comparing the view to some baseline, which may include, but is not limited to, a copy of the image used to create the VM, (e.g., lists of applications, previous snapshots), cryptographic hashes gathered in the previous scan, analyzing logs of the VMs, instantiating a copy of the VM and executing the instance or applications executed by the VM in a sandbox, analyzing the machine memory, as reflected in the page file, or any combination of these techniques. Some example embodiments for analyzing the snapshots and the types of detected vulnerabilities and threats are provided above.

At S**250**, the detected cyber threats and/or vulnerabilities are reported, for example, as alerts. In an embodiment, S**250** may include filtering and prioritizing the reported alerts. In an embodiment, the prioritization is based, in part, on the risk level of a vulnerable machine. The filtering and prioritizing allow for reduction of the number of alerts reported to the user. The filtering can be performed on external intelligence on the likelihood of this vulnerability being exploited, analyzing the machine configuration in order to deduce the vulnerability relevancy, and correlating the vulnerability with the network location and by weighting the risk of this machine being taken over by the attacker by taking into consideration the criticality of the machine in the organization based on the contents stored or on other assets accessible from the VM.

At optional S**260**, a mitigation action may be triggered to mitigate a detected threat or vulnerability. A mitigation action may be executed by a mitigation tool and triggered by the system **140**. Such an action may include blocking traffic from untrusted networks, halting the operation of the VM, quarantining an infected VM, preventing its launch and the like.

FIG. **3** is an example block diagram of the security system **140** according to an embodiment. The security system **140** includes a processing circuitry **310** coupled to a memory **320**, a storage **330**, and a network interface **340**. In an embodiment, the components of the security system **140** may be communicatively connected via a bus **360**.

The processing circuitry **310** may be realized as one or more hardware logic components and circuits. For example, and without limitation, illustrative types of hardware logic components that can be used include field programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), application-specific standard products (ASSPs), system-on-a-chip systems (SOCs), general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), and the like, or any other hardware logic components that can perform calculations or other manipulations of information.

The memory **320** may be volatile (e.g., RAM, etc.), non-volatile (e.g., ROM, flash memory, etc.), or a combination thereof. In one configuration, computer readable instructions to implement one or more embodiments disclosed herein may be stored in the storage **330**.

In another embodiment, the memory **320** is configured to store software. Software shall be construed broadly to mean 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, cause the processing circuitry **310** to perform the various processes described herein. Specifically, the instructions, when executed, cause the processing circuitry **310** to determine over-privileged role vulnerabilities in serverless functions.

The storage **330** may be magnetic storage, optical storage, and the like, and may be realized, for example, as flash memory or as another other memory technology, as CD-ROMs, Digital Versatile Disks (DVDs), hard-drives, SSDs, or any other medium which can be used to store the desired information. The storage **330** may store communication consumption patterns associated with one or more communications devices.

The network interface **340** allows the security system **140** to communicate with the external systems, such as intelligence systems, SIEM systems, mitigation systems, a cloud management console, a user console, and the like.

It should be understood that the embodiments described herein are not limited to the specific architecture illustrated in FIG. **3**, and other architectures may be equally used without departing from the scope of the disclosed embodiments.

The various embodiments disclosed herein can be implemented as hardware, firmware, software, or any combination thereof. Moreover, the software is preferably 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 comprising any suitable architecture. Preferably, the machine is 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 herein 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 is any computer readable medium except for a transitory propagating signal.

As used herein, the phrase "at least one of" followed by a listing of items means that any of the listed items can be utilized individually, or any combination of two or more of the listed items can be utilized. For example, if a system is described as including "at least one of A, B, and C," the system can include A alone; B alone; C alone; A and B in combination; B and C in combination; A and C in combination; or A, B, and C in combination.

All examples and conditional language recited herein are intended for pedagogical purposes to aid the reader in understanding the principles of the disclosed embodiment and the concepts contributed by the inventor to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the disclosed embodiments, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

What is claimed is:

1. A system for inspecting data, the system comprising: at least one processor configured to:

establish an interface between a client environment and security components;

using the interface, utilize cloud computing platform APIs to identify virtual disks of a virtual machine in the client environment;

use the computing platform APIs to query a location of at least one of the identified virtual disks;

receive an identification of the location of the virtual disks of the virtual machine;

perform at least one of: (i) taking at least one snapshot, and (ii) requesting taking at least one snapshot of the virtual machine at rest, wherein the at least one snapshot represents a copy of the virtual disks of the virtual machine at a point in time;

analyze the at least one snapshot to detect vulnerabilities, wherein during the detection of the vulnerabilities by analyzing the at least one snapshot, the virtual machine is inactive; and

report the detected vulnerabilities as alerts.

2. The system of claim **1**, wherein reporting the detected vulnerabilities as alerts includes indicating priority levels associated with the detected vulnerabilities.

3. The system of claim **1**, wherein the at least one processor is further configured to implement a remedial action for at least one of the detected vulnerabilities.

4. The system of claim **1**, wherein the identification of the location of the virtual disks of the virtual machine includes a virtual address of at least one of the virtual disks.

5. The system of claim **1**, wherein the at least one snapshot includes a change log of at least one of the virtual disks configured to restore the virtual machine to a particular point in time.

6. The system of claim **1**, wherein the snapshot includes a page file of memory associated with the virtual machine, the page file being configured to allow deduction of one or more applications running on the virtual machine.

7. The system of claim **1**, wherein the at least one snapshot includes a plurality of snapshots, and the at least one processor is configured to generate the plurality of snapshots according to a predetermined schedule.

8. The system of claim **1**, wherein the at least one processor is configured to generate the at least one snapshot in response to a predetermined trigger event.

9. A computer-implemented method for inspecting data, the method comprising:

establishing an interface between a client environment and security components;

using the interface to utilize cloud computing platform APIs to identify virtual disks of a virtual machine in the client environment;

using the computing platform APIs to query a location of at least one of the identified virtual disks;

receiving an identification of the location of the virtual disks of the virtual machine;

emulating the virtual disks for the virtual machine;

performing at least one of: (i) taking at least one snapshot, and (ii) requesting taking at least one snapshot of the virtual machine at rest, wherein the at least one snapshot represents a copy of the virtual disks of the virtual machine at a point in time;

analyzing the at least one snapshot to detect vulnerabilities, wherein during the detection of the vulnerabilities by analyzing the at least one snapshot, the virtual machine is inactive; and

reporting the detected vulnerabilities as alerts.

10. The method of claim **9**, wherein reporting the detected vulnerabilities as alerts includes providing indications of the detected vulnerabilities based on priority levels associated with the detected vulnerabilities.

11. The method of claim **9**, further comprising implementing a remedial action for at least one of the detected vulnerabilities.

12. The method of claim **9**, wherein the identification of the location of the virtual disks of the virtual machine includes a virtual address of at least one of the virtual disks.

13. The method of claim **9**, wherein the at least one snapshot includes a change log of at least one of the virtual disks configured to restore the virtual machine to a particular point in time.

14. The method of claim **9**, wherein the snapshot includes a page file of memory associated with the virtual machine, the page file being configured to allow deduction of one or more applications running on the virtual machine.

15. The method of claim **9**, wherein the at least one snapshot includes a plurality of snapshots and wherein the method further includes generating the plurality of snapshots based on a predetermined schedule.

16. A non-transitory computer-readable medium storing instructions, which, when executed by at least one processor, cause a computing device to:

establish an interface between a client environment and security components;

using the interface, utilize cloud computing platform APIs to identify virtual disks of a virtual machine in the client environment;

use the computing platform APIs to query a location of at least one of the identified virtual disks;

receive an identification of the location of the virtual disks of the virtual machine;

emulate the virtual disks for the virtual machine;

perform at least one of: (i) taking at least one snapshot, and (ii) requesting taking at least one snapshot of the virtual machine at rest, wherein the at least one snapshot represents a copy of the virtual disks of the virtual machine at a point in time;

analyze the at least one snapshot to detect vulnerabilities, wherein during the detection of the vulnerabilities by analyzing the at least one snapshot, the virtual machine is inactive; and

report the detected vulnerabilities as alerts.

* * * * *

EXHIBIT 2

US011663032B2

(12) **United States Patent**
Shua

(10) **Patent No.: US 11,663,032 B2**
(45) **Date of Patent: *May 30, 2023**

(54) **TECHNIQUES FOR SECURING VIRTUAL MACHINES BY APPLICATION USE ANALYSIS**

(71) Applicant: **Orca Security Ltd.**, Tel Aviv (IL)

(72) Inventor: **Avi Shua**, Tel Aviv (IL)

(73) Assignee: **Orca Security Ltd.**, Tei Aviv (IL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **18/055,150**

(22) Filed: **Nov. 14, 2022**

(65) **Prior Publication Data**

US 2023/0089313 A1 Mar. 23, 2023

Related U.S. Application Data

(63) Continuation of application No. 17/330,998, filed on May 26, 2021, now Pat. No. 11,516,231, which is a continuation of application No. 16/585,967, filed on Sep. 27, 2019, now Pat. No. 11,431,735.

(60) Provisional application No. 62/797,718, filed on Jan. 28, 2019.

(51) **Int. Cl.**

H04L 9/40	(2022.01)
G06F 9/455	(2018.01)
G06F 16/11	(2019.01)
G06F 11/14	(2006.01)

(52) **U.S. Cl.**

CPC ***H04L 63/1416*** (2013.01); ***G06F 9/45558*** (2013.01); ***G06F 11/1464*** (2013.01); ***G06F 16/128*** (2019.01); ***H04L 63/1433*** (2013.01); ***H04L 63/1441*** (2013.01); *G06F 2009/45562* (2013.01); *G06F 2009/45583* (2013.01); *G06F 2009/45587* (2013.01);

(Continued)

(58) **Field of Classification Search**

None

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

9,069,983	B1	6/2015	Nijjar
9,177,145	B2	11/2015	Todorović
9,229,758	B2	1/2016	Ammons et al.

(Continued)

OTHER PUBLICATIONS

Rani et al.; "An Efficient Approach to Forensic Investigation in Cloud using VM Snapshots", 2015 International Conference on Pervasive Computing (ICPC), 5 pages, (2015).

(Continued)

Primary Examiner — Joseph P Hirl

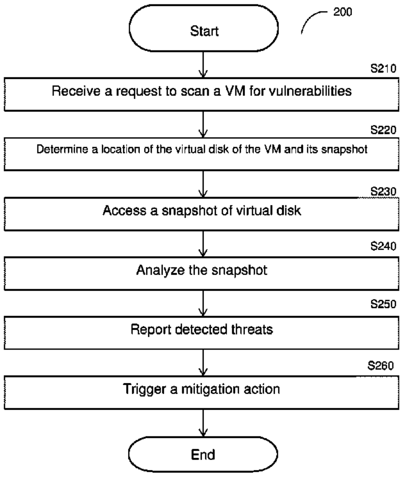
Assistant Examiner — Hassan Saadoun

(74) *Attorney, Agent, or Firm* — Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.

(57) **ABSTRACT**

A system and method for securing virtual cloud assets in a cloud computing environment against cyber threats. The method includes: determining a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is instantiated in the cloud computing environment; accessing the snapshot of the virtual disk based on the determined location; analyzing the snapshot of the protected virtual cloud asset to detect

(Continued)


200
Start
S210
Receive a request to scan a VM for vulnerabilities
S220
Determine a location of the virtual disk of the VM and its snapshot
S230
Access a snapshot of virtual disk
S240
Analyze the snapshot
S250
Report detected threats
S260
Trigger a mitigation action
End

potential cyber threats risking the protected virtual cloud asset; and alerting detected potential cyber threats based on a determined priority.

25 Claims, 4 Drawing Sheets

(52) **U.S. Cl.**
CPC *G06F 2009/45591* (2013.01); *G06F 2009/45595* (2013.01); *G06F 2201/84* (2013.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

9,268,689	B1	2/2016	Chen et al.	
9,519,781	B2	12/2016	Golshan et al.	
9,563,777	B2	2/2017	Deng et al.	
9,734,325	B1	8/2017	Neumann et al.	
9,756,070	B1	9/2017	Crowell et al.	
9,798,885	B2	10/2017	Deng et al.	
9,858,105	B1	1/2018	Upadhyay et al.	
10,079,842	B1	9/2018	Brandwine et al.	
10,402,560	B2	9/2019	Gilbert	
10,412,109	B2	9/2019	Loureiro et al.	
10,469,304	B1	11/2019	Kempe et al.	
10,534,915	B2	1/2020	Cherny et al.	
10,536,471	B1 *	1/2020	Derbeko	G06F 21/53
10,782,952	B1	9/2020	Doring et al.	
10,944,778	B1 *	3/2021	Golan	H04L 63/1491
11,068,353	B1 *	7/2021	Ved	G06F 9/45558
11,120,124	B2	9/2021	Fusenig et al.	
11,216,563	B1	1/2022	Veselov et al.	
11,431,735	B2	8/2022	Shua	
11,516,231	B2	11/2022	Shua	
2007/0266433	A1	11/2007	Moore	
2008/0189788	A1	8/2008	Bahl	
2008/0263658	A1	10/2008	Michael et al.	
2009/0007100	A1	1/2009	Field et al.	
2010/0017512	A1	1/2010	Ciano et al.	
2011/0289584	A1	11/2011	Palagummi	
2012/0323853	A1	12/2012	Fries et al.	
2013/0191643	A1 *	7/2013	Song	H04L 9/3265 713/176
2014/0096135	A1	4/2014	Kundu et al.	
2014/0137190	A1	5/2014	Carey et al.	
2015/0052520	A1	2/2015	Crowell et al.	
2016/0004449	A1 *	1/2016	Lakshman	G06F 3/0604 711/162
2016/0094568	A1 *	3/2016	Balasubramanian	G06F 9/45558 726/23
2016/0364255	A1	12/2016	Chefalas et al.	
2017/0011138	A1	1/2017	Venkatesh et al.	
2017/0031704	A1	2/2017	Sudhakaran et al.	
2017/0103212	A1 *	4/2017	Deng	G06F 3/0619
2017/0111384	A1	4/2017	Loureiro et al.	
2018/0137032	A1	5/2018	Tannous et al.	
2018/0255080	A1	9/2018	Paine	
2018/0293374	A1	10/2018	Chen	
2020/0042707	A1	2/2020	Kucherov et al.	

OTHER PUBLICATIONS

Wei et al., "Managing Security of Virtual Machine Images in a Cloud Environment", CCSW'09, pp. 91-96, Nov. 13, 2009.
Almulla et al., "Digital Forensic of a Cloud Based Snapshot", The Sixth International Conference on Innovative Computing Technology (INTECH 2016), pp. 724-729, (2016).
Pandey et al., "An Approach for Virtual Machine Image Security", Computer Science and Engineering MNNIT, Allahabad, 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT), pp. 616-623, (2014).
Rajasekaran et al., "Scalable Cloud Security via Asynchronous Virtual Machine Introspection", 8th USENIX Workshop on Hot Topics in Cloud Computing, Cover page and pp. 1-6, (2016).
Kaur et al., "Secure VM Backup and Vulnerability Removal in Infrastructure Clouds", 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1217-1226, (2014).
Fernandez et al., "Two patterns for cloud computing: Secure Virtual Machine Image Repository and Cloud Policy Management Point", PLoP '13: Proceedings of the 20th Conference on Pattern Languages of Programs Oct. 2013 Article No. 15, Association for Computing Machinery (ACM), Cover sheet and pp. 1-11, (2013).
Fernandez et al. "Building a security reference architecture for cloud systems", Springer, Requirements Eng, vol. 21, pp. 225-249, (2016).
Ammons et al., "Virtual machine images as structured data: the Mirage image library", IBM Research, Cover sheet 2 pages and pp. 1-6, (2011).
"IBM Point of View Security and Cloud Computing", IBM SmartCloud Enterprise, Cloud Computing White Paper, 13 sheets of cover pages and pp. 1-20, (2009).
Cui et al., "A Less Resource-Consumed Security Architecture on Cloud Platform", Wuhan University Journal of Natural Sciences, vol. 21, No. 5, pp. 407-414, (2016).
Bugiel et al., "AmazonIA: When Elasticity Snaps Back", CCS' 11, ACM, pp. 389-400, (2011).
Non-Final Office Action U.S. Appl. No. 16/585,967 dated Feb. 3, 2022, in the United States Patent and Trademark Office.
Notice of Allowance U.S. Appl. No. 16/585,967 dated Jul. 7, 2022, in the United States Patent and Trademark Office.
Non-Final Office Action U.S. Appl. No. 17/330,998 dated Mar. 4, 2022, in the United States Patent and Trademark Office.
Final Office Action U.S. Appl. No. 17/330,998 dated Jun. 30, 2022, in the United States Patent and Trademark Office.
Notice of Allowance U.S. Appl. No. 17/330,998 dated Aug. 10, 2022, in the United States Patent and Trademark Office.
Non-Final Office Action U.S. Appl. No. 17/361,861 dated Aug. 29, 2022, in the United States Patent and Trademark Office.
Advisory Action U.S. Appl. No. 17/361,861 dated May 20, 2022, in the United States Patent and Trademark Office.
Final Office Action U.S. Appl. No. 17/361,861 dated Mar. 8, 2022, in the United States Patent and Trademark Office.
Non-Final Office Action U.S. Appl. No. 17/361,861 dated Oct. 25, 2021, in the United States Patent and Trademark Office.

* cited by examiner

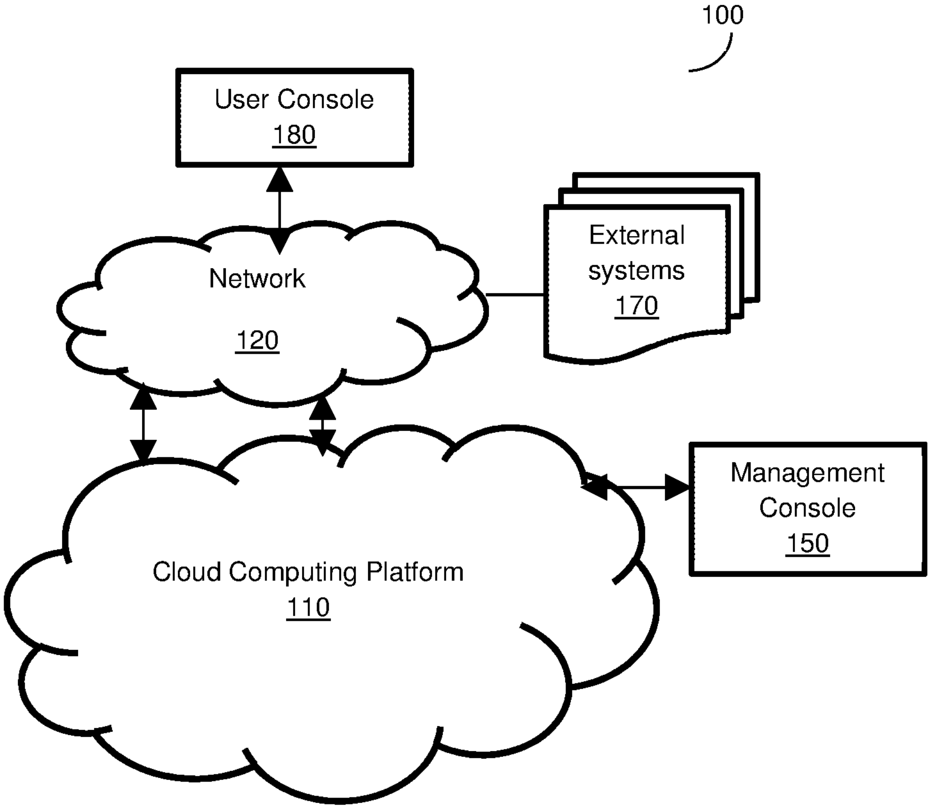

100
User Console
180
Network
120
External
systems
170
Cloud Computing Platform
110
Management
Console
150

FIG. 1A

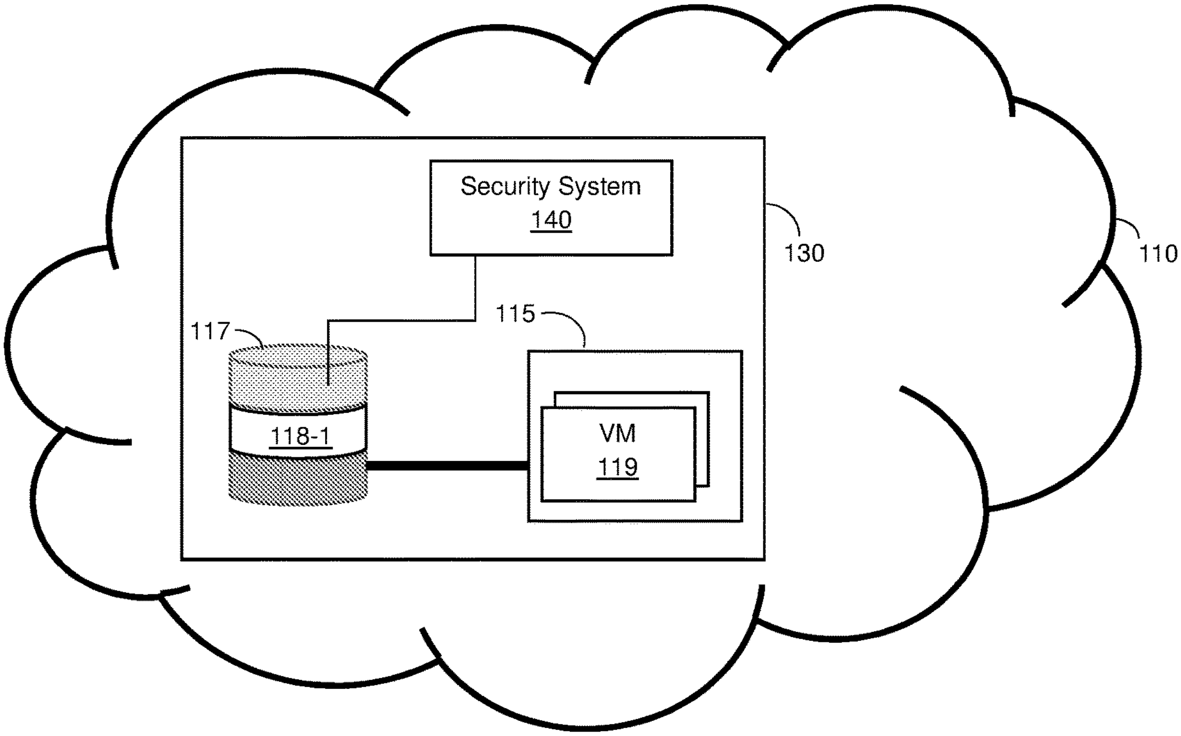


FIG. 1B

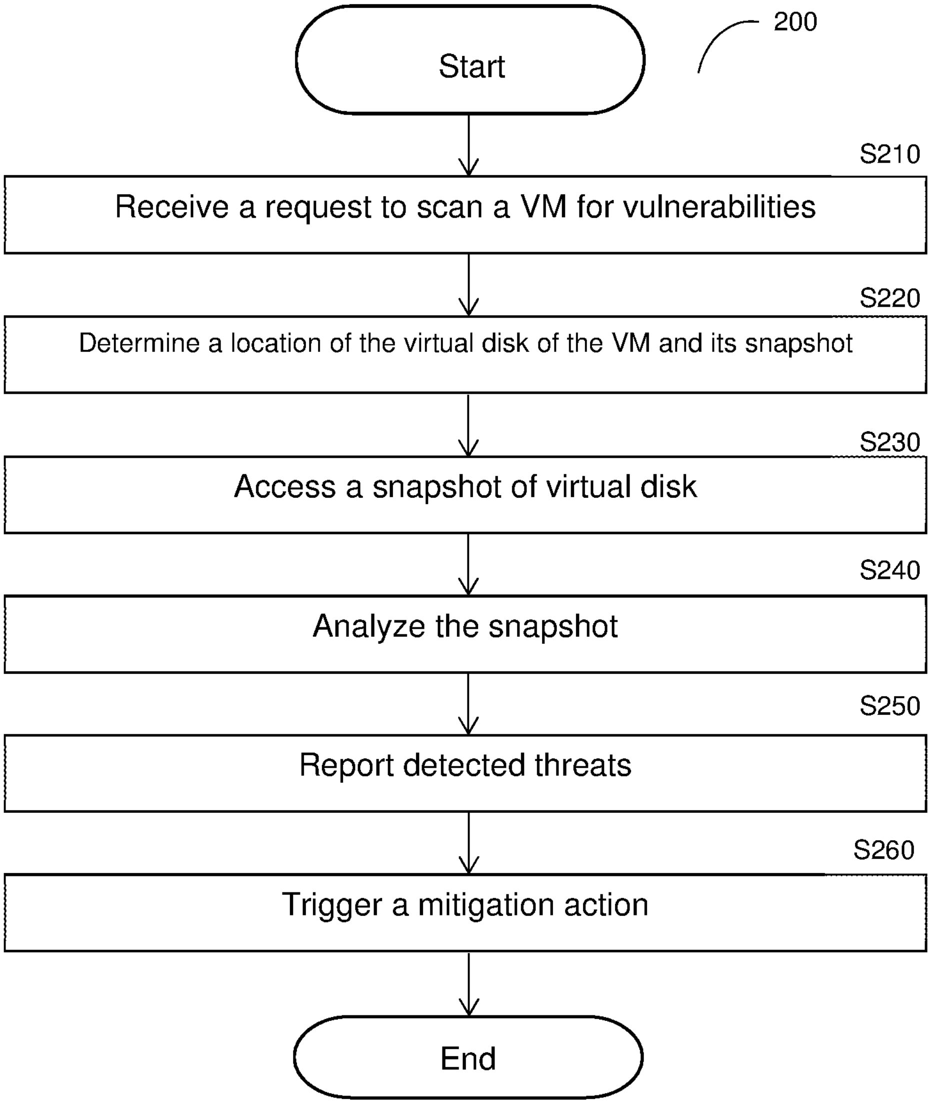

200
Start
S210
Receive a request to scan a VM for vulnerabilities
S220
Determine a location of the virtual disk of the VM and its snapshot
S230
Access a snapshot of virtual disk
S240
Analyze the snapshot
S250
Report detected threats
S260
Trigger a mitigation action
End

FIG. 2

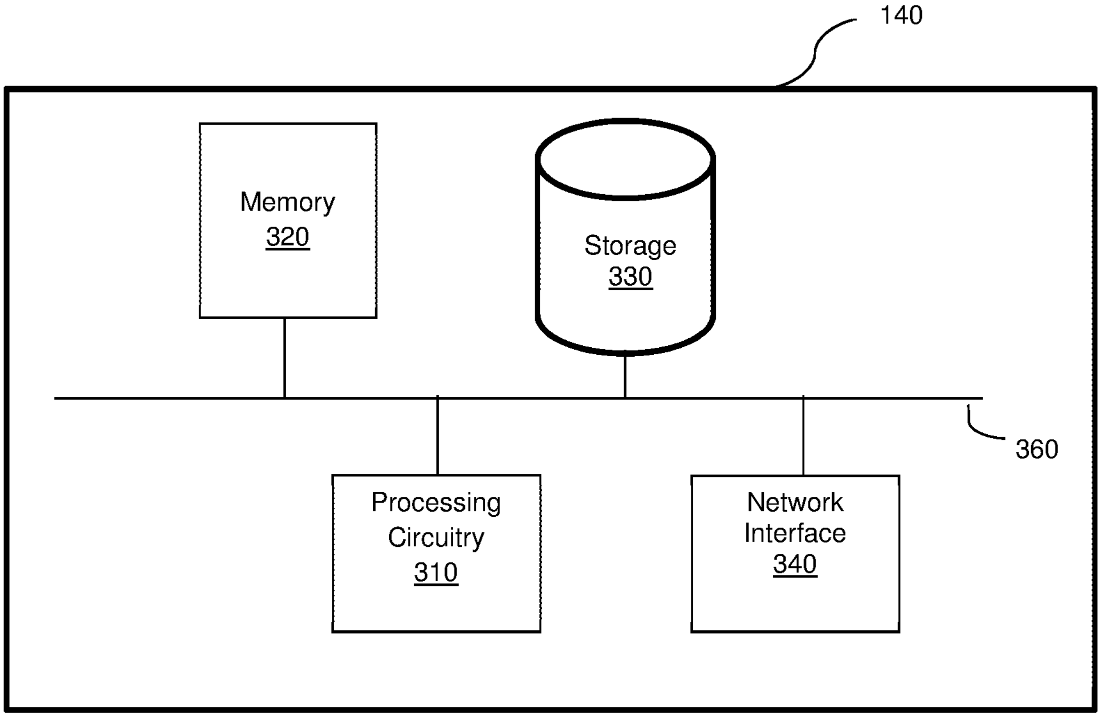

140
Memory
320
Storage
330
360
Processing
Circuitry
310
Network
Interface
340

FIG. 3

TECHNIQUES FOR SECURING VIRTUAL MACHINES BY APPLICATION USE ANALYSIS

This application is a continuation of U.S. application Ser. No. 17/330,998 (now allowed), filed May 26, 2021, which is a continuation of U.S. application Ser. No. 16/585,967 (now U.S. Pat. No. 11,431,735), filed Sep. 27, 2019, which claims the benefit of U.S. Provisional Application No. 62/797,718 filed on Jan. 28, 2019, the contents of each of which are hereby incorporated by reference in their entireties.

TECHNICAL FIELD

This disclosure relates generally to cyber-security systems and, more specifically, to techniques for securing virtual machines.

BACKGROUND

Organizations have increasingly adapted their applications to be run from multiple cloud computing platforms. Some leading public cloud service providers include Amazon®, Microsoft®, Google®, and the like.

Virtualization is a key role in a cloud computing, allowing multiple applications and users to share the same cloud computing infrastructure. For example, a cloud storage service can maintain data of multiple different users.

In one instance, virtualization can be achieved by means of virtual machines. A virtual machine emulates a number of "computers" or instances, all within a single physical device. In more detail, virtual machines provide the ability to emulate a separate operating system (OS), also referred to as a guest OS, and therefore a separate computer, from an existing OS (the host). This independent instance is typically isolated as a completely standalone environment.

Modern virtualization technologies are also adapted by cloud computing platforms. Examples for such technologies include virtual machines, software containers, and serverless functions. With their computing advantages, applications and virtual machines running on top of virtualization technologies are also vulnerable to some cyber threats. For example, virtual machines can execute vulnerable software applications or infected operating systems.

Protection of a cloud computing infrastructure, and particularly of virtual machines can be achieved via inspection of traffic. Traditionally, traffic inspection is performed by a network device connected between a client and a server (deployed in a cloud computing platform or a data center) hosting virtual machines. Traffic inspection may not provide an accurate indication of the security status of the server due to inherent limitations, such as encryption and whether the necessary data is exposed in the communication.

Furthermore, inspection of computing infrastructure may be performed by a network scanner deployed out of path. The scanner queries the server to determine if the server executes an application that possess a security threat, such as vulnerability in the application. The disadvantage of such a scanner is that the server may not respond to all queries by the scanner, or not expose the necessary data in the response. Further, the network scanner usually communicates with the server, and the network configuration may prevent it. In addition, some types of queries may require credentials to access the server. Such credentials may not be available to the scanner.

Traffic inspection may also be performed by a traffic monitor that listens to traffic flows between clients and the server. The traffic monitor can detect some cyber threats, e.g., based on the volume of traffic. However, the monitor can detect threats only based on the monitored traffic. For example, misconfiguration of the server may not be detected by the traffic monitor. As such, traffic monitoring would not allow detection of vulnerabilities in software executed by the server.

To overcome the limitations of traffic inspection solutions, some cyber-security solutions, such as vulnerability management and security assessment solutions are based on agents installed in each server in a cloud computing platform or data center. Using agents is a cumbersome solution for a number of reasons, including IT resources management, governance, and performance. For example, installing agents in a large data center may take months.

It would therefore be advantageous to provide a security solution that would overcome the deficiencies noted above.

SUMMARY

A summary of several example embodiments of the disclosure follows. This summary is provided for the convenience of the reader to provide a basic understanding of such 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 concepts of one or more embodiments in a simplified form as a prelude to the more detailed description that is presented later. For convenience, the term "some embodiments" or "certain embodiments" may be used herein to refer to a single embodiment or multiple embodiments of the disclosure.

Certain embodiments disclosed herein include a method for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising: determining a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is instantiated in the cloud computing environment; accessing the snapshot of the virtual disk based on the determined location; analyzing the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset; and alerting detected potential cyber threats based on a determined priority.

Certain embodiments disclosed herein also include a non-transitory computer readable medium having stored thereon instructions for causing a processing circuitry to execute a process, the process comprising: determining a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is instantiated in the cloud computing environment; accessing the snapshot of the virtual disk based on the determined location; analyzing the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset; and alerting detected potential cyber threats based on a determined priority.

Certain embodiments disclosed herein also include a system for securing virtual cloud assets in a cloud computing environment against cyber threats, comprising: a processing circuitry; and a memory, the memory containing instructions that, when executed by the processing circuitry, configure the system to: determine a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the virtual cloud asset is instantiated in the cloud computing environment; access the snapshot of the virtual disk based on the determined location; analyze the snapshot of the protected virtual cloud asset to detect potential cyber threats risking the protected virtual cloud asset; and alert detected potential cyber threats based on a determined priority.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features, and advantages of the disclosed embodiments will be apparent from the following detailed description taken in conjunction with the accompanying drawings.

FIGS. **1**A and **1**B are network diagrams utilized to describe the various embodiments.

FIG. **2** is a flowchart illustrating a method detecting cyber threats, including potential vulnerabilities in virtual machines executed in a cloud computing platform according to some embodiments.

FIG. **3** is an example block diagram of the security system according to an embodiment.

DETAILED DESCRIPTION

It is important to note that the embodiments disclosed herein are only examples of the many advantageous uses of the innovative teachings herein. In general, statements made in the specification of the present application do not necessarily limit any of the various claimed embodiments. Moreover, some statements may apply to some inventive features but not to others. In general, unless otherwise indicated, singular elements may be in plural and vice versa with no loss of generality. In the drawings, like numerals refer to like parts through several views.

FIGS. **1**A and **1**B show an example network diagram **100** utilized to describe the various embodiments. A cloud computing platform **110** is communicably connected to a network **120**. Examples of the cloud computing platform **110** may include a public cloud, a private cloud, a hybrid cloud, and the like. Examples for a public cloud, but are not limited to, AWS® by Amazon®, Microsoft Azure®, Google Cloud®, and the like. In some configurations, the disclosed embodiments operable in on premise virtual machines environments. The network **120** may be the Internet, the world-wide-web (WWW), a local area network (LAN), a wide area network (WAN), and other networks.

The arrangement of the example cloud computing platform **110** is shown in FIG. **1**B. As illustrated, the platform **110** includes a server **115** and a storage **117**, serving as the storage space for the server **115**. The server **115** is a physical device hosting at least one virtual machine (VM) **119**. The VM **119** is a protected VM, which may be any virtual cloud asset including, but not limited to, a software container, a micro-service, a serverless function, and the like.

The storage **117** emulates virtual discs for the VMs executed in by the server **115**. The storage **117** is typically connected to the server **115** through a high-speed connection, such as optic fiber allowing fast retrieval of data. In other configurations, the storage **117** may be part of the server **115**. In this example illustrated in FIG. **1**B, virtual disk **118-1** is allocated for the VM **119**. The server **115**, and hence the VM **119**, may be executed in a client environment **130** within the platform **110**.

The client environment **130** is an environment within the cloud computing platform **110** utilized to execute cloud-hosted applications of the client. A client may belong to a specific tenant. In some example embodiment, the client environment **130** may be part of a virtualized environment or on-premises virtualization environment, such as a VMware® based solution.

Also deployed in the cloud computing platform **110** is a security system **140** configured to perform the various disclosed embodiments. In some embodiments, the system **140** may be part of the client environment **130**. In an embodiment, the security system **140** may be realized as a physical machine configured to execute a plurality of virtual instances, such as, but not limited to virtual machines executed by a host server. In yet another embodiment, the security system **140** may be realized as a virtual machine executed by a host server. Such a host server is a physical machine (device) and may be either the server **115**, a dedicated server, a different shared server, or another virtualization-based computing entity, such as a serverless function.

In an embodiment, the interface between the client environment **130** and the security system **140** can be realized using APIs or services provided by the cloud computing platform **110**. For example, in AWS, a cross account policy service can be utilized to allow interfacing the client environment **130** with the security system **140**.

In the deployment, illustrated in FIG. **1**, the configuration of resources of the cloud computing platform **110** is performed by means of the management console **150**. As such, the management console **150** may be queried on the current deployment and settings of resources in the cloud computing platform **110**. Specifically, the management console **150** may be queried, by the security system **140**, about as the location (e.g., virtual address) of the virtual disk **118-1** in the storage **117**. The system **140** is configured to interface with the management console **150** through, for example, an API.

In some example embodiments, the security system **140** may further interface with the cloud computing platform **110** and external systems **170**. The external systems may include intelligence systems, security information and event management (SIEM) systems, and mitigation tools. The external intelligence systems may include common vulnerabilities and exposures (CVE®) databases, reputation services, security systems (providing feeds on discovered threats), and so on. The information provided by the intelligence systems may detect certain known vulnerabilities identified in, for example, a CVE database.

According to the disclosed embodiments, the security system **140** is configured to detect vulnerabilities and other cyber threats related to the execution VM **119**. The detection is performed while the VM **119** is live, without using any agent installed in the server **115** or the VM **119**, and without relying on cooperation from VM **119** guest OS. Specifically, the security system **140** can scan and detect vulnerable software, non-secure configuration, exploitation attempts, compromised asserts, data leaks, data mining, and so on. The security system **140** may be further utilized to provide security services, such as incident response, anti-ransomware, and cyber insurance by accessing the security posture.

In some embodiments, the security system **140** is configured to query the cloud management console **150** for the address of the virtual disk **118-1** serving the VM **119** and a location of the snapshot. A VM's snapshot is a copy of the machine's virtual disk (or disk file) at a given point in time. Snapshots provide a change log for the virtual disk and are used to restore a VM to a particular point in time when a failure error occurs. Typically, any data that was writable on a VM becomes read-only when the snapshot is taken. Multiple snapshots of a VM can be created at multiple possible point-in-time restore points. When a VM reverts to a snapshot, current disk and memory states are deleted and the snapshot becomes the new parent snapshot for that VM.

The snapshot of the VM **119** is located and may be saved from the virtual disk **118-1** is accessed by the system **140**. In an embodiment, the VM's **119** snapshot may be copied to the system **140**. If such a snapshot does not exist, the system **140** may take a new snapshot, or request such an action. The snapshots may be taken at a predefined schedule or upon predefined events (e.g., a network event or abnormal event). Further, the snapshots may be accessed or copied on a predefined schedule or upon predefined events. It should be noted that when the snapshot is taken or copied, the VM **119** still runs.

It should be noted that the snapshot of the virtual disk **118-1** may not be necessary stored in the storage **117**, but for ease of the discussion it is assumed that the snapshot is saved in the storage **117**. It should be further noted that the snapshot is being accessed without cooperation of the guest, virtual OS of the virtual machine.

The snapshot is parsed and analyzed by the security system **140** to detect vulnerabilities. This analysis of the snapshot does not require any interaction and/or information from the VM **119**. As further demonstrated herein, the analysis of the snapshot by the system **140** does not require any agent installed on the server **115** or VM **119**.

Various techniques can be utilized to analyze the snapshots, depending on the type of vulnerability and cyber threats to be detected. Following are some example embodiments for techniques that may be implemented by the security system **140**.

In an embodiment, the security system **140** is configured to detect whether there is vulnerable code executed by the VM **119**. The VM **119** being checked may be running, paused, or shutdown. To this end, the security system **140** is configured to match installed application lists, with their respective versions, to a known list of vulnerable applications. Further, the security system **140** may be configured to match the application files, either directly (using binary comparison) or by computing a cryptographic hash against database of files in vulnerable applications. The matching may be also on sub-modules of an application. Alternatively, the security system **140** may read installation logs of package managers used to install the packages of the application.

In yet another embodiment, the security system **140** is configured to verify whether the vulnerability is relevant to the VM **119**. For example, if there is a vulnerable version or module not in use, the priority of that issue is reduced dramatically.

To this end, the security system **140** may be configured to check the configuration files of the applications and operating system of the VM **119**; to verify access times to files by the operating system; and/or to analyze the active application and/or system logs in order to deduce what applications and modules are running.

In yet another embodiment, the security system **140** may instantiate a copy of the VM **119** and/or a subset of applications of the VM **119** on the server **115** or a separate server and monitor all activity performed by the instance of the VM. The execution of the instance of the VM is an isolated sandbox, which can be a full VM or subset of it, such as a software container (e.g., Docker® container) or another virtualized instances. The monitored activity may be further analyzed to determine abnormality. Such analysis may include monitoring of API activity, process creation, file activity, network communication, registry changes, and active probing of the said subset in order to assess its security posture. This may include, but not limited to, actively communicating with the VM **119**, using either legitimate communicate and/or attack attempts, to assess its posture and by that deriving the security posture of the entire VM **119**.

In order to determine if the vulnerability is relevant to the VM **119**, the security system **140** is configured to analyze the machine memory, as reflected in the page file. The page file is saved in the snapshot and extends how much system-committed memory (also known as "virtual memory") a system can back. In an embodiment, analyzing the page file allows deduction of running applications and modules by the VM **119**.

In an embodiment, the security system **140** is configured to read process identification number (PID) files and check their access or write times, which are matched against process descriptors. The PID can be used to deduce which processes are running, and hence the priority of vulnerabilities detected in processes existing on the disk. It should be noted the PID files are also maintained in the snapshot.

In yet another embodiment, the security system **140** is configured to detect cyber threats that do not represent vulnerabilities. For example, the security system **140** may detect and alert on sensitive data not being encrypted on the logical disk, private keys found on the disks, system credentials stored clearly on the disk, risky application features (e.g., support of weak cipher suites or authentication methods), weak passwords, weak encryption schemes, a disable address space layout randomization (ASLR) feature, suspicious manipulation to a boot record, suspicious PATH, LD_LIBRARY_PATH, or LD_PRELOAD definitions, services running on startup, and the like.

In an embodiment, the security system **140** may further monitor changes in sensitive machine areas, and alert on unexpected changes (e.g., added or changed application files without installation). In an example embodiment, this can be achieved by computing a cryptographic hash of the sensitive areas in the virtual disk and checking for differences over time.

In some embodiments, the detected cyber threats (including vulnerabilities) are reported to a user console **180** and/or a security information and event management (SIEM) system (not shown). The reported cyber threats may be filtered or prioritized based in part on their determined risk. Further, the reported cyber threats may be filtered or prioritized based in part on the risk level of the machine. This also reduces the number of alerts reported to the user.

In an embodiment, any detected cyber threats related to sensitive data (including personally identifiable information, PII) is reported at a higher priority. In an embodiment, such data is determined by searching for the PII, analyzing the application logs to determine whether the machine accessed PII/PII containing servers, or whether the logs themselves contain PII, and searching the machine memory, as reflected in the page file, for PII.

In an embodiment, the security system **140** may determine the risk of the VM **119** based on communication with an untrusted network. This can be achieved by analyzing the VM's **119** logs as saved in the virtual disk and can be derived from the snapshot.

In an example embodiment, the security system **140** may cause an execution of one or more mitigation actions. Examples of such actions may include blocking traffic from untrusted networks, halting the operation of the VM, quarantining an infected VM, and the like. The mitigation actions may be performed by a mitigation tool and not the system **140**.

It should be noted that the example implementation shown in FIG. **1** is described with respect to a single cloud computing platform **110** hosting a single VM **119** in a single server **115**, merely for simplicity purposes and without limitation on the disclosed embodiments. Typically, virtual machines are deployed and executed in a single cloud computing platform, a virtualized environment, or data center and can be protected without departing from the scope of the disclosure. It should be further noted that the disclosed embodiments can operate using multiple security systems **140**, each of which may operate in a different client environment.

FIG. **2** shows an example flowchart **200** illustrating a method for detecting cyber threats including potential vulnerabilities in virtual machines executed in a cloud computing platform according to some embodiments. The method may be performed by the security system **140**.

At S**210**, a request, for example, to scan a VM for vulnerabilities is received. The request may be received, or otherwise triggered every predefined time interval or upon detection of an external event. An external event may be a preconfigured event, such as a network event or abnormal event including, but not limited to, changes to infrastructure such as instantiation of an additional container on existing VM, image change on a VM, new VM created, unexpected shutdowns, access requests from unauthorized users, and the like. The request may at least designate an identifier of the VM to be scanned.

At S**220**, a location of a snapshot of a virtual disk of the VM to be scanned is determined. In an embodiment, S**220** may include determining the virtual disk allocated for the VM, prior to determining the location of the snapshot. As noted above, this can be achieved by querying a cloud management console. At S**230**, a snapshot of the virtual disk is accessed, or otherwise copied.

At S**240**, the snapshot is analyzed to detect cyber threats and potential vulnerabilities. S**240** may be also include detecting cyber threats that do not represent vulnerabilities. Examples for cyber threats and vulnerabilities are provided above.

In an embodiment, S**240** may include comparing the snapshot to some baseline, which may include, but is not limited to, a copy of the image used to create the VM, (e.g., lists of applications, previous snapshots), cryptographic hashes gathered in the previous scan, analyzing logs of the VMs, instantiating a copy of the VM and executing the instance or applications executed by the VM in a sandbox, analyzing the machine memory, as reflected in the page file, or any combination of these techniques. Some example embodiments for analyzing the snapshots and the types of detected vulnerabilities and threats are provided above.

At S**250**, the detected cyber threats and/or vulnerabilities are reported, for example, as alerts. In an embodiment, S**250** may include filtering and prioritizing the reported alerts. In an embodiment, the prioritization is based, in part, on the risk level of a vulnerable machine. The filtering and prioritizing allow to reduce the number of alerts reported to the user. The filtering can be done performed on external intelligence on the likelihood of this vulnerability being exploited, analyzing the machine configuration in order to deduce the vulnerability relevancy, and correlating the vulnerability with the network location, and by weighting the risk of this machine being taken over by the attacker by taking into consideration the criticality of the machine in the organization based by the contents stored or other assets accessible from the VM **110**.

At optional S**260**, a mitigation action may be triggered to mitigate a detected threat or vulnerability. A mitigation action may be executed by a mitigation tool and triggered by the system **140**. Such an action may include blocking traffic from untrusted networks, halting the operation of the VM, quarantining an infected VM, and the like.

FIG. **3** is an example block diagram of the security system **140** according to an embodiment. The security system **140** includes a processing circuitry **310** coupled to a memory **320**, a storage **330**, and a network interface **340**. In an embodiment, the components of the security system **140** may be communicatively connected via a bus **360**.

The processing circuitry **310** may be realized as one or more hardware logic components and circuits. For example, and without limitation, illustrative types of hardware logic components that can be used include field programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), application-specific standard products (ASSPs), system-on-a-chip systems (SOCs), general-purpose microprocessors, microcontrollers, digital signal processors (DSPs), and the like, or any other hardware logic components that can perform calculations or other manipulations of information.

The memory **310** may be volatile (e.g., RAM, etc.), non-volatile (e.g., ROM, flash memory, etc.), or a combination thereof. In one configuration, computer readable instructions to implement one or more embodiments disclosed herein may be stored in the storage **330**.

In another embodiment, the memory **320** is configured to store software. Software shall be construed broadly to mean 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, cause the processing circuitry **310** to perform the various processes described herein. Specifically, the instructions, when executed, cause the processing circuitry **310** to determine over-privileged roles vulnerabilities in serverless functions.

The storage **330** may be magnetic storage, optical storage, and the like, and may be realized, for example, as flash memory or other memory technology, CD-ROM, Digital Versatile Disks (DVDs), hard-drives, SSD, or any other medium which can be used to store the desired information. The storage **330** may store communication consumption patterns associated with one or more communications devices.

The network interface **340** allows the security system **140** to communicate with the external systems, such as intelligence systems, SIEM systems, mitigation systems, a cloud management console, a user console, and the like.

It should be understood that the embodiments described herein are not limited to the specific architecture illustrated in FIG. **3**, and other architectures may be equally used without departing from the scope of the disclosed embodiments.

The various embodiments disclosed herein can be implemented as hardware, firmware, software, or any combination thereof. Moreover, the software is preferably 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 comprising any suitable architecture. Preferably, the machine is 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 herein 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 is any computer readable medium except for a transitory propagating signal.

As used herein, the phrase "at least one of" followed by a listing of items means that any of the listed items can be utilized individually, or any combination of two or more of the listed items can be utilized. For example, if a system is described as including "at least one of A, B, and C," the system can include A alone; B alone; C alone; A and B in combination; B and C in combination; A and C in combination; or A, B, and C in combination.

All examples and conditional language recited herein are intended for pedagogical purposes to aid the reader in understanding the principles of the disclosed embodiment and the concepts contributed by the inventor to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the disclosed embodiments, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e., any elements developed that perform the same function, regardless of structure.

What is claimed is:

1. A method for securing virtual cloud assets against cyber vulnerabilities in a cloud computing environment, the method comprising:

determining, using an API or service provided by the cloud computing environment, a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the protected virtual cloud asset is instantiated in the cloud computing environment;

accessing, based on the determined location and using an API or service provided by the cloud computing environment, the snapshot of the at least one virtual disk;

analyzing the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications;

determining, based on the matching, an existence of potential cyber vulnerabilities of the protected virtual cloud asset;

determining whether the matching installed applications are used by the protected virtual cloud asset;

prioritizing the potential cyber vulnerabilities based on the use determinations; and

reporting the determined potential cyber vulnerabilities, as prioritized alerts according to the use determinations.

2. The method of claim **1**, wherein determining whether the matching installed applications are used by the protected virtual cloud asset includes determining whether at least one of the matching installed applications is not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for a matching installed application not in use.

3. The method of claim **1**, wherein determining whether the matching installed applications are used by the protected virtual cloud asset includes checking configuration files of the matching installed applications to determine whether at least one of the matching installed applications is not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for a matching installed application not in use.

4. The method of claim **1**, wherein determining whether the matching installed applications are used by the protected virtual cloud asset includes verifying access times to files by an operating system of the protected virtual cloud asset to determine whether at least one of the matching installed applications is not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for a matching installed application not in use.

5. The method of claim **2**, wherein determining whether the matching installed applications are used by the protected virtual cloud asset includes analyzing application logs or system logs to determine matching installed applications not in use by the protected virtual cloud asset, and wherein prioritizing reduces priority of potential cyber vulnerabilities for at least one of the matching installed applications not in use.

6. The method of claim **1**, wherein reporting the determined potential cyber vulnerabilities includes communicating the determined potential cyber vulnerabilities to a user console or a security information and event management (SIEM) system.

7. The method of claim **1**, wherein analyzing the snapshot of the at least one virtual disk further includes matching application files on the snapshot of the at least one virtual disk directly against application files associated with a known list of vulnerable applications.

8. The method of claim **1**, wherein analyzing the snapshot of the at least one virtual disk further includes matching application files on the snapshot of the at least one virtual disk by:

computing a cryptographic hash against at least one application file to be matched; and

matching the computed cryptographic hash against a database of files associated with a known list of vulnerable applications.

9. The method of claim **1**, wherein analyzing the snapshot of the at least one virtual disk further includes:

parsing the snapshot of the at least one virtual disk; and

scanning the parsed snapshot of the at least one virtual disk to detect the potential cyber vulnerabilities.

10. The method of claim **9**, wherein scanning the parsed snapshot further includes at least one of:

checking configuration files of applications and an operating system installed in the protected virtual cloud asset;

verifying access times to files by the operating system installed in the in the protected virtual cloud asset; or

analyzing system logs to deduce applications and modules executed in the protected virtual cloud asset.

11. The method of claim **1**, further comprising mitigating a potential cyber vulnerability posing a risk to the protected virtual cloud asset.

12. The method of claim **11**, wherein mitigating a potential cyber vulnerability includes at least one of:

blocking traffic from untrusted networks to the protected virtual cloud asset,

halting operation of the protected virtual cloud asset, or

quarantining the protected virtual cloud asset.

13. The method of claim **1**, wherein determining the location of the snapshot of at least one virtual disk further includes determining a virtual disk allocated to the protected virtual cloud asset.

14. The method of claim **1**, wherein determining the location of the snapshot of at least one virtual disk includes taking a new snapshot of the at least one virtual disk of a protected virtual cloud asset.

15. The method of claim **1**, wherein determining the location of the snapshot of at least one virtual disk includes requesting the taking of a new snapshot of the at least one virtual disk of a protected virtual cloud asset.

16. The method of claim **1**, wherein determining the location of the snapshot of at least one virtual disk further includes querying a cloud management console of the cloud computing environment for the location of the snapshot and the location of the virtual disk.

17. The method of claim **1**, further comprising:

making a copy of a snapshot of the virtual disk; and

wherein analyzing the snapshot includes analyzing the copy of the snapshot by matching installed applications identified in the copy of the snapshot with applications on the known list of vulnerable applications.

18. A non-transitory computer readable medium containing instructions that when executed by at least one processor cause the at least one processor to perform operations for securing virtual cloud assets against cyber vulnerabilities in a cloud computing environment, the operations comprising:

determining, using an API or service provided by the cloud computing environment, a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the protected virtual cloud asset is instantiated in the cloud computing environment;

accessing, based on the determined location and using an API or service provided by the cloud computing environment, the snapshot of the virtual disk;

analyzing the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications;

determining, based on the matching, an existence of a plurality of potential cyber vulnerabilities;

determining whether the matching installed applications are used by the protected virtual cloud asset;

prioritizing the potential cyber vulnerabilities based on the use determinations; and

reporting the determined plurality of potential cyber vulnerabilities, as prioritized alerts according to the use determinations.

19. The non-transitory computer readable medium of claim **18**, wherein analyzing the snapshot of the at least one virtual disk further includes matching application files on the snapshot of the at least one virtual disk directly against application files associated with a known list of vulnerable applications.

20. The non-transitory computer readable medium of claim **18**, wherein determining the location of the snapshot of at least one virtual disk further includes querying a cloud management console of the cloud computing environment for the location of the snapshot and the location of the virtual disk.

21. The non-transitory computer readable medium of claim **18**, wherein:

the instructions further comprise making a copy of a snapshot of the virtual disk; and

wherein analyzing the snapshot includes analyzing the copy of the snapshot by matching installed applications identified in the copy of the snapshot with applications on the known list of vulnerable applications.

22. A system for securing virtual cloud assets against cyber vulnerabilities in a cloud computing environment, the system comprising:

at least one processor configured to:

determine, using an API or service provided by the cloud computing environment, a location of a snapshot of at least one virtual disk of a protected virtual cloud asset, wherein the protected virtual cloud asset is instantiated in the cloud computing environment;

access, based on the determined location and using an API or service provided by the cloud computing environment, the snapshot of the virtual disk;

analyze the snapshot of the at least one virtual disk by matching installed applications with applications on a known list of vulnerable applications;

determine, based on the matching, an existence of a plurality of potential cyber vulnerabilities;

determine whether the matching installed applications are used by the protected virtual cloud asset;

prioritize the potential cyber vulnerabilities based on the use determinations; and

report the determined plurality of potential cyber vulnerabilities, as prioritized alerts according to the use determinations.

23. The system of claim **22**, wherein determining the location of the snapshot of at least one virtual disk further includes taking a new snapshot of the at least one virtual disk of a protected virtual cloud asset.

24. The system of claim **22**, wherein determining the location of the snapshot of at least one virtual disk further includes requesting the taking of a new snapshot of the at least one virtual disk of a protected virtual cloud asset.

25. The system of claim **22**, wherein:

the at least one processor is further configured to make a copy of a snapshot of the virtual disk; and

wherein analyzing the snapshot includes analyzing the copy of the snapshot by matching installed applications identified in the copy of the snapshot with applications on the known list of vulnerable applications.

* * * * *

EXHIBIT 3

orca security

SideScanning™ Technical Brief

SideScanning™ – How the Engine that Powers Orca Security Works

Every organization is searching for effective ways to scan its cloud estate to look for risks. These include vulnerabilities, misconfigurations, malware, improper segmentation, and customer data at risk. They're also seeking to verify compliance with security frameworks and government/industry regulations.

Orca Security introduces an innovative approach that secures the entire cloud estate without disrupting business operations in live environments—and with absolutely no need for agents or network scanners that fail to account for everything.

Table of Contents

Background

As organizations put their workloads into the cloud, they must have the security policies, tools, and controls that will protect their assets from data breaches, misuse, and attacks. Securing the cloud requires complete visibility into vulnerable software, compromised resources, live exploitation attempts, neglected assets, misconfigurations, and more.

Until now, providing for cloud security has involved legacy tools that have been adapted from their original on-premise versions. These adaptations bring with them tremendous cost, complexity, and limitations that can prohibit full visibility—especially in complex, multi-cloud environments.

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

Orca Security delivers instant-on, work-load level visibility across 100% of AWS, Azure, and GCP assets without running a single opcode in the customer environment. The result is to help organizations:

- Detect risks such as vulnerabilities, malware, misconfiguration, lateral movement risk, and unsecured sensitive data
- See its entire full-stack cloud inventory
- Discover and see previously missed assets

This new technology is called SideScanning™ and provides both technical and business advantages to organizations. This white paper focuses on the technical aspects of how the engine that powers Orca Security works.

Traditional Scanning Methods

There aren't many options when performing security vulnerability scans of a physical machine. When businesses began to move their workloads to the cloud, these technologies were lifted and shifted there. But cloud workloads are vastly different than '90s-style physical servers running on bare metal. 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, and to which it's possible to authenticate. What's more, the assets must be capable of having an agent installed and maintained, and they 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 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 scans can derive good results on 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 really make sure it's set up that way. This leaves many organizations exposed to vulnerabilities in areas where the scanner hasn't reached.

While unauthenticated scanners act like an attacker, they often get stuck in flows where a real attacker would not. In many cases there are measures in place, such as CAPTCHA, which 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. For example, 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 their cloud infrastructure footprint. This is mostly due to their inability to keep up with the incredibly high TCO involved with agent deployment and maintenance.

Cloud Security Posture Management (CSPM)

But there is one scanning tool developed specifically for the cloud. Rather than going inside the machine, a cloud security posture manager (CSPM) analyzes the cloud configuration itself for errors. This type of tool is used to discover, assess, and solve cloud misconfigurations, but provides shallow coverage where cloud security is concerned. CSPMs will never detect critical issues 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 on the issues they see. But without the context seen by the CSPM, this results in alert fatigue on behalf of security analysts.

A Radical New Approach – SideScanning™

Orca Security uses our patent-pending SideScanning™ technology. It's a radical approach because Orca doesn't go *inside* each workload to fetch 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 the cloud infrastructure. It supports Amazon Web Services, Google Cloud Platform, and Microsoft Azure. 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 everything within is clearly visible. Orca can probe the read-only view it has obtained in an entirely touchless manner.

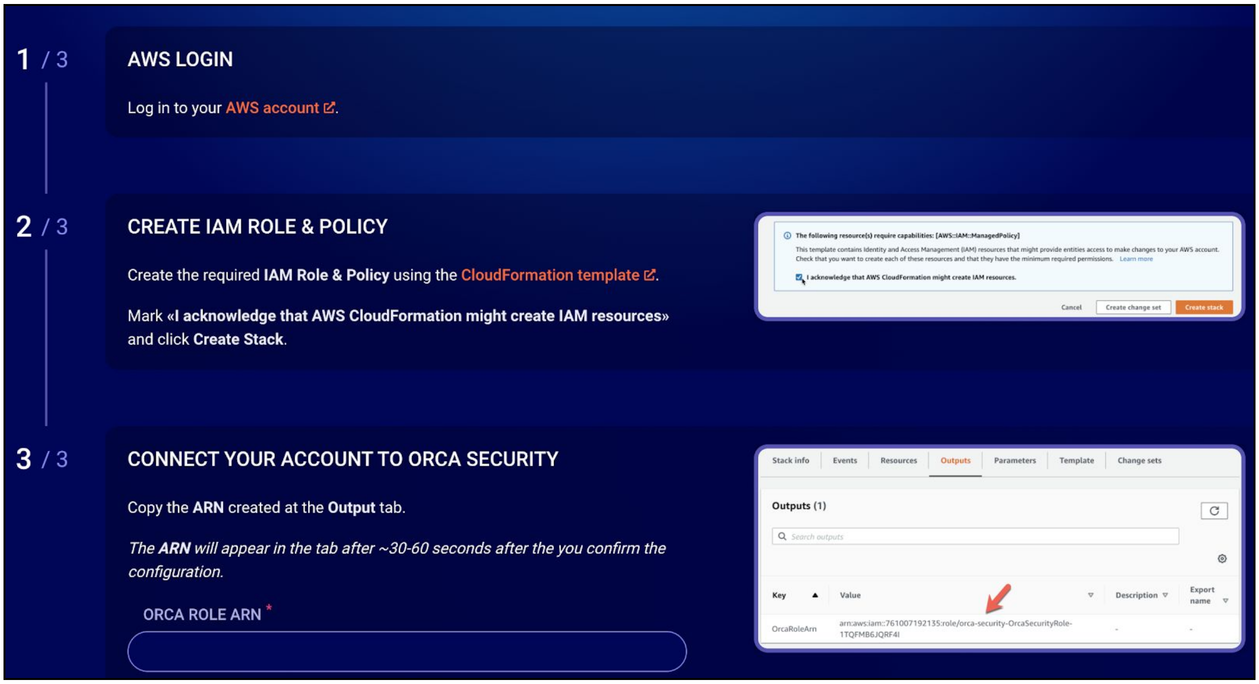
Orca doesn't affect or run on the cloud machines, where it might consume resources. This lets an organization fully deploy it across 100% of its cloud environment without worrying about potential side effects. And it does this without the friction of working with disparate teams (e.g.,, DevOps) to assess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

How SideScanning™ Works

Although this paper uses AWS terminology, readers can apply the same information to Azure and GCP, where everything works virtually the same.

Onboarding

Overview – Orca's onboarding process is simple and quick. You provide it 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.



1 / 3 AWS LOGIN

Log in to your AWS account.

2 / 3 CREATE IAM ROLE & POLICY

Create the required **IAM Role & Policy** using the CloudFormation template.

Mark «**I acknowledge that AWS CloudFormation might create IAM resources**» and click **Create Stack**.

The following resource(s) require capabilities: [AWS::IAM::ManagedPolicy]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. Learn more

☑ **I acknowledge that AWS CloudFormation might create IAM resources.**

3 / 3 CONNECT YOUR ACCOUNT TO ORCA SECURITY

Copy the **ARN** created at the **Output** tab.

*The **ARN** will appear in the tab after ~30-60 seconds after the you confirm the configuration.*

ORCA ROLE ARN *

Outputs (1)

Key	Value	Description	Export name
OrcaRoleArn	arn:aws:iam::761007192135:role/orca-security-OrcaSecurityRole-1TQFMB6JQRF4I	-	-

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.)

Orca requests permission to read encrypted key management service (KMS) volumes so as 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.

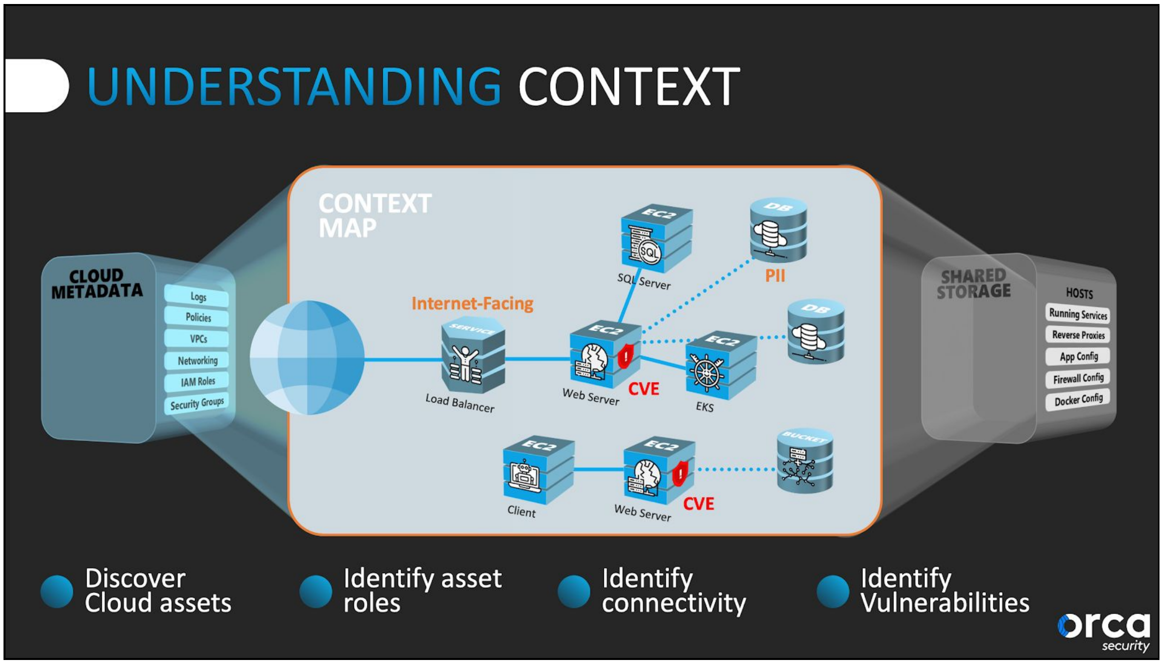
The Scan Process

After you've added your account to Orca, the 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

Once all of the information is gathered, the scan 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 everything found in the customer account.



UNDERSTANDING CONTEXT

- Discover Cloud assets
- Identify asset roles
- Identify connectivity
- Identify Vulnerabilities

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 ones are connected to each other in order 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.

The Data Plane Path

Its data plane path, or asset scan, is what makes Orca stand out as a comprehensive cloud security tool. 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. A collector—an ephemeral EC2 instance—is created in the Orca side of the cloud (usually in the form of a spot instance). It 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:

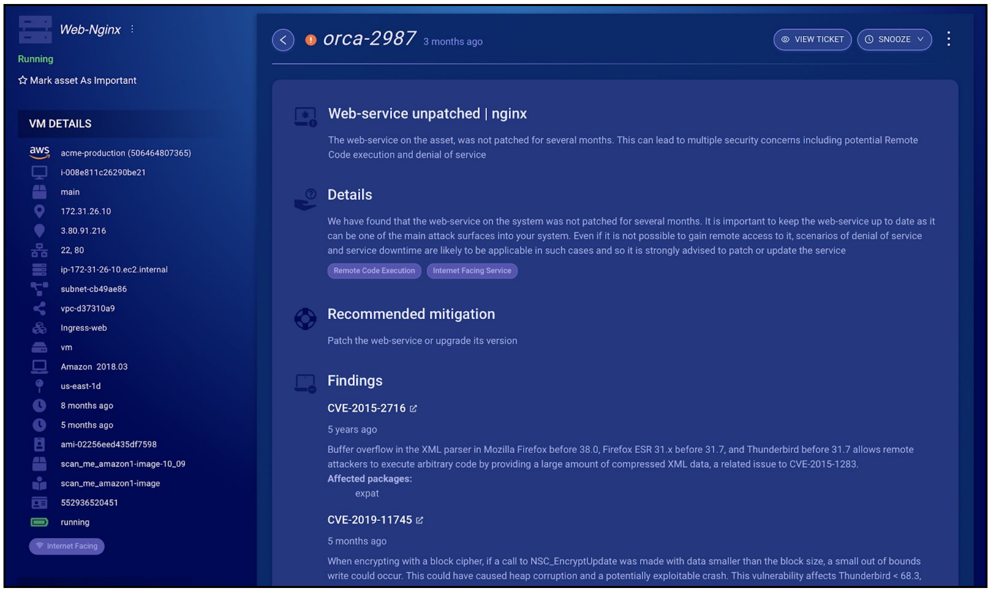
- Vulnerability Scanning
- Configuration Scanning
- Malware Scanning
- Lateral Movement Risk
- Exploitable Keys and Weak Password Detection
- Sensitive Information Scanning
- Container Scanning

Each step is discussed in more detail in a moment.

Orca doesn't run your machine volumes directly. 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.

Vulnerability Scanning



Web-Nginx

Running

☆ Mark asset As Important

VM DETAILS

acme-production (506464807365)
i-008e811c26290be21
main
172.31.26.10
3.80.91.216
22, 80
ip-172-31-26-10.ec2.internal
subnet-cb49ae86
vpc-d37310a9
Ingress-web
vm
Amazon 2018.03
us-east-1d
8 months ago
5 months ago
ami-02256eed435df7598
scan_me_amazon1-image-10_09
scan_me_amazon1-image
552936520451
running

Internet Facing

orca-2987 3 months ago

Web-service unpatched | nginx

The web-service on the asset, was not patched for several months. This can lead to multiple security concerns including potential Remote Code execution and denial of service

Details

We have found that the web-service on the system was not patched for several months. It is important to keep the web-service up to date as it can be one of the main attack surfaces into your system. Even if it is not possible to gain remote access to it, scenarios of denial of service and service downtime are likely to be applicable in such cases and so it is strongly advised to patch or update the service

Remote Code Execution Internet Facing Service

Recommended mitigation

Patch the web-service or upgrade its version

Findings

CVE-2015-2716

5 years ago

Buffer overflow in the XML parser in Mozilla Firefox before 38.0, Firefox ESR 31.x before 31.7, and Thunderbird before 31.7 allows remote attackers to execute arbitrary code by providing a large amount of compressed XML data, a related issue to CVE-2015-1283.
Affected packages:
expat

CVE-2019-11745

5 months ago

When encrypting with a block cipher, if a call to NSC_EncryptUpdate was made with data smaller than the block size, a small out of bounds write could occur. This could have caused heap corruption and a potentially exploitable crash. This vulnerability affects Thunderbird < 68.3,

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

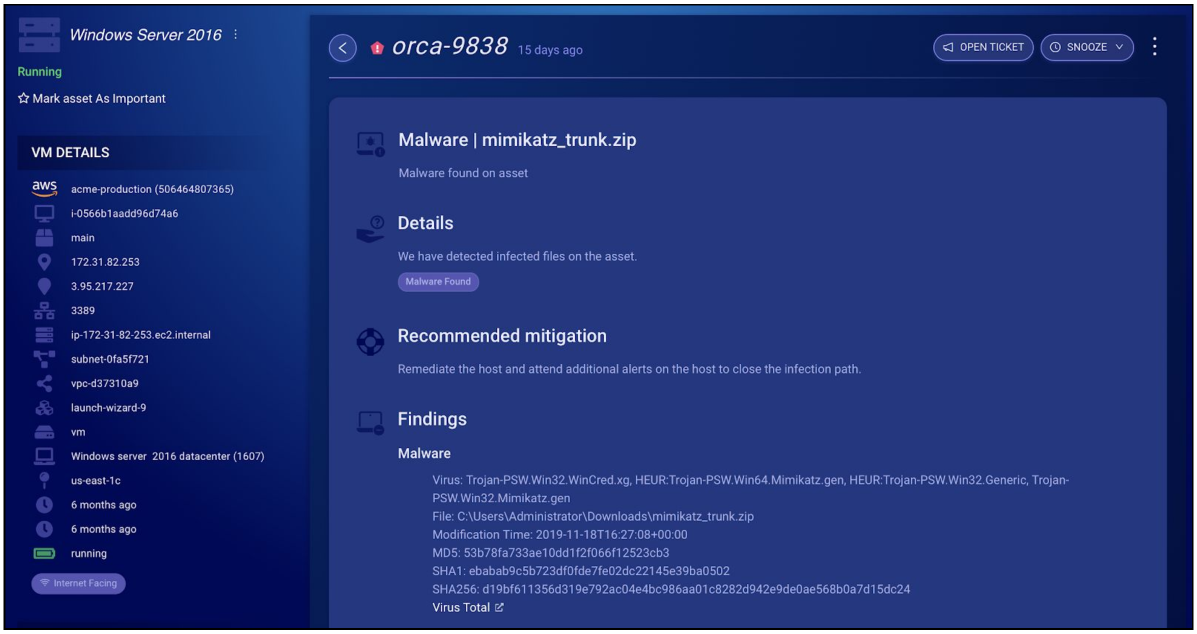
Note: This list is current as of May 2020. More sources will be occasionally added.

Configuration Scanning

Because some packages might only be vulnerable in specific configurations, this is later 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



Windows Server 2016

Running

☆ Mark asset As Important

VM DETAILS

aws acme-production (506464807365)
i-0566b1aadd96d74a6
main
172.31.82.253
3.95.217.227
3389
ip-172-31-82-253.ec2.internal
subnet-0fa5f721
vpc-d37310a9
launch-wizard-9
vm
Windows server 2016 datacenter (1607)
us-east-1c
6 months ago
6 months ago
running

Internet Facing

orca-9838 15 days ago

Malware | mimikatz_trunk.zip

Malware found on asset

Details

We have detected infected files on the asset.

Malware Found

Recommended mitigation

Remediate the host and attend additional alerts on the host to close the infection path.

Findings

Malware

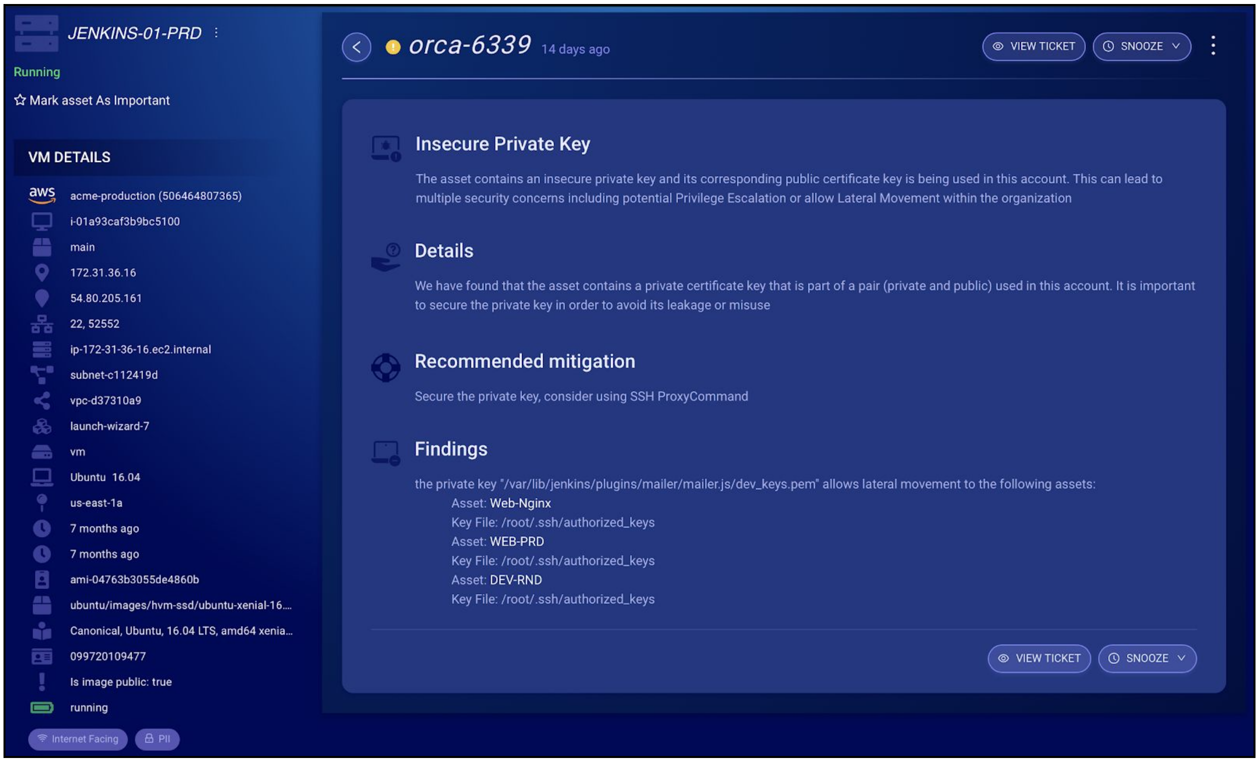
Virus: Trojan-PSW.Win32.WinCred.xg, HEUR:Trojan-PSW.Win64.Mimikatz.gen, HEUR:Trojan-PSW.Win32.Generic, Trojan-PSW.Win32.Mimikatz.gen
File: C:\Users\Administrator\Downloads\mimikatz_trunk.zip
Modification Time: 2019-11-18T16:27:08+00:00
MD5: 53b78fa733ae10dd1f2f066f12523cb3
SHA1: ebabab9c5b723df0fde7fe02dc22145e39ba0502
SHA256: d19bf611356d319e792ac04e4bc986aa01c8282d942e9de0ae568b0a7d15dc24
Virus Total

Another data 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 totally on the Orca side, so it doesn't affect production workloads. Orca is very liberal with respect to its compute operation devoted to scanning because—unlike an agent—it isn't limited by the customer machine's CPU and available memory.

Another aspect 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.

Detecting Lateral Movement Risk, Exploitable Keys, and Weak Passwords



aws acme-production (506464807365)
i-01a93caf3b9bc5100
main
172.31.36.16
54.80.205.161
22, 52552
ip-172-31-36-16.ec2.internal
subnet-c112419d
vpc-d37310a9
launch-wizard-7
vm
Ubuntu 16.04
us-east-1a
7 months ago
7 months ago
ami-04763b3055de4860b
ubuntu/images/hvm-ssd/ubuntu-xenial-16....
Canonical, Ubuntu, 16.04 LTS, amd64 xenia...
099720109477
Is image public: true
running

Insecure Private Key

The asset contains an insecure private key and its corresponding public certificate key is being used in this account. This can lead to multiple security concerns including potential Privilege Escalation or allow Lateral Movement within the organization

Details

We have found that the asset contains a private certificate key that is part of a pair (private and public) used in this account. It is important to secure the private key in order to avoid its leakage or misuse

Recommended mitigation

Secure the private key, consider using SSH ProxyCommand

Findings

the private key "/var/lib/jenkins/plugins/mailer/mailer.js/dev_keys.pem" allows lateral movement to the following assets:

Asset: Web-Nginx
Key File: /root/.ssh/authorized_keys
Asset: WEB-PRD
Key File: /root/.ssh/authorized_keys
Asset: DEV-RND
Key File: /root/.ssh/authorized_keys

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.

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 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 issues that are only exploitable from internal machines but 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

Also in the data plane path, Orca scans any containers—just as it has the host machines. Here it reconstructs each container's layered file system and recursively runs all the collection steps on it, thereby yielding the same information previously outlined. Orca reads the container's network configuration so as to extend the contextual map built during its control plane assessment phase.

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.

Combining Information, Analysis, and Reporting

Continuing to build on its ability to achieve such high fidelity and meaningful results. 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 augments this information with a vulnerability management solution. For example, Orca takes a customer's instance, maps the running services on it, then check them for vulnerabilities. It 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.)

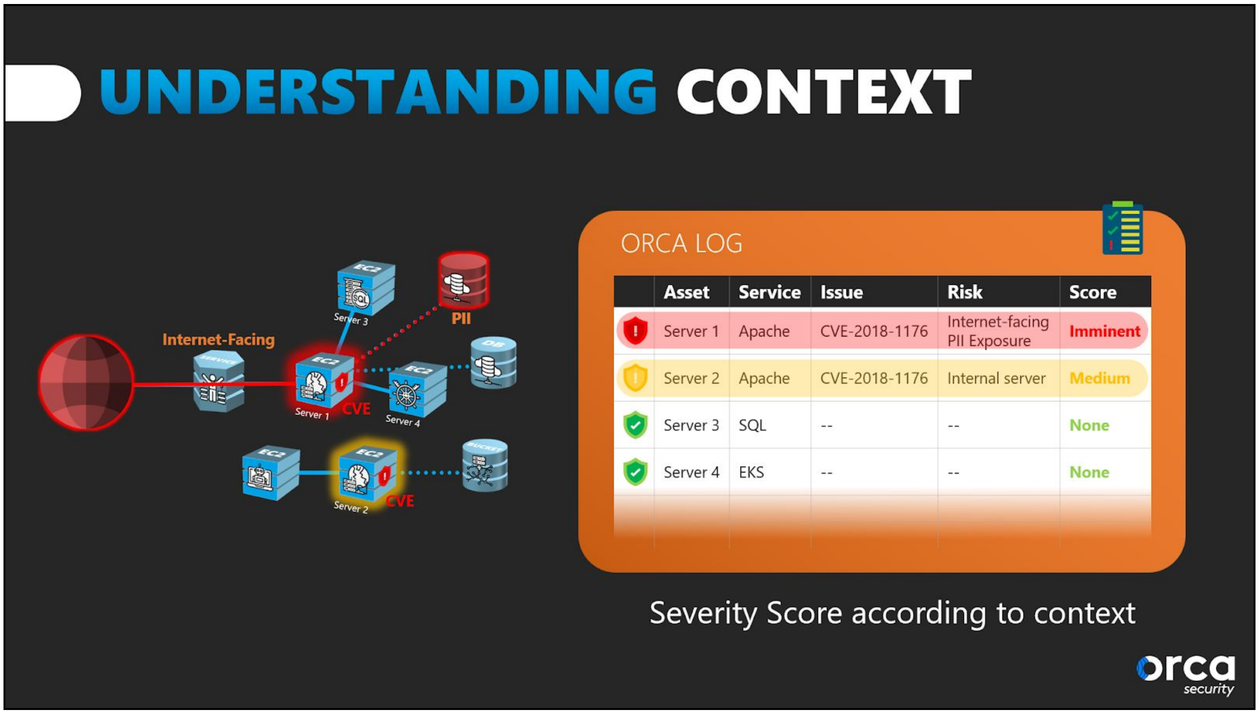
Consider a vulnerability in a web service. Orca scores it as:

- 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. It 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 it has 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.

It's a big advantage that the "bird's eye view" of the control plane path and the "detailed view" of the data plane path are done by a single vendor and all the data is in a single holistic database. Now, everything can be connected together and viewed in full context. No integration of disparate data sources is necessary.



UNDERSTANDING CONTEXT

ORCA LOG

	Asset	Service	Issue	Risk	Score
!	Server 1	Apache	CVE-2018-1176	Internet-facing PII Exposure	Imminent
!	Server 2	Apache	CVE-2018-1176	Internal server	Medium
✓	Server 3	SQL	--	--	None
✓	Server 4	EKS	--	--	None

Severity Score according to context

orca security

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 the status quo, where an organization purchases best of breed security tools for each of these to perform its own, independent vulnerability detection. Getting the set of tools to talk to one another and provide 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.

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 duration because it's not internet-facing or connected to other machines, its risk score will be low—despite the fact that it has hundreds of vulnerabilities. This helps analysts in prioritizing which security fixes to address first.

Applicability to Containers

Orca's data aggregation and deep analysis apply to containers just as it does to discrete workloads. As Orca processes containers, network information about each container is parsed into the map. This means Orca understands which services within the container are mapped for external attackers, which ports are used within each, and which protocols are externally accessible. All such context is merged into the "master map" discussed earlier, where it's continually augmented with even more context enhancing information to keep it current.

In Summary

The 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 deeper visibility into AWS, Azure, and GCP without the gaps in coverage and operational costs of agents. With Orca Security, there are no overlooked assets, no DevOps headaches, and no performance hits on live environments.

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 you with only the critical few that matter, along with their precise path to remediation.

Delivered as SaaS, Orca Security's patent-pending SideScanning™ technology reads your cloud configuration and workloads' run-time block storage out-of-band. It detects vulnerabilities, malware, misconfigurations, lateral movement risk, weak and leaked passwords, and high-risk data such as PII.

For more information, please visit https://orca.security

EXHIBIT 4

Agentless Scanning

Wiz uses several techniques to scan your entire cloud environment without a single agent or sidecar deployed on your workloads. This assures that you can get Wiz up and running across your environment in minutes without suffering from the coverage gaps that the limited deployment of agents typically cause.

How it works

Cloud API interrogation

Wiz connects using a reader role to your cloud APIs (AWS, Azure, GCP, Kubernetes, etc.) in order to list the cloud resources and interrogate the control plane for their configuration.

Workload scanning

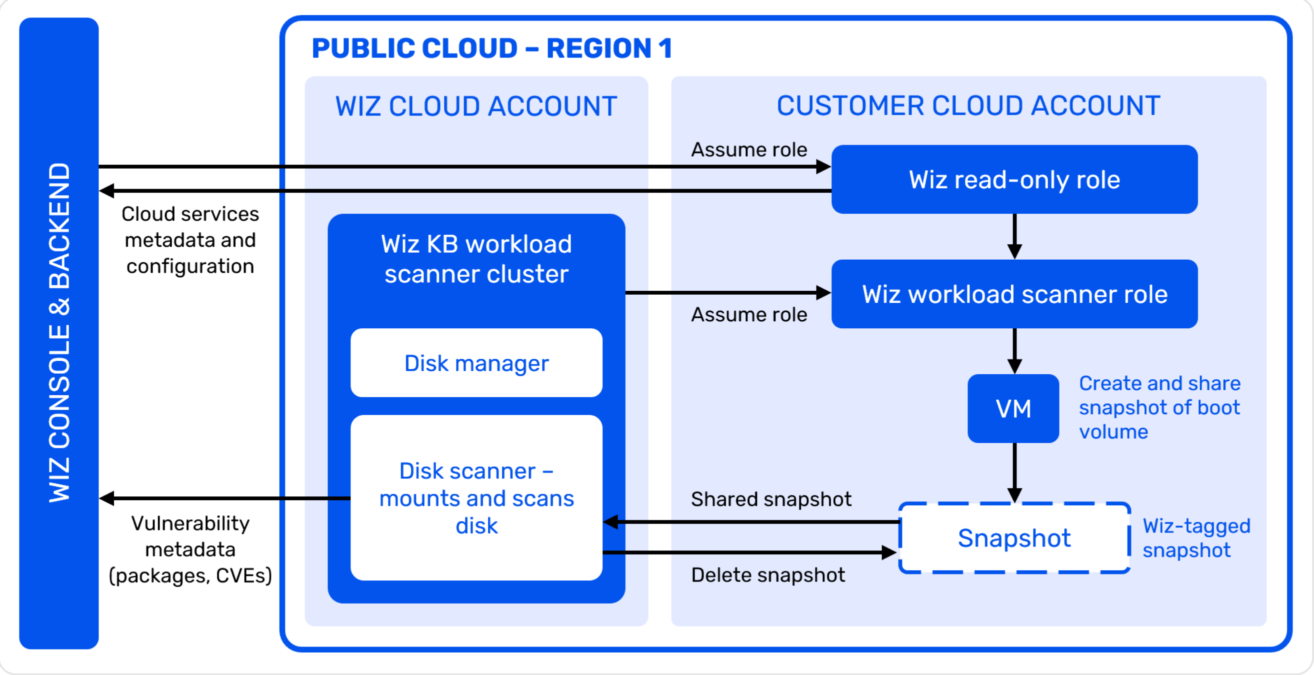
Snapshot scanning is the new way to perform workload scanning. Instead of using an intrusive agent, Wiz leverages cloud-native tools to perform scans without interrupting or impacting production workloads. Just like an MRI performs a 3D scan of the body without affecting the body itself, snapshot scanning achieves deep analysis of the workload without any impact or interruption to the live workload.

At its core, snapshot scanning is a very simple process. In order to scan the workload, a snapshot is created from the running workload and that snapshot is scanned by Wiz to extract vulnerabilities and misconfigurations.

The snapshots are created by the Wiz workload scanner, and live only during the scan period, always remaining within the customer tenant. Once the snapshot is created, a workload scanner running within the same region performs the snapshot scan by mounting a read-only volume that is backed by the snapshot. Once the scan is complete, the snapshot is deleted.

The results of the scan are then sent back to the Wiz backend. For a detailed list of the results sent see below.

Snapshot scanning architecture


WIZ CONSOLE & BACKEND
PUBLIC CLOUD – REGION 1
WIZ CLOUD ACCOUNT
CUSTOMER CLOUD ACCOUNT
Assume role
Wiz read-only role
Cloud services metadata and configuration
Wiz KB workload scanner cluster
Disk manager
Wiz workload scanner role
Assume role
VM
Create and share snapshot of boot volume
Disk scanner – mounts and scans disk
Vulnerability metadata (packages, CVEs)
Shared snapshot
Snapshot
Wiz-tagged snapshot
Delete snapshot

Encrypted volumes

Wiz supports encrypted volumes for all cloud native encryption types in AWS, Azure and GCP.

- In AWS, this is achieved without Wiz having access to the original encryption key thanks to the permission kms:ReEncryptTo.
- In Azure and GCP, this is supported with the standard snapshot and volumes permissions required by the Azure connector and GCP connector. No additional permissions are required, as creating a volume from a snapshot of an encrypted volume does not require additional permissions or encryption methods.

The snapshotting process

There are four distinct steps in the snapshot scan process:

1. Scan configuration—The list of disks for scanning is composed by the cloud fetcher leveraging the cloud provider APIs and sent to the Wiz workload scanner.
2. Snapshot creation—The workload scanner, which runs in a dedicated account, creates the snapshot and shares it with the scanner cluster. These snapshots are created with 'wiz: auto-gen-snapshot' tag to help identify them.
3. Snapshot scan—The snapshot is mapped as a read-only volume and scanned. The scan results include metadata on packages, vulnerabilities and mis-configurations and are sent to the backend.
4. Cleanup—The snapshot is deleted from the customer tenant.

The Wiz workload scanner runs in a dedicated account, or it can be deployed by the customer via the Outpost deployment (see the Wiz Outpost Overview).

FAQ

What operating systems, file systems, container runtime technologies and virtual appliances are supported?

Wiz scans:

- Operating systems—see table below
- File systems—NTFS, ext2, ext3, ext4, XFS, OSTree
- Encrypted file systems—crypto_LUKS (Azure integration) and BitLocker (Azure integration)
- Container runtimes—Docker, containerd, CRI-O
- Virtual appliances—F5 BIG-IP Advanced Firewall Manager, FortiOS, IBM Security Access Manager (ISAM), IBM Security Verify Access (formerly ISAM), PAN-OS

> ℹ VMs running containers with a supported OS image on a non-supported host OS are still scanned by Wiz.

Operating System	Detection	Technologies	Vulnerabilities	Malware	Secrets
SUSE Linux Enterprise Server	✅	✅	✅	✅	✅
Amazon Linux 2022	✅	✅	✅	✅	✅
Windows Server 2022	✅	✅	✅	✅	✅
Windows 11	✅	✅	✅	✅	✅
Container-Optimized OS	✅	✅	✅	✅	✅
Linux Photon	✅	✅	✅	✅	✅
Linux Oracle	✅	✅	✅	✅	✅
Amazon Linux 2	✅	✅	✅	✅	✅
Amazon Linux AMI	✅	✅	✅	✅	✅
Azure Virtual Desktop	✅	✅	✅	✅	✅
Windows Server 2019	✅	✅	✅	✅	✅

Operating System	Detection	Technologies	Vulnerabilities	Malware	Secrets
Windows Server 2016	✅	✅	✅	✅	✅
Windows Server 2012 R2	✅	✅	✅	✅	✅
Windows Server 2012	✅	✅	✅	✅	✅
Windows Server 2008 R2	✅	✅	✅	✅	✅
Windows Server 2008	✅	✅	✅	✅	✅
Windows Server 2003 R2	✅	✅	✅	✅	✅
Windows Server 2003	✅	✅	✅	✅	✅
Windows Server	✅	✅	✅	✅	✅
Windows 8.1	✅	✅	✅	✅	✅
Windows 7	✅	✅	✅	✅	✅
Windows 10	✅	✅	✅	✅	✅
Linux Alpine	✅	✅	✅	✅	✅
Linux CentOS	✅	✅	✅	✅	✅
Amazon Linux	✅	✅	✅	✅	✅
Linux openSUSE	✅	✅	✅	✅	✅
Linux Red Hat	✅	✅	✅	✅	✅
Linux Fedora	✅	✅	✅	✅	✅
Linux Ubuntu	✅	✅	✅	✅	✅
Linux Debian	✅	✅	✅	✅	✅
Wind River Linux	✅	Partial	Partial	✅	✅
Oracle Linux Server	✅	Partial	Partial	✅	✅

Operating System	Detection	Technologies	Vulnerabilities	Malware	Secrets
Buildroot	✅	Partial	Partial	✅	✅
McAfee Linux OS	✅	Partial	Partial	✅	✅
Wind River Linux	✅	Partial	Partial	✅	✅
Rocky Linux	✅	Partial	Partial	✅	✅
PexOS	✅	Partial	Partial	✅	✅
TanOS	✅	Partial	Partial	✅	✅
Appgate SDP	✅	Partial	Partial	✅	✅
Arch Linux	✅	Partial	Partial	✅	✅
Aruba ClearPass Platform	✅	Partial	Partial	✅	✅
Common Base Linux Delridge	✅	Partial	Partial	✅	✅
Silver Peak VXOA	✅	Partial	Partial	✅	✅
Linux Gentoo	✅	Partial	Partial	✅	✅
Trend Micro Smart Protection Server	✅	Partial	Partial	✅	✅
Clear Linux OS	✅	Partial	Partial	✅	✅
Darktrace OS	✅	Partial	Partial	✅	✅
Imperva SecureSphere	✅	Partial	Partial	✅	✅
Barracuda CloudGen Firewall	✅	Partial	Partial	✅	✅
F5 TMOS Linux	✅	Partial	Partial	✅	✅
N-centralOS Linux	✅	Partial	Partial	✅	✅

Operating System	Detection	Technologies	Vulnerabilities	Malware	Secrets
MgmtOS	✅	Partial	Partial	✅	✅
Sangoma Linux	✅	Partial	Partial	✅	✅
AWS BottleRocket	✅	-	-	-	-

> ℹ Partial support means that Wiz can apply only file-based detection.

How much data is flowing in terms of byte transfer?

Each disk scan is around 2-3 MB.

What attributes and parameters are transferred for any resources while scanning for any finding, secret, vulnerability, or data?

- List of installed packages + versions
- List of programming languages libraries + versions
- Local users
- Authentication configuration
- Operating system info
- Hashes of all files
- CIS benchmarks output
- Secret metadata (without the sensitive info)
- Deployed Git repositories
- Deployed containers
- For Windows machines: installed programs, services, and installed KBs

How does Wiz identify the compute instances' boot volumes?

It depends on your cloud provider:

- AWS—For each instance fetched using the API, AWS reports the `RootDeviceName` (e.g. - `"RootDeviceName": "/dev/sda1"`) and `BlockDeviceMappings` . The volume ID written under the DeviceName from the RootDeviceName is the OS volume; the rest are the data volumes.
- Azure—For each instance fetched using the API, Azure reports the `osDisk` . The disk under osDisk is the boot volume; the rest (under dataDisks) are the data volumes.

- GCP—For each instance fetched using API, GCP reports whether it is a `boot` volume. The boot volume is marked using `boot: true`, and the rest using `boot: false`.

Does Wiz capture ephemeral resources?

Yes, all VMs that exist when the snapshot is created are represented on the Security Graph. Moreover, Wiz groups all ephemeral resources instantiated from the same parent into a single Compute Instance Group, which is a persistent object on the Security Graph. All vulnerabilities, Findings, Issues, etc. associated with ephemeral resources are attached to their parent Compute Instance Groups in order to prevent duplication.

However, Wiz does not retroactively track the number of ephemeral resources, so the number of ephemeral resources in a Compute Instance Group does not reflect its "history". For instance, if a particular Compute Instance Group included 1,000 VMs at 10:00 am but 990 of them were taken down at 12:00 pm, then Wiz would show only the 10 VMs that still existed when the scan occurred the following night.

Does Wiz detect spot instances and databricks instance pools as created from the same group?

Yes, Wiz groups together spot instances and databricks instances based on common tags and presents them as compute instance groups. The tag keys used for grouping are:

- DatabricksInstancePoolCreatorId
- spotinst:aws:ec2:group:id
- aws:ec2:fleet-id
- gitlab_autoscaler_token

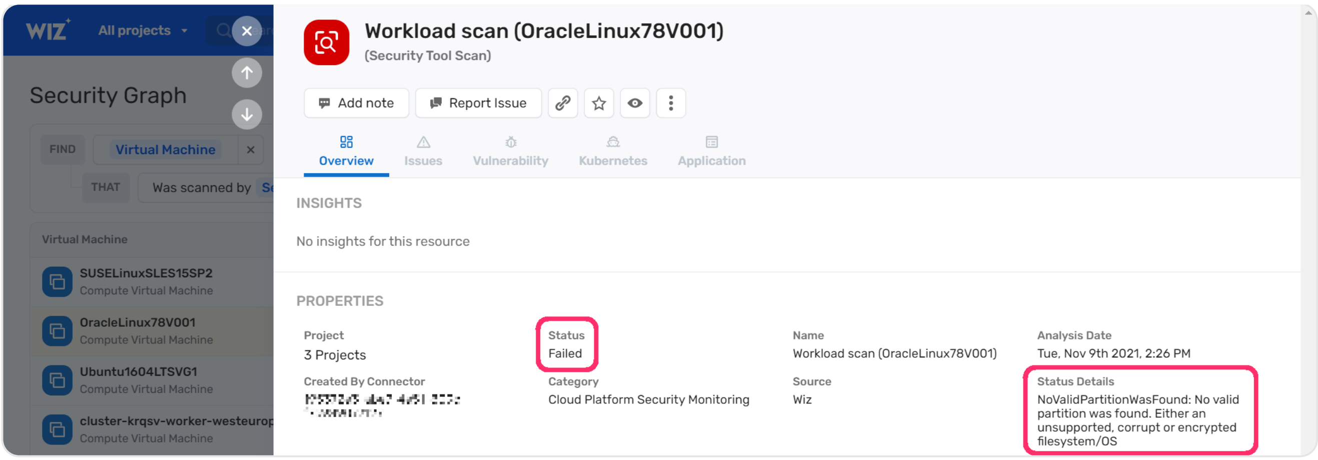
How does Wiz identify an ephemeral resource?

Because there is no single attribute across all cloud providers that identifies a resource as ephemeral, Wiz uses an in-house dictionary that captures different types of short-lived resources. These various definitions are normalized on the Security Graph as the ephemeral property.

Why wasn't a disk scanned?

There are several reasons a disk scan can fail. To troubleshoot disk scan errors:

1. Identify all VMs whose disks failed to scan or were skipped.
2. Click a **Security Tool Scan**. Its details drawer opens on the right.



Workload scan (OracleLinux78V001)
(Security Tool Scan)

INSIGHTS

No insights for this resource

PROPERTIES

Project	Status	Name	Analysis Date
3 Projects	Failed	Workload scan (OracleLinux78V001)	Tue, Nov 9th 2021, 2:26 PM
Created By Connector	**Category**	**Source**	**Status Details**
[redacted]	Cloud Platform Security Monitoring	Wiz	NoValidPartitionWasFound: No valid partition was found. Either an unsupported, corrupt or encrypted filesystem/OS

3. Look up its **Status** and **Status Details**:

Status	Status Details	Description
Failed	Internal error	The Wiz backend failed to complete the scan.
Failed	Missing permissions: *	Either the Wiz connector missing the required permission, or there exists a policy that blocks Wiz from accessing a required resource.
Failed	Missing Key Vault permissions to read the ADE encrypted disk secret	Follow the guide to grant permissions to the specified Key Vault.
Failed	NoValidPartitionWasFound: No valid partition was found. Either an unsupported, corrupt or encrypted filesystem/OS	No valid partition was found. Either an unsupported, corrupt or encrypted filesystem/OS.
Failed	Unexpected error	The Wiz backend failed to complete the scan.
Failed	VolumeIsFull: Unable to attach full volume	To ensure that the attached disk has the correct volume to scan, Wiz writes to that volume scan-related identifiers. Therefore, Wiz cannot scan volumes that are completely full.
Failed	VolumeIsReadOnly: Unable to operate on read-only volume	Wiz is unable to scan read-only volumes as it requires writing scan-related metadata.
Skipped	Databricks	Wiz does not scan Databricks.
Skipped	Instance group sampling	Instead of scanning all disk volumes in an instance group, Wiz samples only one.

Status	Status Details	Description
Skipped	Locked by: *	The specified resource group is locked, which would prevent Wiz deleting snapshots. To fix this, either remove the lock from the specified resource group or use the Dedicated Resource Group option in the Connector settings to set a dedicated resource group in which snapshots will be created, scanned and then deleted by Wiz.
Skipped	Secret External ID: *	Access to the specified secret is blocked from Wiz, which is required to perform the scan on encrypted disks.
Skipped	Unmanaged disk	Legacy disk format that Wiz does not support.
Skipped	Volume contains tag "wiz"	Wiz does not scan resources it created in customers environment.
Skipped	Volume not found	VMs or ephemeral resources that existed when Wiz intiated the scan but were destroyed later when Wiz tried to create a snapshot.

Why does Wiz need access to storage in order to scan functions in Azure?

In order to scan functions for vulnerabilities, Wiz needs access to the function code and dependencies. In Azure, App Services code is saved in a zip in a dedicated storage account, so Wiz requires access to the code files in those specific storage accounts. There is no other way in Azure to get the function code—it is always in a storage account. For some application types (e.g. Python apps), the code can be quite a lot of files, which can trigger alerts.

To access these specific storage accounts, Wiz uses the keys for the dedicated storage accounts containing service app code files. This access is granted by the following permissions:

- “Microsoft.Web/sites/config/list/Action”
- “Microsoft.Web/sites/slots/config/list/Action”

It is important to emphasize that Wiz does not have permissions to read data from other storage accounts, only to the storage used for function code. You can review our permissions to validate that Wiz does not have generic access to storage data. You can also see in your cloud logs exactly which storage accounts Wiz accesses.

Finally, when Wiz scans the function code for vulnerabilities, dependencies, etc., the scan is performed ad-hoc in memory. We do not store the code files anywhere. Wiz only ends up storing the metadata, i.e. the vulnerabilities.

Can specific workloads be excluded from scanning?

Yes. In order to exclude a workload from being scanned, add the following tag to the machine:

- key: "WizExclude"
- value: null

If you would like to use a tag other than "WizExclude", let us know.

How often does Wiz scan?

By default, your entire environment is scanned every ~24 hours. You can initiate manual scans on an individual resource from its details drawer, on a subscription, or on a connector.

Can scans be scheduled to start or end at a specific time of the day?

No. Wiz automatically scans your entire environment every ~24 hours, but these scans are staggered and optimized to avoid overloading the Wiz backend. This means there is some inherent variability around when exactly each account, subscription, VM, etc. was last scanned and will be scanned again.

You can always manually initiate scans of a specific VM or connector.

How many snapshots does Wiz scan concurrently?

Up to 20 disk snapshots are scanned concurrently.

Does Wiz support scanning physically attached disks (e.g. AWS instance store volumes)?

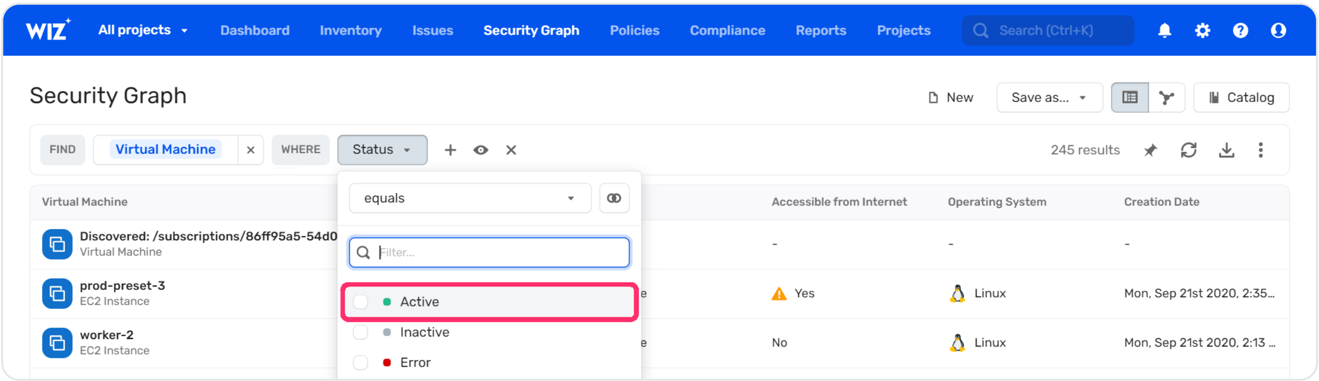
No. Physically attached disks do not support snapshots, so Wiz cannot scan them.

Does Wiz scan inactive VMs?

Absolutely. Inactive VMs are no less risky than active VMs for a number of reasons:

- A VM can be turned on very easily. It's better to know about its risks before it is turned on.
- VMs can be turned off for good security reasons, but you still want to know about their potential risks.
- A VM that was safe when it was turned off can become unsafe due to newly disclosed vulnerabilities, other infrastructure changes, etc.
- With sufficient permissions, malicious actors can identify vulnerable inactive VMs in much the same way Wiz does, turn them on, and then move laterally through your environment, escalate privileges, or gain access to sensitive data.

If you want to focus only on active VMs, you can always click **+** > **Status** > **Active** on the VM criterion in the query builder:



Security Graph

FIND Virtual Machine WHERE Status

equals

- Active
- Inactive
- Error

Virtual Machine		Accessible from Internet	Operating System	Creation Date
Discovered: /subscriptions/86ff95a5-54d0 Virtual Machine		-	-	-
prod-preset-3 EC2 Instance	e	Yes	Linux	Mon, Sep 21st 2020, 2:35...
worker-2 EC2 Instance	e	No	Linux	Mon, Sep 21st 2020, 2:13 ...

Updated about 1 hour ago

← Connect

Supported Cloud Services →

Did this page help you? Yes No

EXHIBIT 5

AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

PRT254

SPONSORED BY WIZ

Context is everything: Join the CNAPP revolution to secure your AWS deployments

Yinon Costica

Co-Founder and VP of Product
Wiz

Agenda

What is holding back your cloud journey?

A new approach to reducing risk

Fireside chat with John Visneski, CISO of MGM Studios

Open Q&A

Wiz: Redefining cloud security

One of the fastest-growing SaaS companies in history*

5M workloads

Scanned daily

Over 30%

of the Fortune 100

aws PARTNER

- Marketplace Seller
- Security Software Competency

Industry leaders trust Wiz as their cloud security partner

MARS
COLGATE-PALMOLIVE
Takeda
salesforce
BMW
REI co-op
LVMH
priceline
FOX
Blackstone
PerkinElmer
AVERY DENNISON
asos
snowflake
yotpo.
redis
PLAID
BRIDGEWATER
agoda
DocuSign
ManpowerGroup
THIRTY MADISON
Salesloft.
UiPath™

*Forbes Article by Peter Cohan, https://www.forbes.com/sites/petercohan/2021/10/30/outpacing-palo-alto-networks-wizs-valuation-soars-11000-in-10-months/?sh=2e2a725463d2

aws

What is holding back your cloud security journey?

Complex environment

Multiple clouds

Multiple architectures

Thousands of technologies

Complex risk

Vulnerabilities & misconfigurations

Risk likelihood

Risk impact

Complex to operationalize

Lack of visibility

Fragmented suite of tools

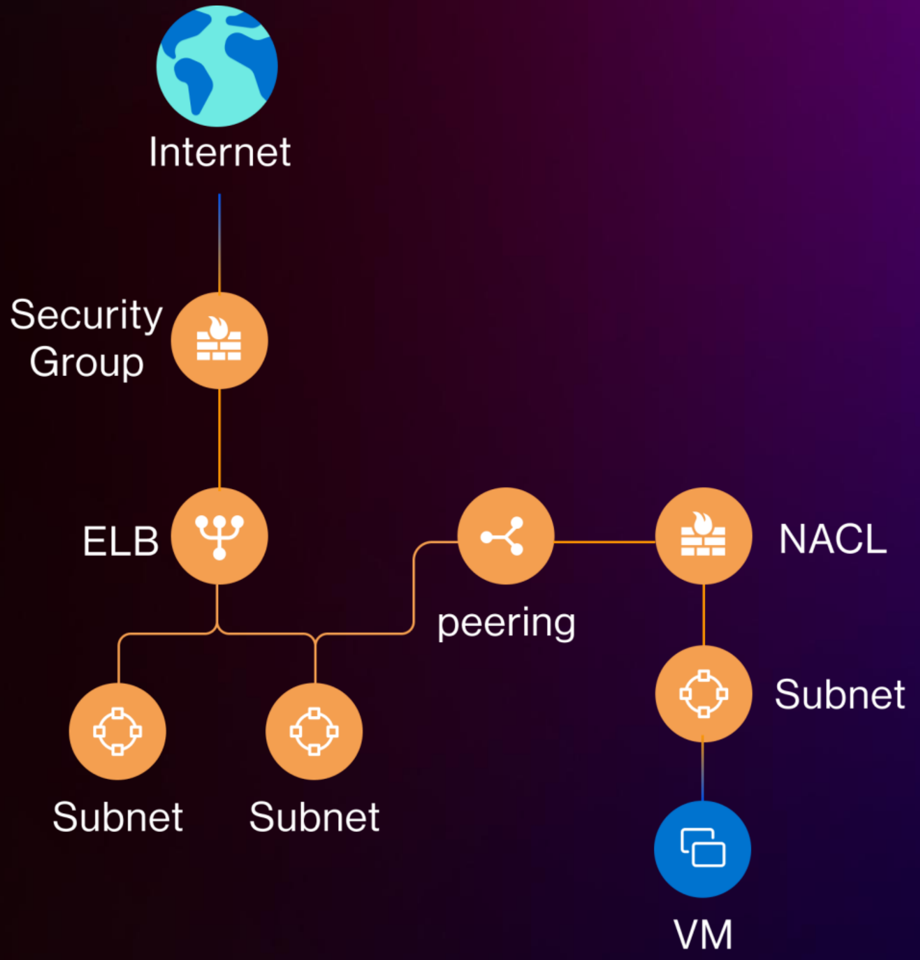
Remediation process across teams

Traditional security does not always detect the real issues

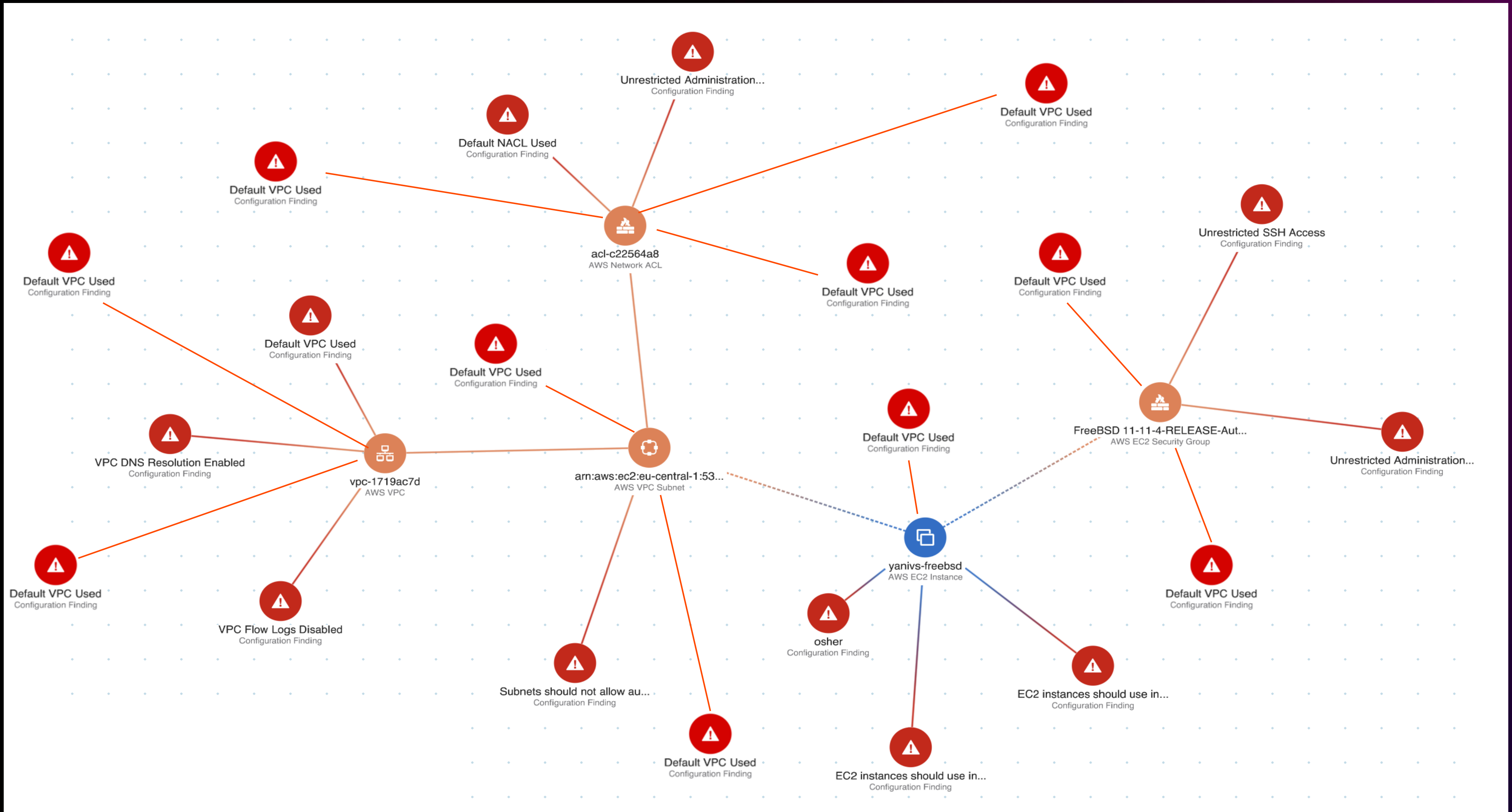
- Traditional tools like CSPMs and vulnerability scanners don't always detect the real issues

- Scan independently and lack context, unable to prioritize

- Generate noise and may increase your key security metrics



Misconfigurations != risks



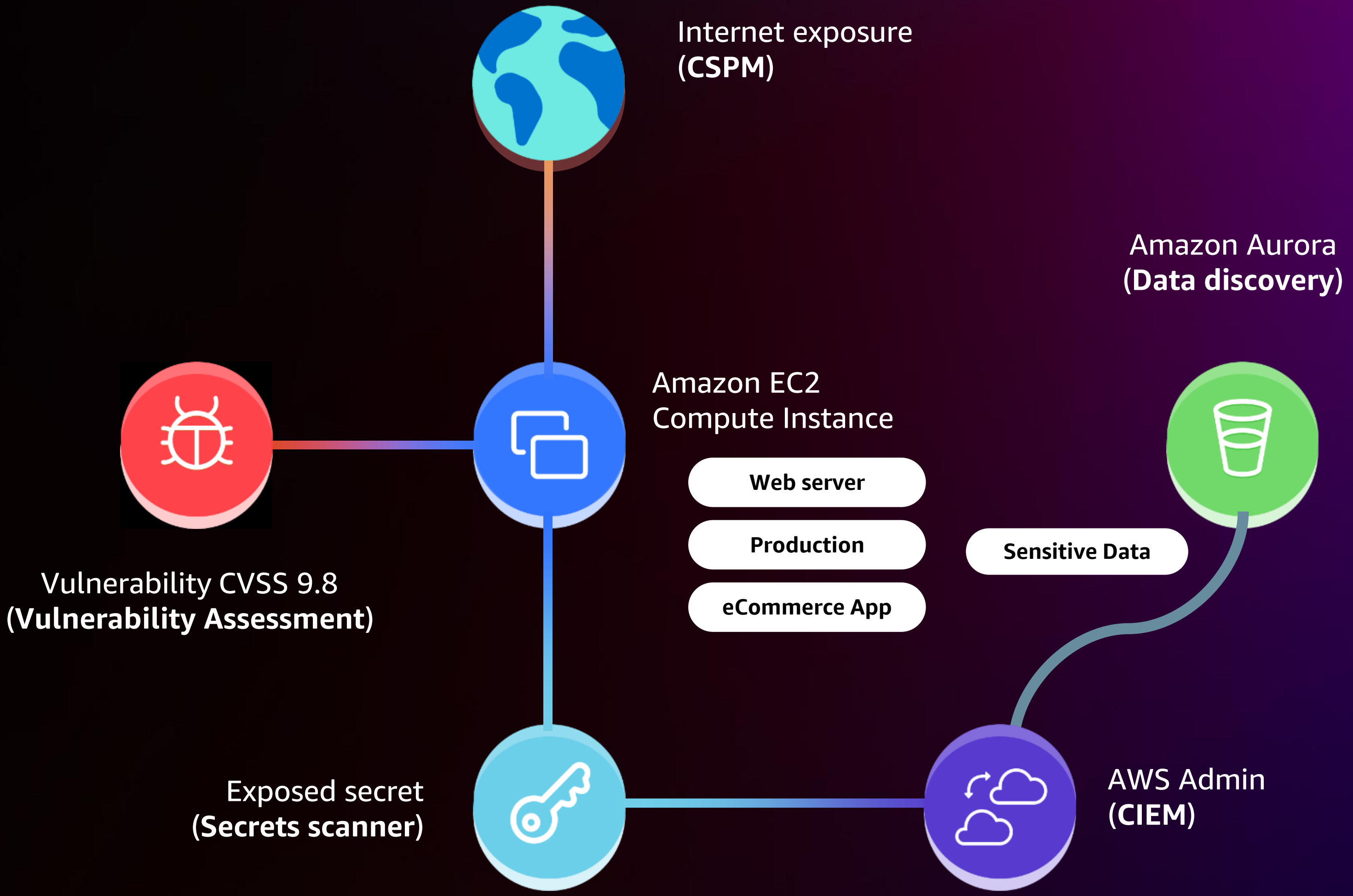
Example: Which security issues do you prioritize?

- Traditional security tooling drive alert fatigue
- Your team lacks the context to focus on the right issue
  - Workload
  - Cloud
  - Business context

Vulnerability CVSS 9.8
(**Vulnerability Assessment**)

Example: Which security issues do you prioritize?

- Traditional security tooling drive alert fatigue
- Your team lacks the context to focus on the right issue
  - Workload
  - Cloud
  - Business context



Internet exposure
(CSPM)

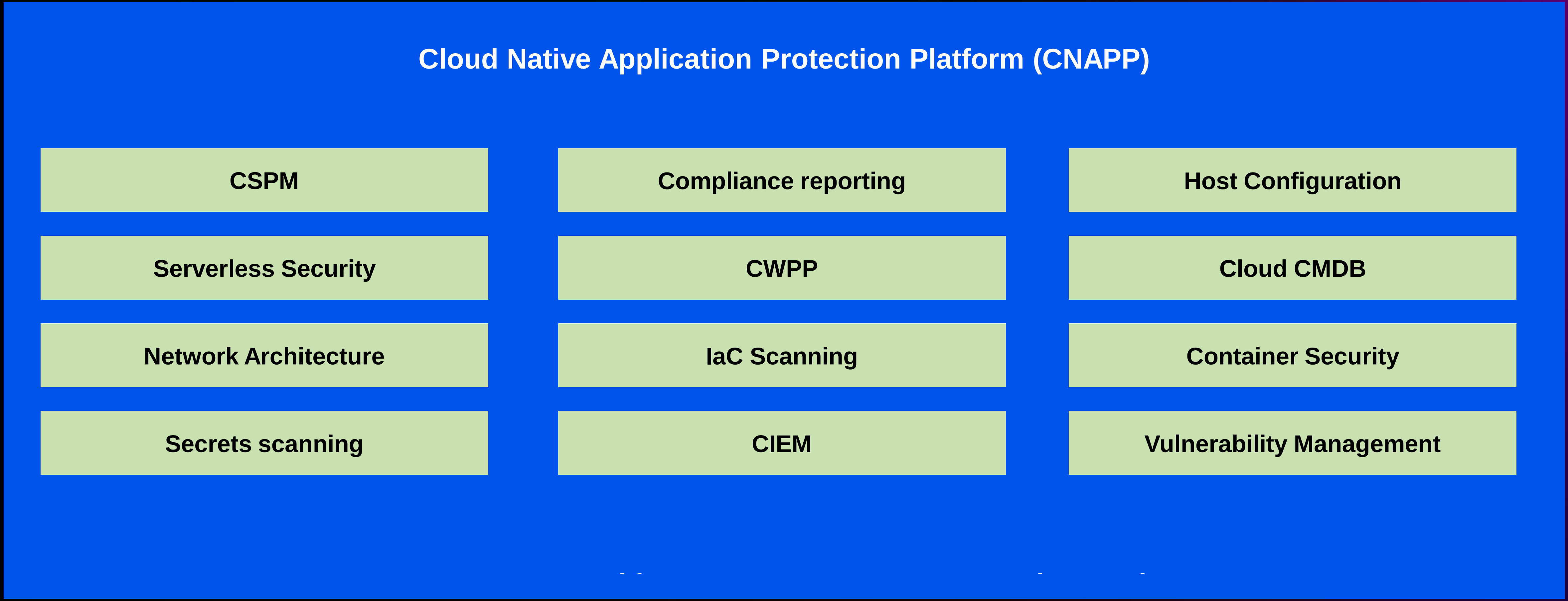
Amazon Aurora
(Data discovery)

Amazon EC2
Compute Instance

Vulnerability CVSS 9.8
(Vulnerability Assessment)

Exposed secret
(Secrets scanner)

AWS Admin
(CIEM)



Cloud Native Application Protection Platform (CNAPP)

CSPM

Compliance reporting

Host Configuration

Serverless Security

CWPP

Cloud CMDB

Network Architecture

IaC Scanning

Container Security

Secrets scanning

CIEM

Vulnerability Management

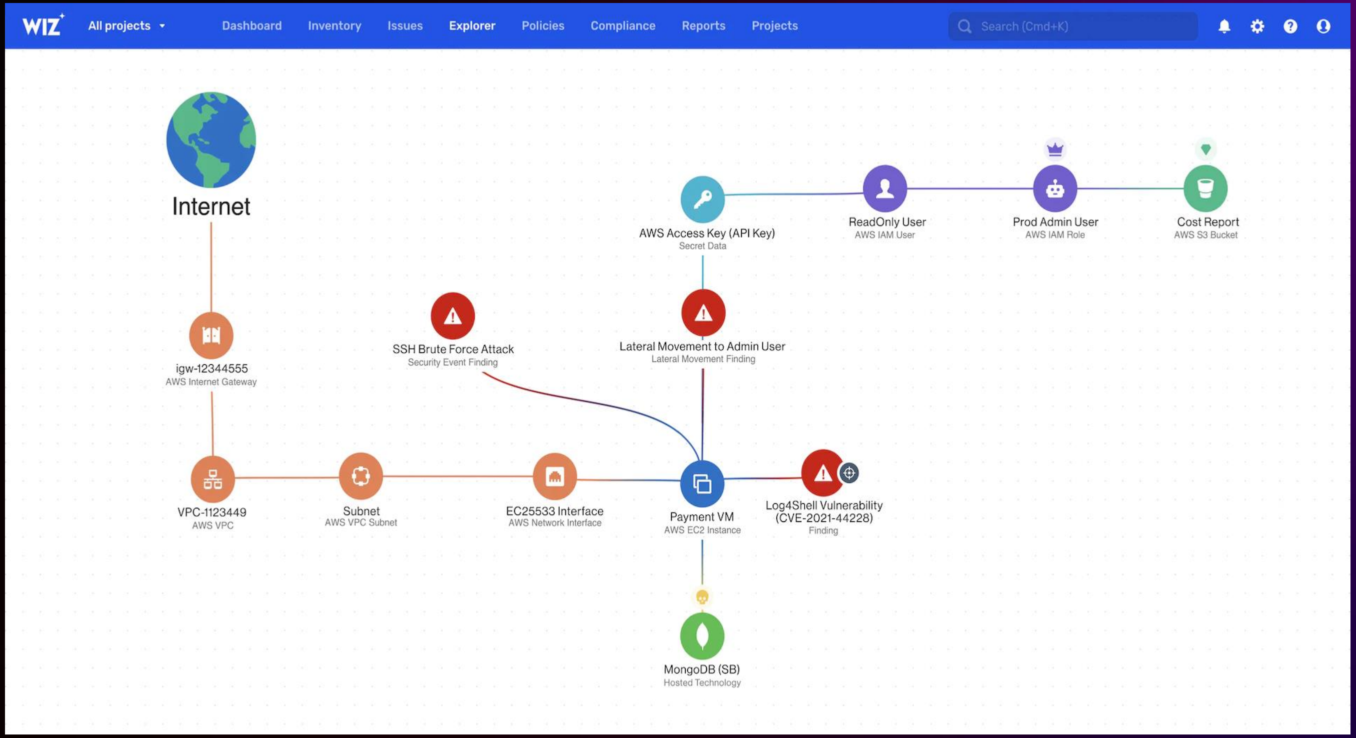
Wiz is a new approach to securing your cloud

Agentless, graph-based security solution to give you a comprehensive understanding of your AWS resources and risk

Frictionless visibility

Prioritization and context

Democratized for agility



Securing your AWS environment in 4 steps

1 Agentless scan of cloud metadata and workloads

Serverless
Containers
VMs
PaaS

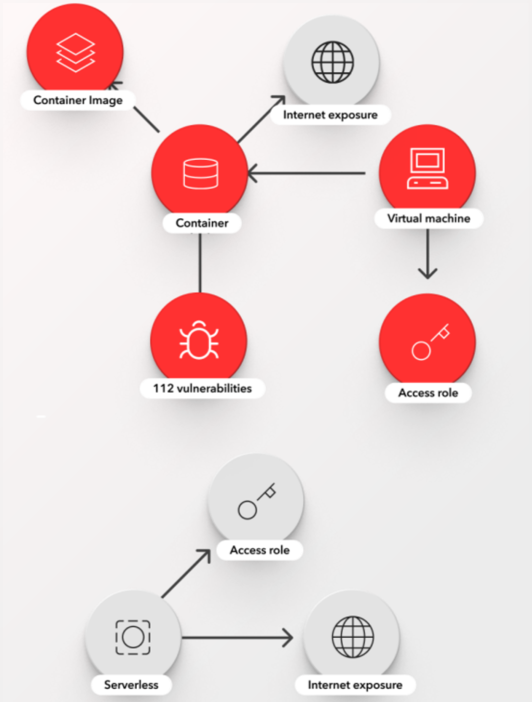
2 Perform a deep cloud assessment

Traditional scanning
Vulnerabilities and patches
Misconfigurations
Malware
Sensitive data

Cloud risk engine
External exposure
Excessive permissions
Exposed secrets
Lateral movement paths

Integrated tools
AWS CloudTrail
Amazon GuardDuty
Amazon Macie
3rd party tools

3 Identify the most critical risks



4 Proactively harden your cloud

Partner integrations

20+ integrations

Cloud remediation
One-click remediation
Automated security response
Remediation guidance

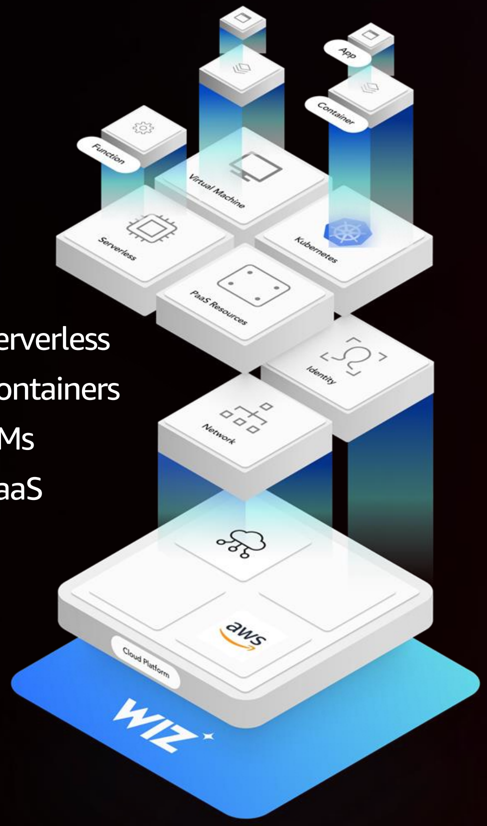
CI/CD Guardrails
One policy across the stack
Container and VM image scan
IaC scanning

Step 1: Full visibility in minutes across 60+ AWS services without agents

1 Agentless scan of cloud metadata and workloads

Frictionless visibility

- ✓ Agentless scanning via API
- ✓ Cloud and architecture agnostic
- ✓ Quick deployment, low maintenance



Serverless
Containers
VMs
PaaS

Compute

- Amazon EC2
- Amazon EKS
- AWS Fargate
- AWS Lamba function
- Amazon ECS
- AWS Transit Gateway
- Amazon VPC
- Amazon Route 53
- Elastic Load Balancer
- Amazon ECR

Security and Identity

- Amazon Cognito
- IAM
- AWS KMS
- AWS Secrets Manager
- Amazon GuardDuty
- AWS CloudTrail
- AWS Systems Manager

Application and Data

- Amazon ElastiCache
- Amazon S3
- Amazon Neptune
- Amazon Redshift
- Amazon DynamoDB
- Amazon RDS
- Amazon SNS
- Amazon SQS
- Amazon SageMaker
- AWS Glue
- MQ
- Amazon CloudFront

Step 2: See the whole risk picture with the Wiz Security Graph

2 Perform a deep cloud assessment

Traditional scanning

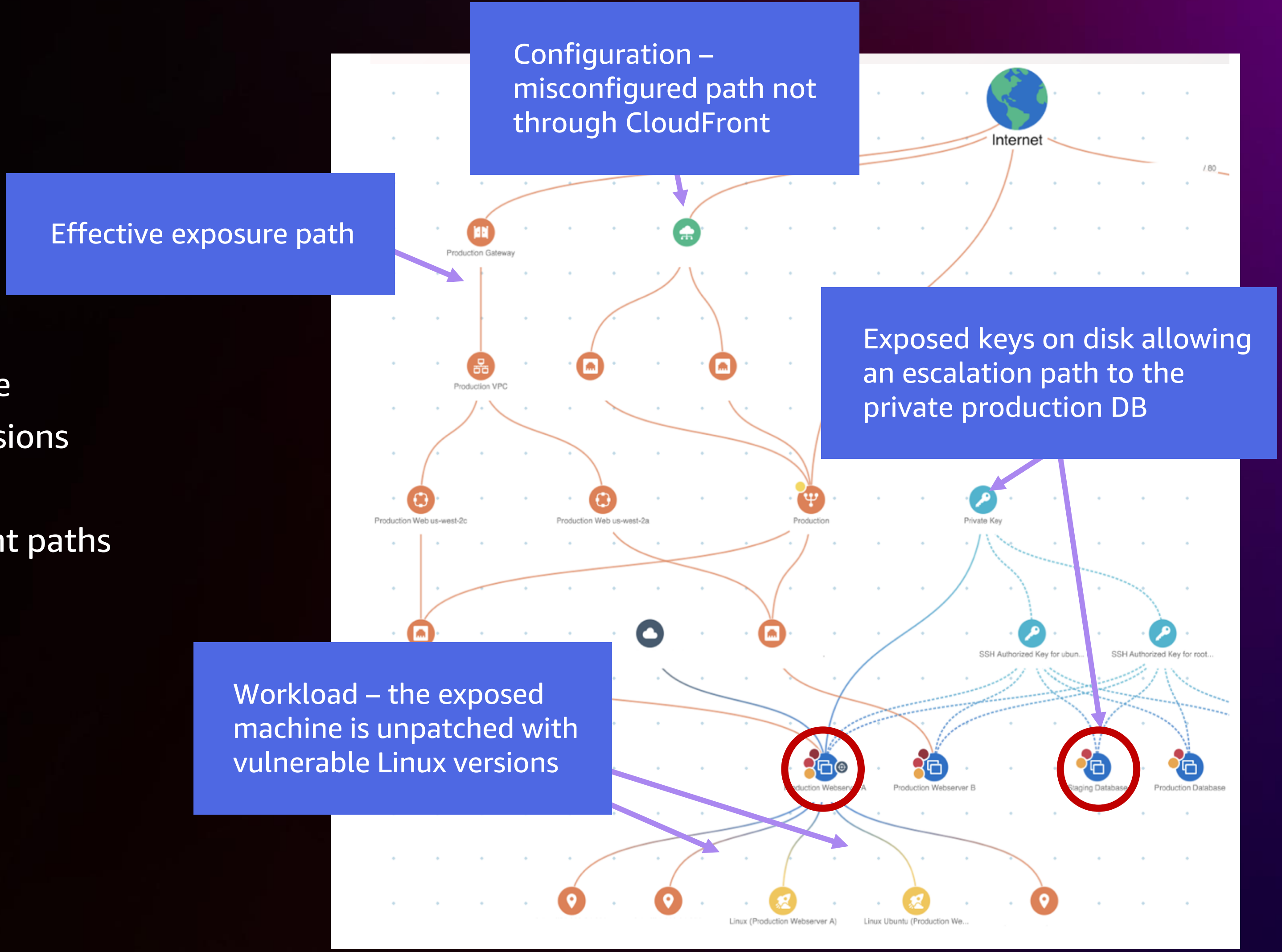
- Vulnerabilities and patches
- Misconfigurations
- Malware
- Sensitive data

Cloud risk engine

- External exposure
- Excessive permissions
- Exposed secrets
- Lateral movement paths

Integrated tools

- AWS CloudTrail
- Amazon GuardDuty
- Amazon Macie
- 3rd party tools



Configuration – misconfigured path not through CloudFront

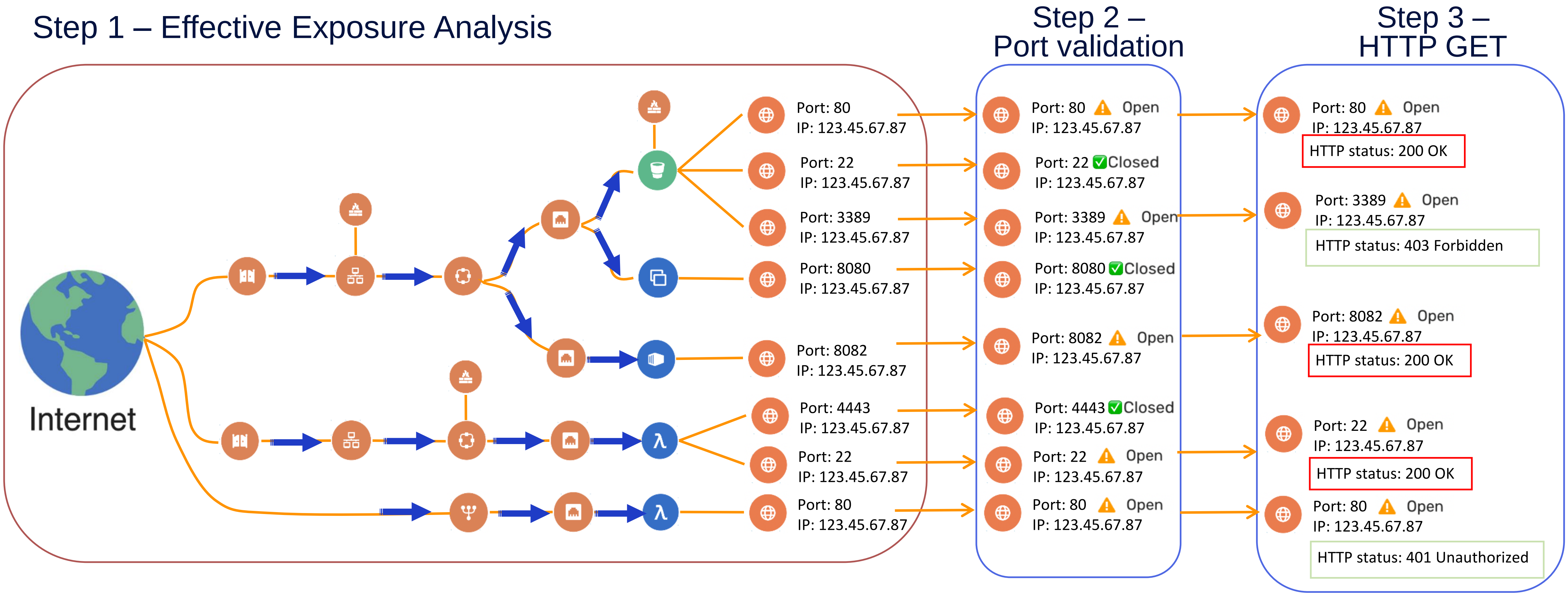
Effective exposure path

Exposed keys on disk allowing an escalation path to the private production DB

Workload – the exposed machine is unpatched with vulnerable Linux versions

Internet
Production Gateway
Production VPC
Production Web us-west-2c
Production Web us-west-2a
Production
Private Key
SSH Authorized Key for ubun...
SSH Authorized Key for root...
...duction Webserv... A
Production Webserver B
Staging Database
Production Database
Linux (Production Webserver A)
Linux Ubuntu (Production We...

Example: Effective network exposure



Step 1 – Effective Exposure Analysis

Port: 80
IP: 123.45.67.87

Port: 22
IP: 123.45.67.87

Port: 3389
IP: 123.45.67.87

Port: 8080
IP: 123.45.67.87

Port: 8082
IP: 123.45.67.87

Port: 4443
IP: 123.45.67.87

Port: 22
IP: 123.45.67.87

Port: 80
IP: 123.45.67.87

Step 2 –
Port validation

Port: 80 ⚠ Open
IP: 123.45.67.87

Port: 22 ✅Closed
IP: 123.45.67.87

Port: 3389 ⚠ Open
IP: 123.45.67.87

Port: 8080 ✅Closed
IP: 123.45.67.87

Port: 8082 ⚠ Open
IP: 123.45.67.87

Port: 4443 ✅Closed
IP: 123.45.67.87

Port: 22 ⚠ Open
IP: 123.45.67.87

Port: 80 ⚠ Open
IP: 123.45.67.87

Step 3 –
HTTP GET

Port: 80 ⚠ Open
IP: 123.45.67.87
HTTP status: 200 OK

Port: 3389 ⚠ Open
IP: 123.45.67.87
HTTP status: 403 Forbidden

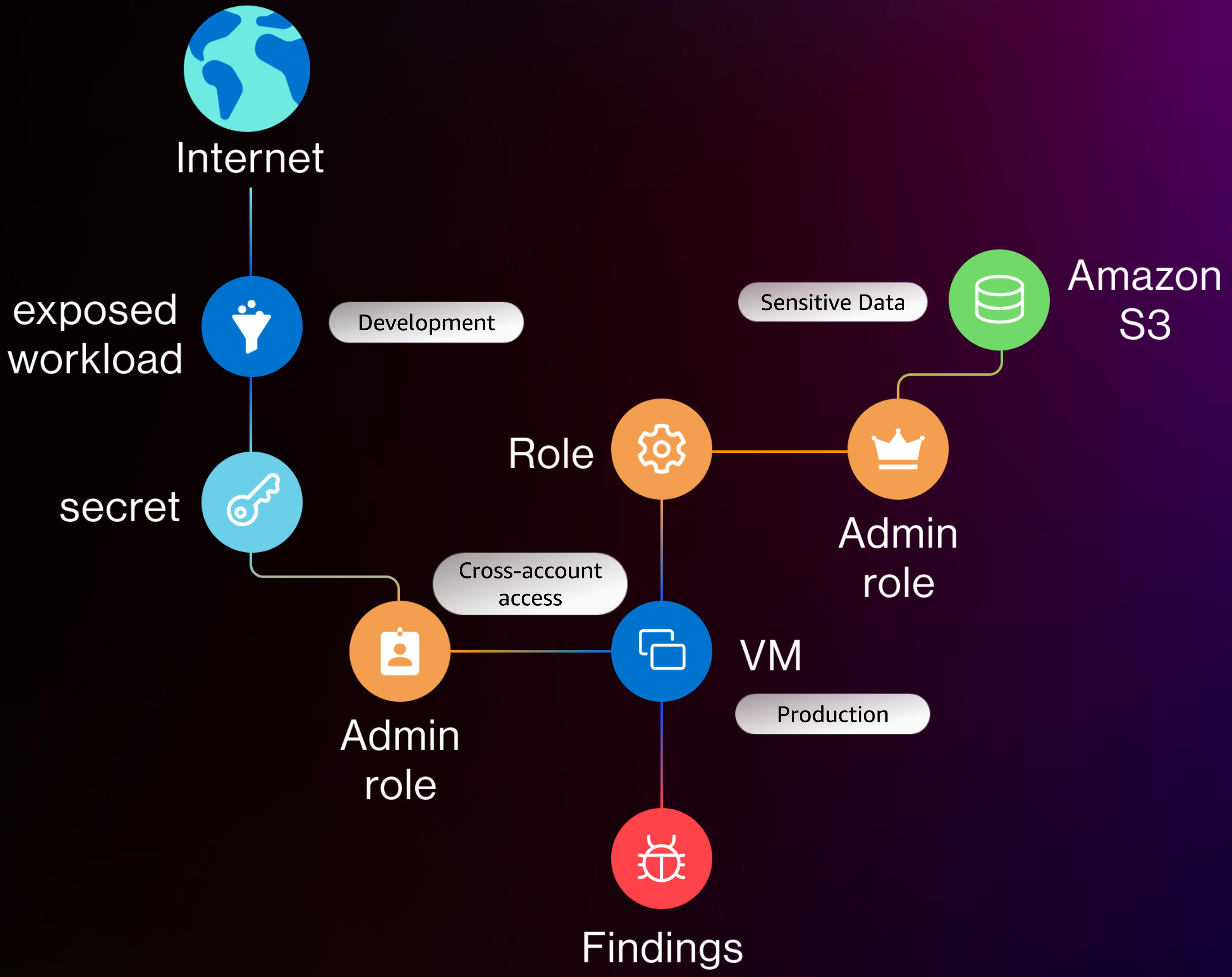
Port: 8082 ⚠ Open
IP: 123.45.67.87
HTTP status: 200 OK

Port: 22 ⚠ Open
IP: 123.45.67.87
HTTP status: 200 OK

Port: 80 ⚠ Open
IP: 123.45.67.87
HTTP status: 401 Unauthorized

Example: Escalation path

- Identify lateral movement and escalation paths using graph analysis
- Correlate exposed secrets, permissions, and the resources they can access



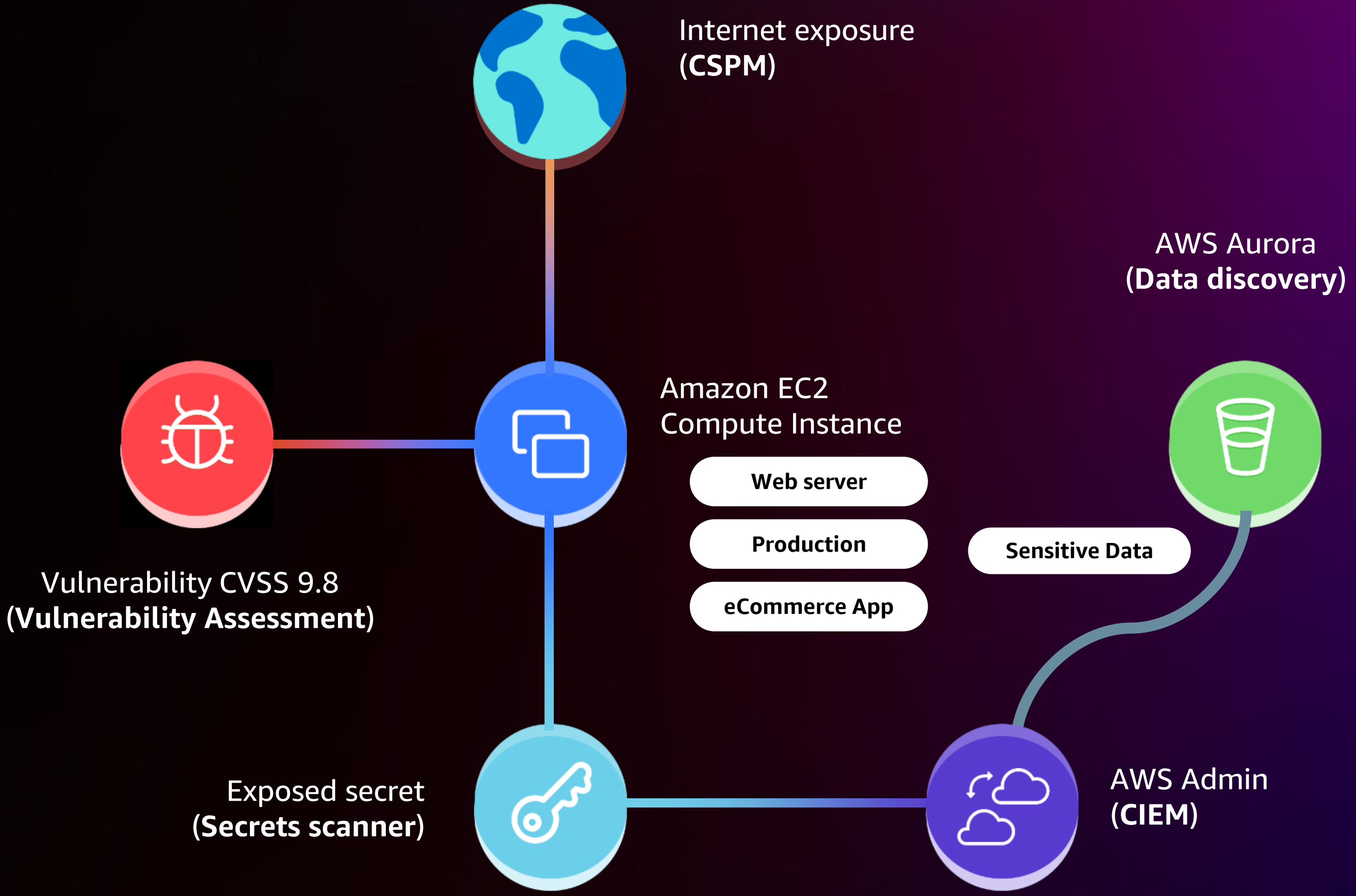
Step 3: Prioritize risk and attack paths to critical assets

3 Identify the most critical risks

⚠

Prioritization and context

- ✓ Security findings from Wiz and AWS
- ✓ Correlates risk from multiple security domains
- ✓ Provides meaningful context
- ✓ High fidelity alerting



Internet exposure
(CSPM)

AWS Aurora
(Data discovery)

Amazon EC2
Compute Instance

Web server

Production

eCommerce App

Sensitive Data

Vulnerability CVSS 9.8
(Vulnerability Assessment)

Exposed secret
(Secrets scanner)

AWS Admin
(CIEM)

Exposure

+

Vulnerability

+

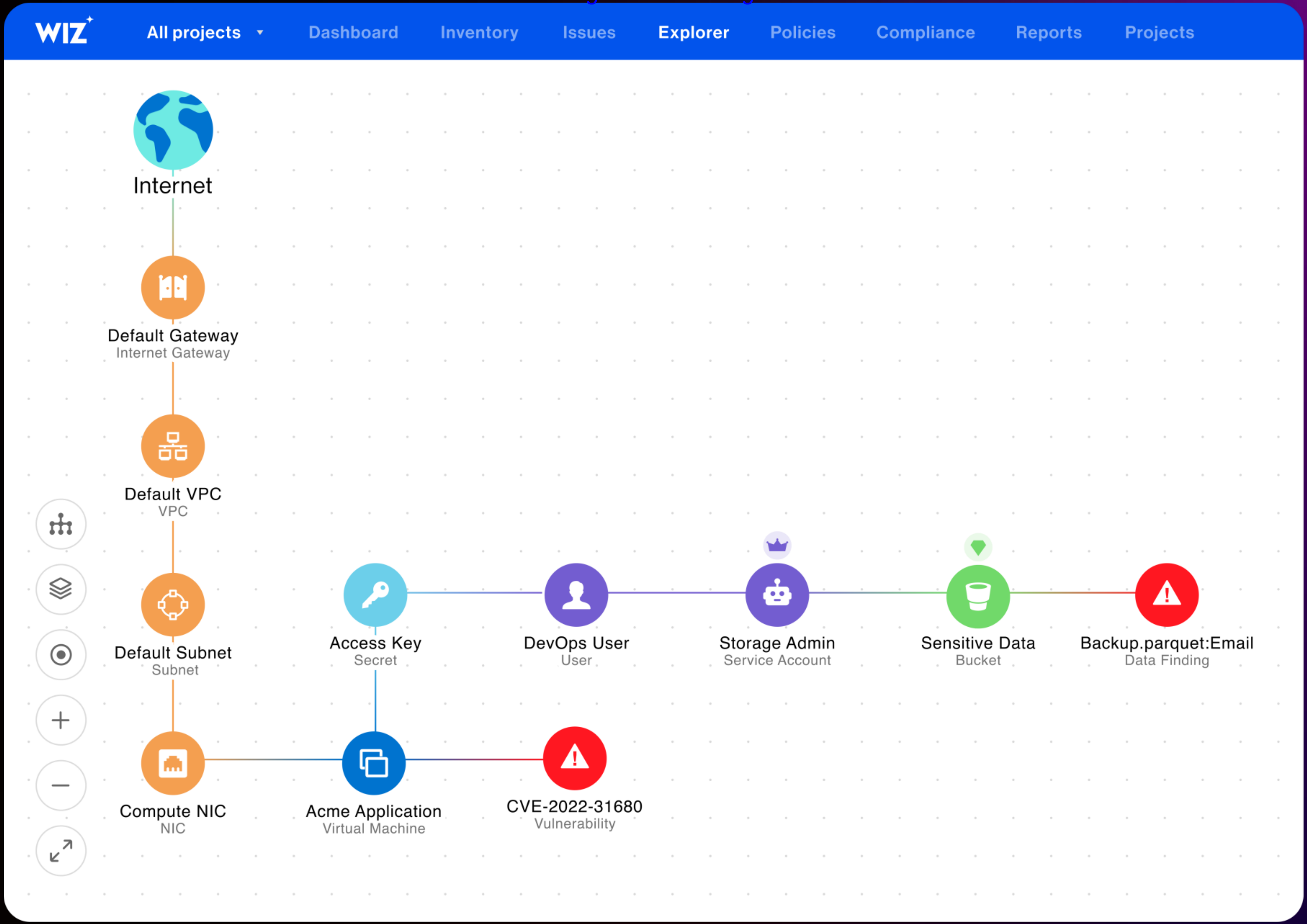
Secrets

+

Identity

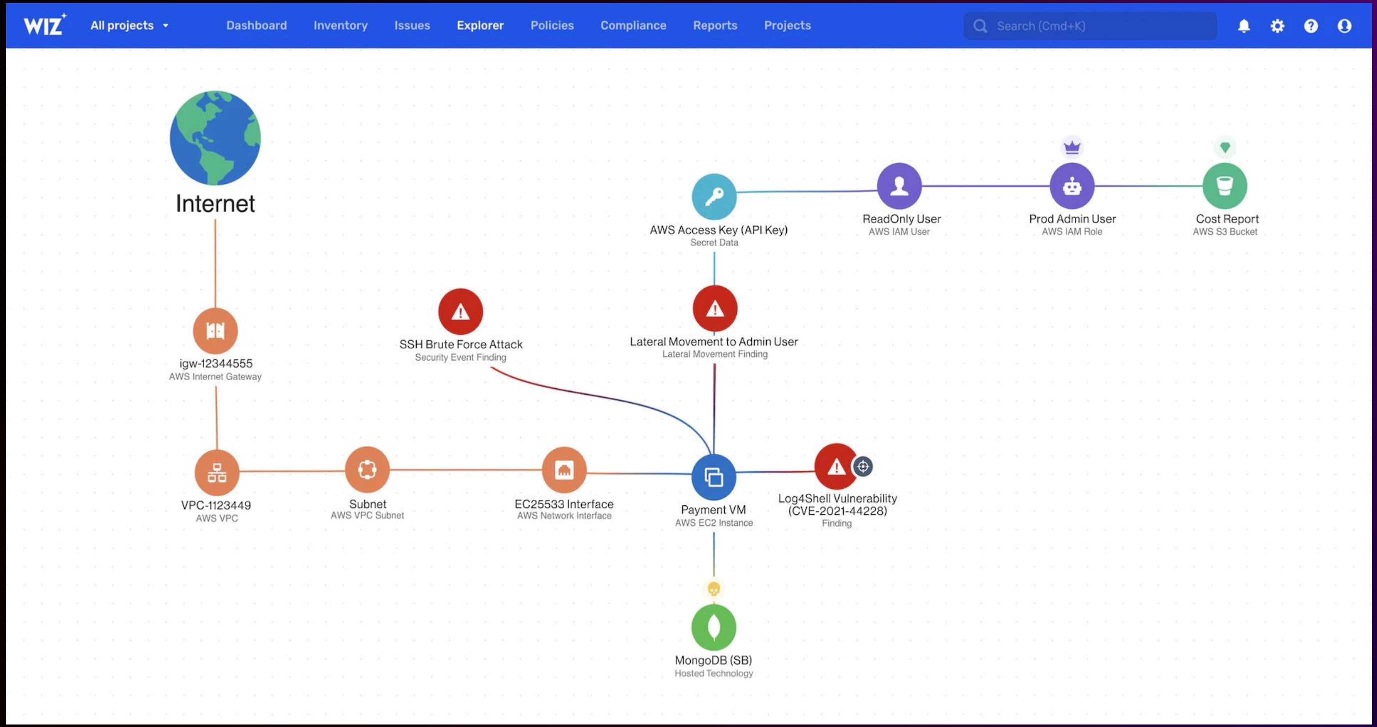
+

Data

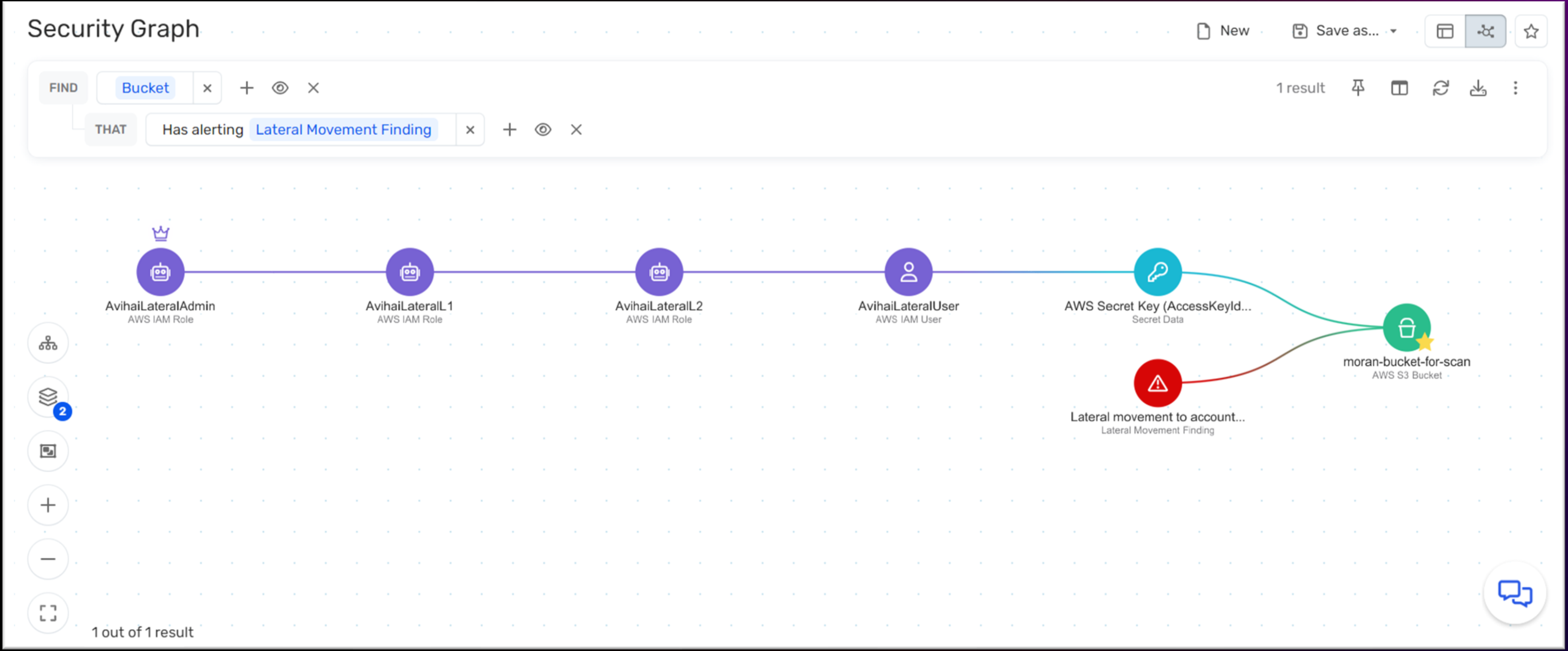


Amazon GuardDuty findings prioritization

- Cloud alerts lack context as well
- Contextualize and prioritize threat detection with the graph
- SSH brute force correlated to a:
  - High-risk workload
  - Password authentication
  - User with a weak password
  - Lateral movement path

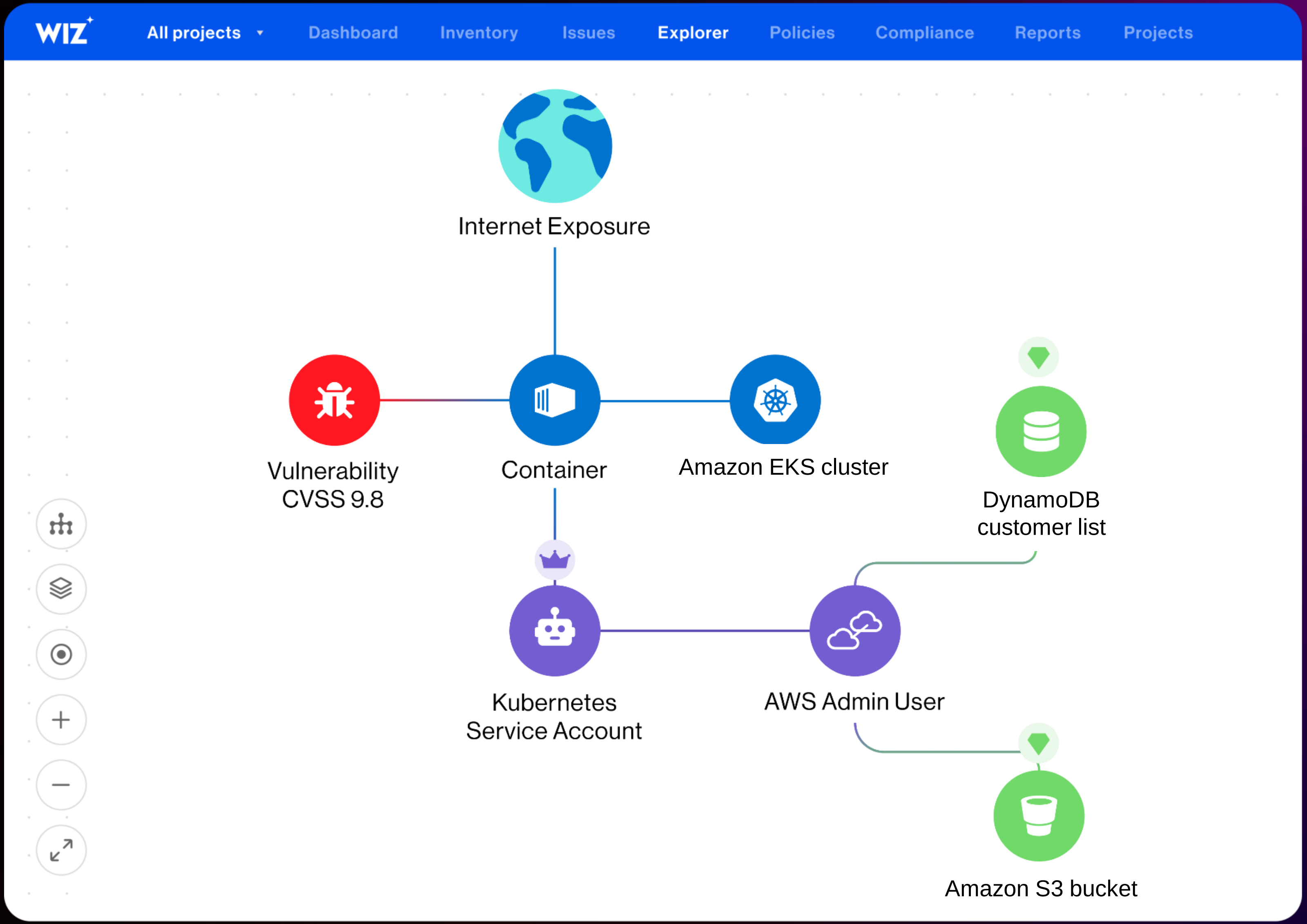


Secrets scanning in data assets



Kubernetes, containers, and serverless

- Network exposure
- Vulnerability scanning
- K8s to cloud identity
- Data scanning

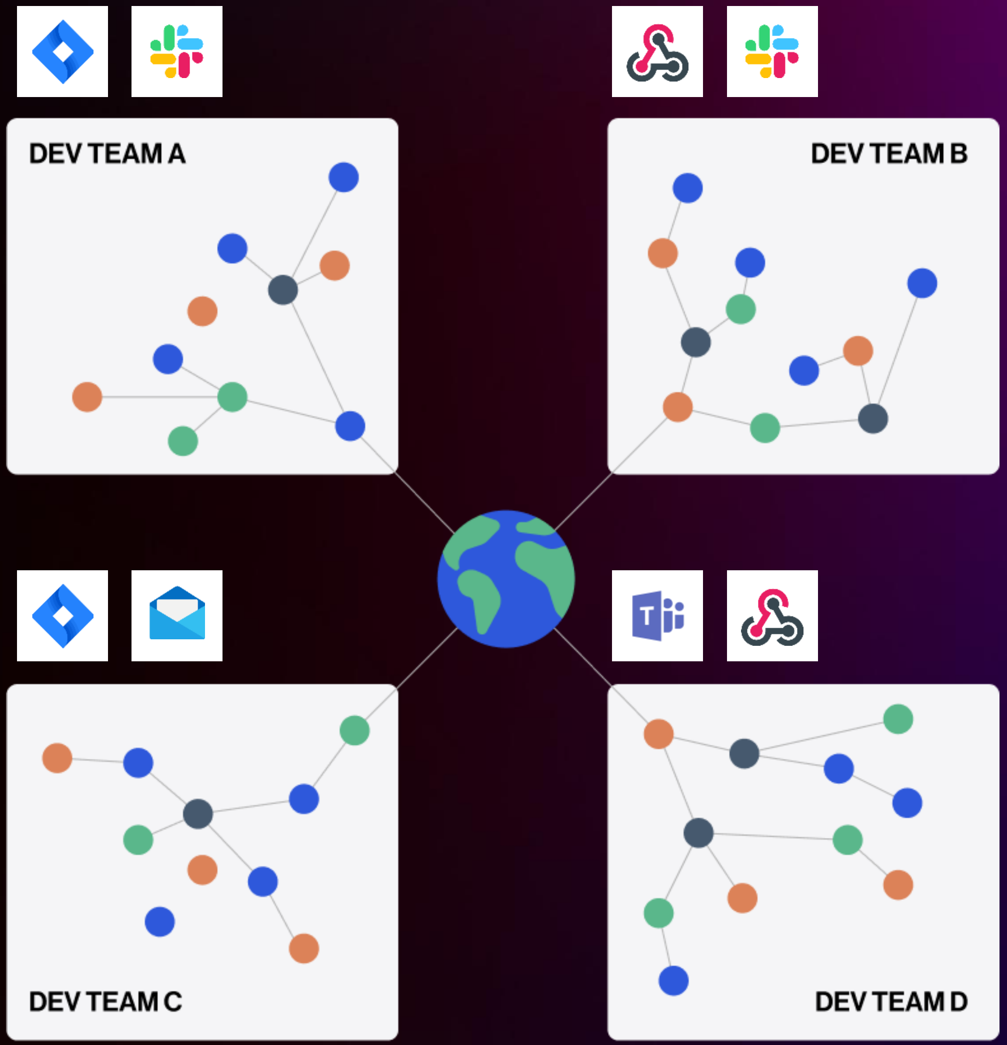


Step 4: Proactively harden your cloud

4 Proactively harden your cloud

Democratize security and build process

- ✓ Works with your developers' native tooling
- ✓ All cloud teams get access to their risks
- ✓ Automation to correctly route issues
- ✓ Remediation to the left and to the right



Wiz for AWS: Cloud Security for Builders and Defenders

Set up within minutes using a cloud role

✓ Connect with 50+ AWS Services

✓ Agentless scanning via API

✓ Cloud and architecture agnostic

✓ Quick deployment, low maintenance

Real risks, out of the box, no tunning required

✓ Security findings from Wiz and AWS

✓ Correlates toxic combinations

✓ Provides meaningful context

✓ High fidelity alerting

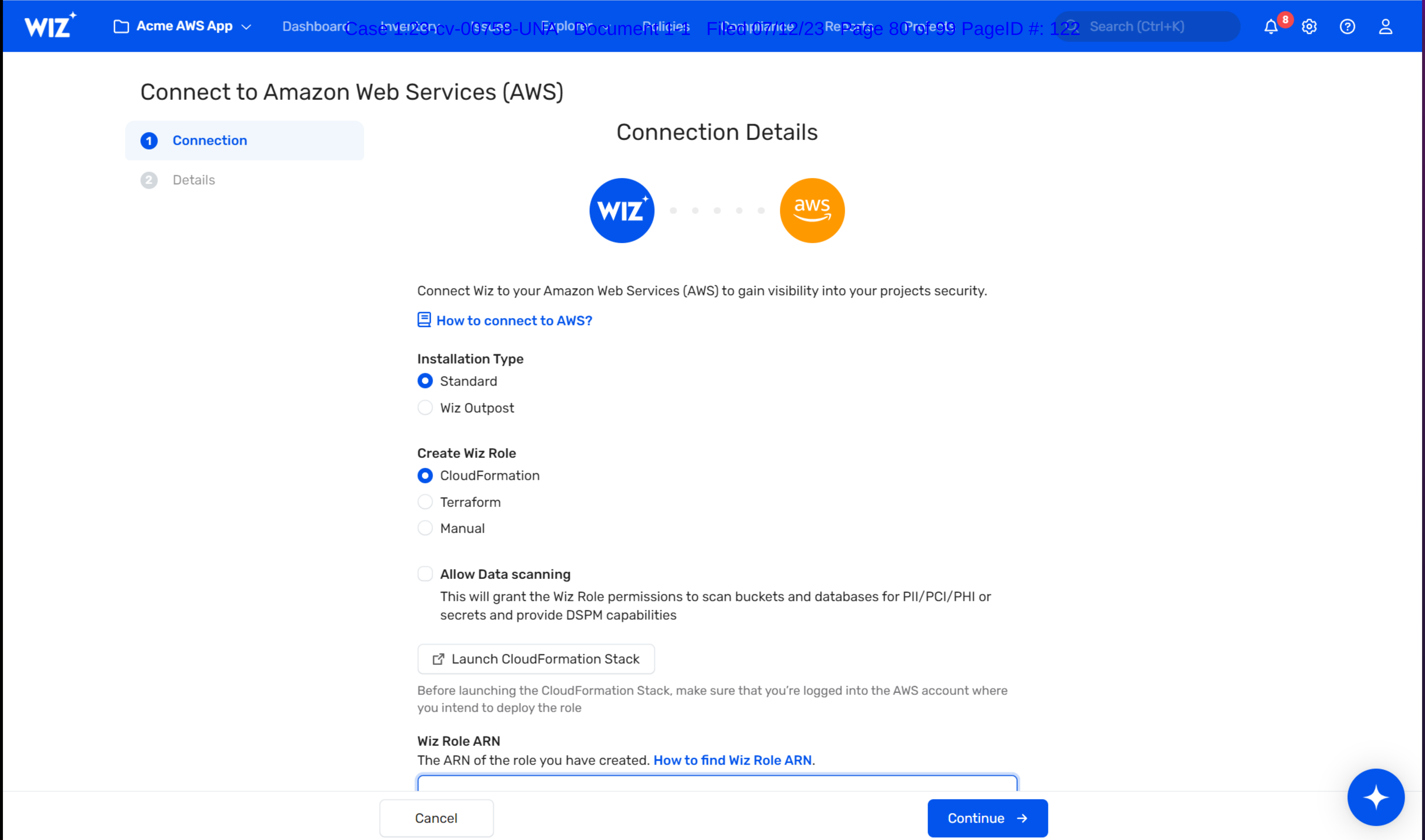
Democratize security and build a program

✓ Works with AWS Tooling

✓ All cloud teams get access to risk

✓ Automation for correctly routing issues

✓ Remediation to the left and to the right



Connect to Amazon Web Services (AWS)

1 Connection

2 Details

Connection Details

Connect Wiz to your Amazon Web Services (AWS) to gain visibility into your projects security.

How to connect to AWS?

Installation Type

● Standard

○ Wiz Outpost

Create Wiz Role

● CloudFormation

○ Terraform

○ Manual

□ **Allow Data scanning**

This will grant the Wiz Role permissions to scan buckets and databases for PII/PCI/PHI or secrets and provide DSPM capabilities

Launch CloudFormation Stack

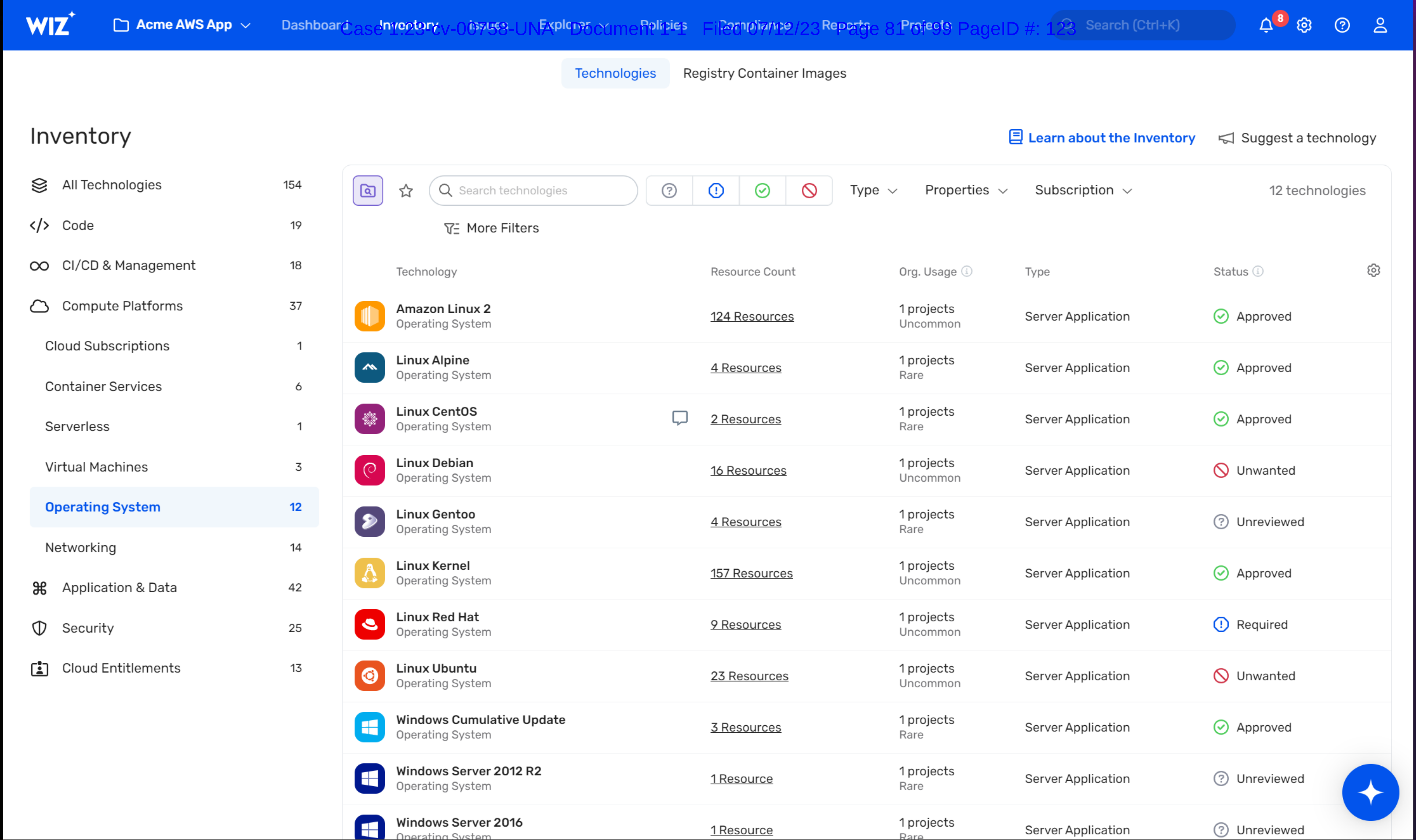
Before launching the CloudFormation Stack, make sure that you're logged into the AWS account where you intend to deploy the role

Wiz Role ARN

The ARN of the role you have created. How to find Wiz Role ARN.

Cancel

Continue →

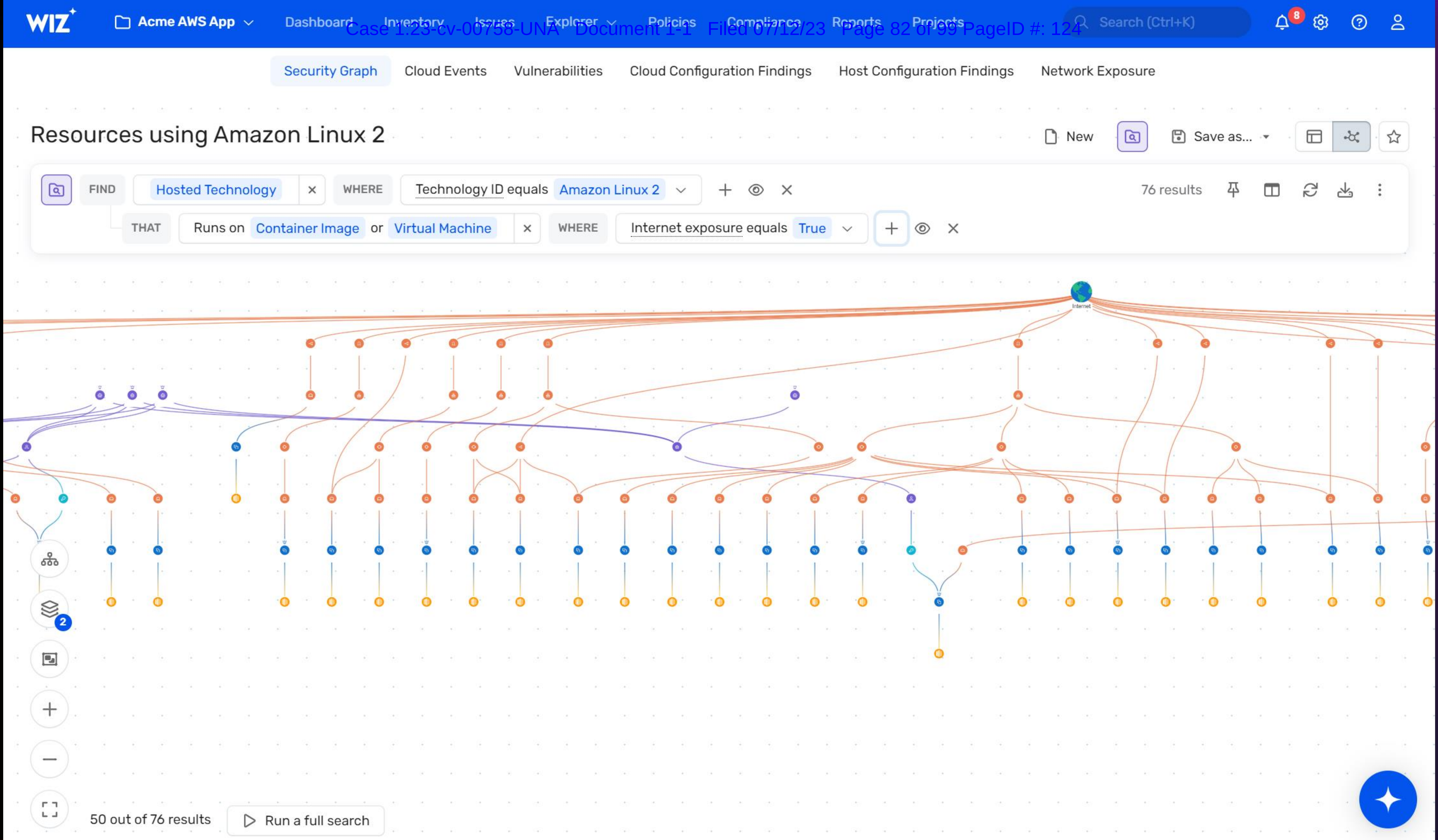


Inventory

All Technologies	154
Code	19
CI/CD & Management	18
Compute Platforms	37
Cloud Subscriptions	1
Container Services	6
Serverless	1
Virtual Machines	3
Operating System	12
Networking	14
Application & Data	42
Security	25
Cloud Entitlements	13

12 technologies

Technology	Resource Count	Org. Usage	Type	Status
Amazon Linux 2 Operating System	124 Resources	1 projects Uncommon	Server Application	Approved
Linux Alpine Operating System	4 Resources	1 projects Rare	Server Application	Approved
Linux CentOS Operating System	2 Resources	1 projects Rare	Server Application	Approved
Linux Debian Operating System	16 Resources	1 projects Uncommon	Server Application	Unwanted
Linux Gentoo Operating System	4 Resources	1 projects Rare	Server Application	Unreviewed
Linux Kernel Operating System	157 Resources	1 projects Uncommon	Server Application	Approved
Linux Red Hat Operating System	9 Resources	1 projects Uncommon	Server Application	Required
Linux Ubuntu Operating System	23 Resources	1 projects Uncommon	Server Application	Unwanted
Windows Cumulative Update Operating System	3 Resources	1 projects Rare	Server Application	Approved
Windows Server 2012 R2 Operating System	1 Resource	1 projects Rare	Server Application	Unreviewed
Windows Server 2016 Operating System	1 Resource	1 projects Rare	Server Application	Unreviewed



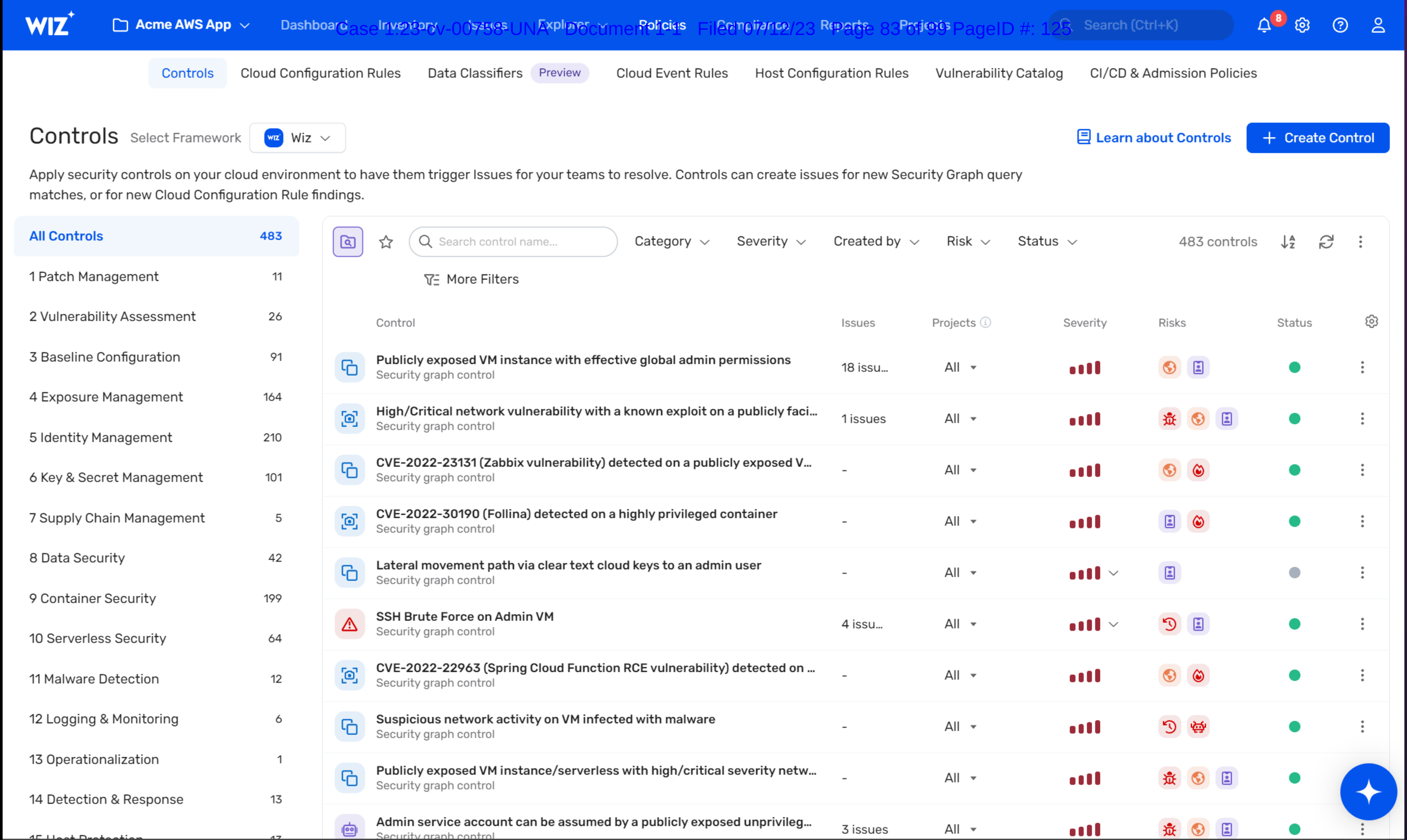
Resources using Amazon Linux 2

FIND Hosted Technology WHERE Technology ID equals Amazon Linux 2

THAT Runs on Container Image or Virtual Machine WHERE Internet exposure equals True

76 results

50 out of 76 results



Controls

Select Framework: Wiz

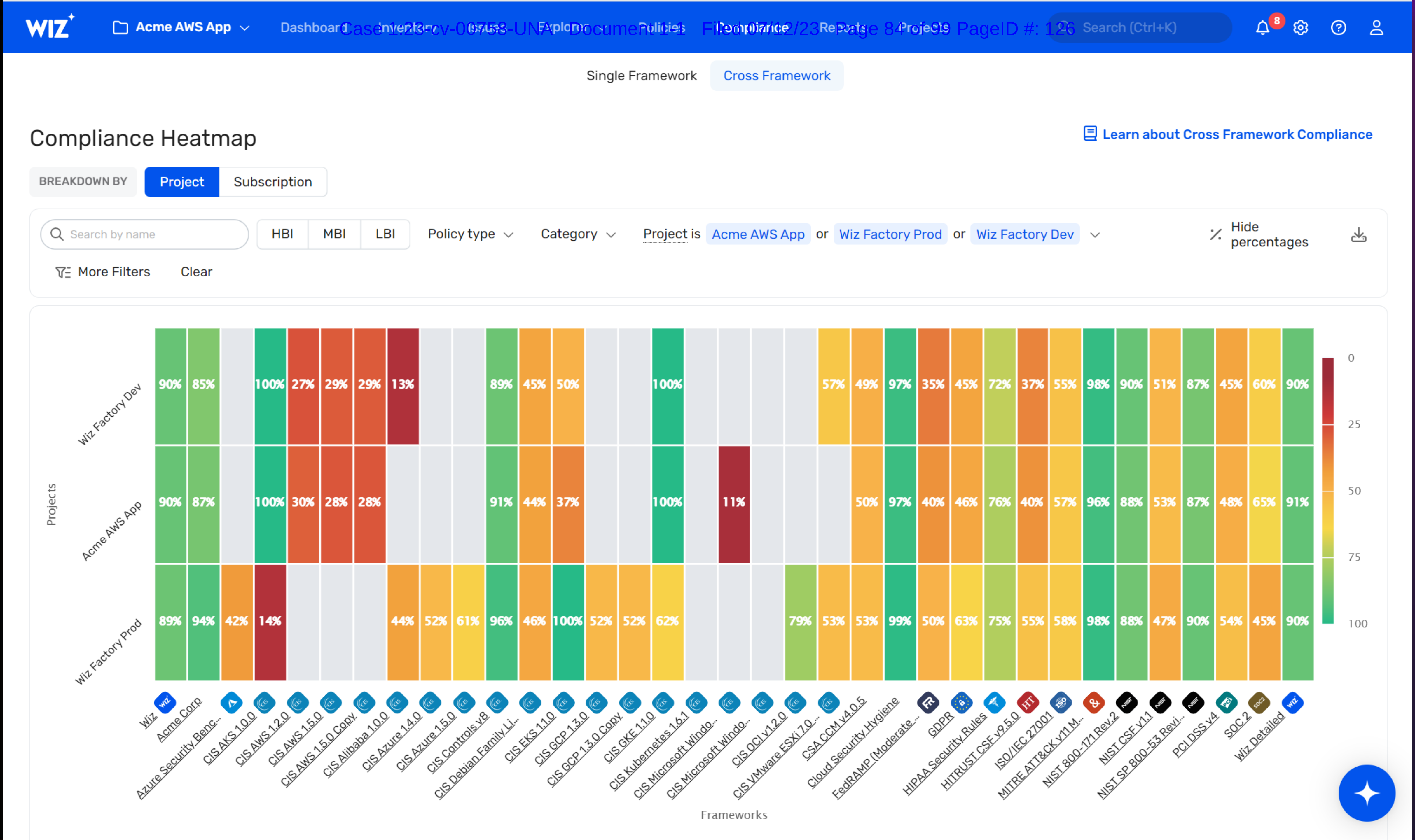
Learn about Controls

+ Create Control

Apply security controls on your cloud environment to have them trigger Issues for your teams to resolve. Controls can create issues for new Security Graph query matches, or for new Cloud Configuration Rule findings.

All Controls	483
1 Patch Management	11
2 Vulnerability Assessment	26
3 Baseline Configuration	91
4 Exposure Management	164
5 Identity Management	210
6 Key & Secret Management	101
7 Supply Chain Management	5
8 Data Security	42
9 Container Security	199
10 Serverless Security	64
11 Malware Detection	12
12 Logging & Monitoring	6
13 Operationalization	1
14 Detection & Response	13
15 Host Protection	13

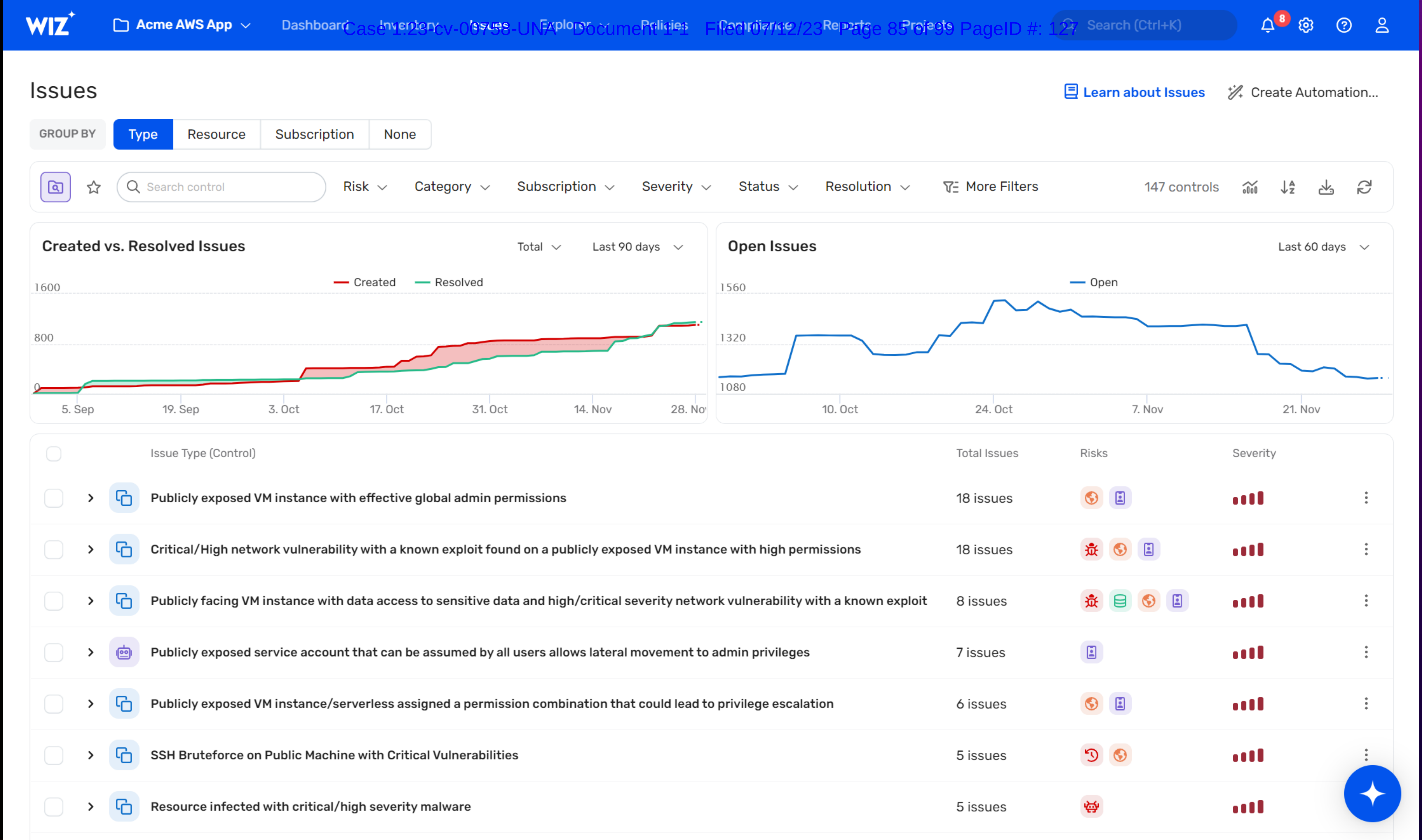
Control	Issues	Projects	Severity	Risks	Status
Publicly exposed VM instance with effective global admin permissions Security graph control	18 issu...	All			●
High/Critical network vulnerability with a known exploit on a publicly faci... Security graph control	1 issues	All			●
CVE-2022-23131 (Zabbix vulnerability) detected on a publicly exposed V... Security graph control	-	All			●
CVE-2022-30190 (Follina) detected on a highly privileged container Security graph control	-	All			●
Lateral movement path via clear text cloud keys to an admin user Security graph control	-	All			●
SSH Brute Force on Admin VM Security graph control	4 issu...	All			●
CVE-2022-22963 (Spring Cloud Function RCE vulnerability) detected on ... Security graph control	-	All			●
Suspicious network activity on VM infected with malware Security graph control	-	All			●
Publicly exposed VM instance/serverless with high/critical severity netw... Security graph control	-	All			●
Admin service account can be assumed by a publicly exposed unprivileg... Security graph control	3 issues	All			●



Compliance Heatmap

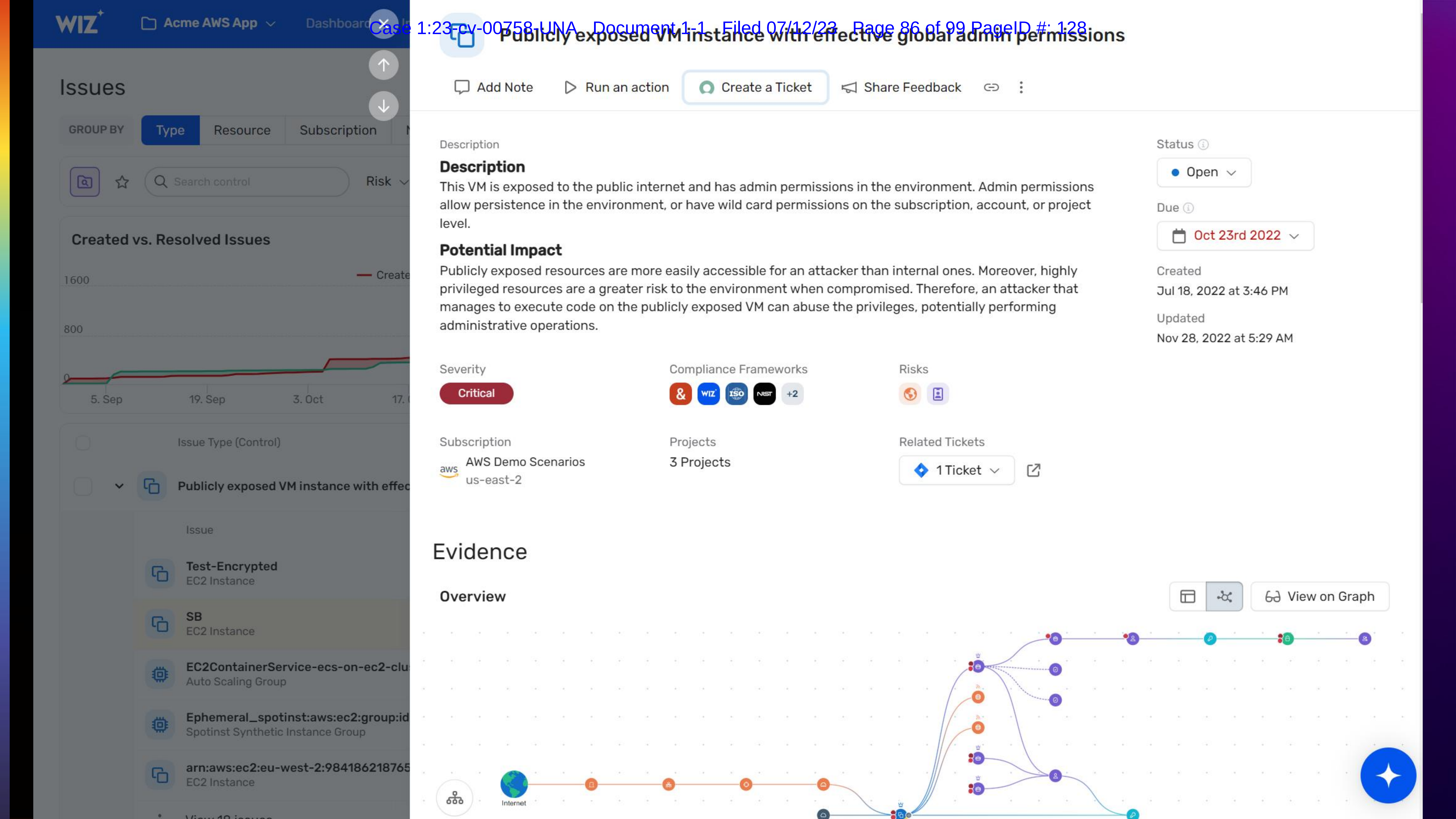
Project is Acme AWS App or Wiz Factory Prod or Wiz Factory Dev

Projects	Wiz	Acme Corp	Azure Security Benc...	CIS AKS 1.0.0	CIS AWS 1.2.0	CIS AWS 1.5.0	CIS AWS 1.5.0 Copy	CIS Alibaba 1.0.0	CIS Azure 1.4.0	CIS Azure 1.5.0	CIS Controls v8	CIS Debian Family Li...	CIS EKS 1.1.0	CIS GCP 1.3.0	CIS GCP 1.3.0 Copy	CIS GKE 1.1.0	CIS Kubernetes 1.6.1	CIS Microsoft Windo...	CIS Microsoft Windo...	CIS OCI v1.2.0	CIS VMware ESXi 7.0 ...	CSA CCM v4.0.5	Cloud Security Hygiene	FedRAMP (Moderate ...	GDPR	HIPAA Security Rules	HITRUST CSF v9.5.0	ISO/IEC 27001	MITRE ATT&CK v11 M...	NIST 800-171 Rev.2	NIST CSF v1.1	NIST SP 800-53 Revi...	PCI DSS v4	SOC 2	Wiz Detailed
Wiz Factory Dev	90%	85%		100%	27%	29%	29%	13%			89%	45%	50%			100%					57%	49%	97%	35%	45%	72%	37%	55%	98%	90%	51%	87%	45%	60%	90%
Acme AWS App	90%	87%		100%	30%	28%	28%				91%	44%	37%			100%		11%				50%	97%	40%	46%	76%	40%	57%	96%	88%	53%	87%	48%	65%	91%
Wiz Factory Prod	89%	94%	42%	14%				44%	52%	61%	96%	46%	100%	52%	52%	62%				79%	53%	53%	99%	50%	63%	75%	55%	58%	98%	88%	47%	90%	54%	45%	90%



Issues

Issue Type (Control)	Total Issues	Risks	Severity
Publicly exposed VM instance with effective global admin permissions	18 issues		
Critical/High network vulnerability with a known exploit found on a publicly exposed VM instance with high permissions	18 issues		
Publicly facing VM instance with data access to sensitive data and high/critical severity network vulnerability with a known exploit	8 issues		
Publicly exposed service account that can be assumed by all users allows lateral movement to admin privileges	7 issues		
Publicly exposed VM instance/serverless assigned a permission combination that could lead to privilege escalation	6 issues		
SSH Bruteforce on Public Machine with Critical Vulnerabilities	5 issues		
Resource infected with critical/high severity malware	5 issues		



Test-Encrypted
EC2 Instance

SB
EC2 Instance

EC2ContainerService-ecs-on-ec2-clu
Auto Scaling Group

Ephemeral_spotinst:aws:ec2:group:id
Spotinst Synthetic Instance Group

arn:aws:ec2:eu-west-2:984186218765
EC2 Instance

Publicly exposed VM instance with effective global admin permissions

Description

Description

This VM is exposed to the public internet and has admin permissions in the environment. Admin permissions allow persistence in the environment, or have wild card permissions on the subscription, account, or project level.

Potential Impact

Publicly exposed resources are more easily accessible for an attacker than internal ones. Moreover, highly privileged resources are a greater risk to the environment when compromised. Therefore, an attacker that manages to execute code on the publicly exposed VM can abuse the privileges, potentially performing administrative operations.

Severity
Critical

Compliance Frameworks
& WIZ ISO NIST +2

Risks

Subscription
aws AWS Demo Scenarios
us-east-2

Projects
3 Projects

Related Tickets
1 Ticket

Status
● Open

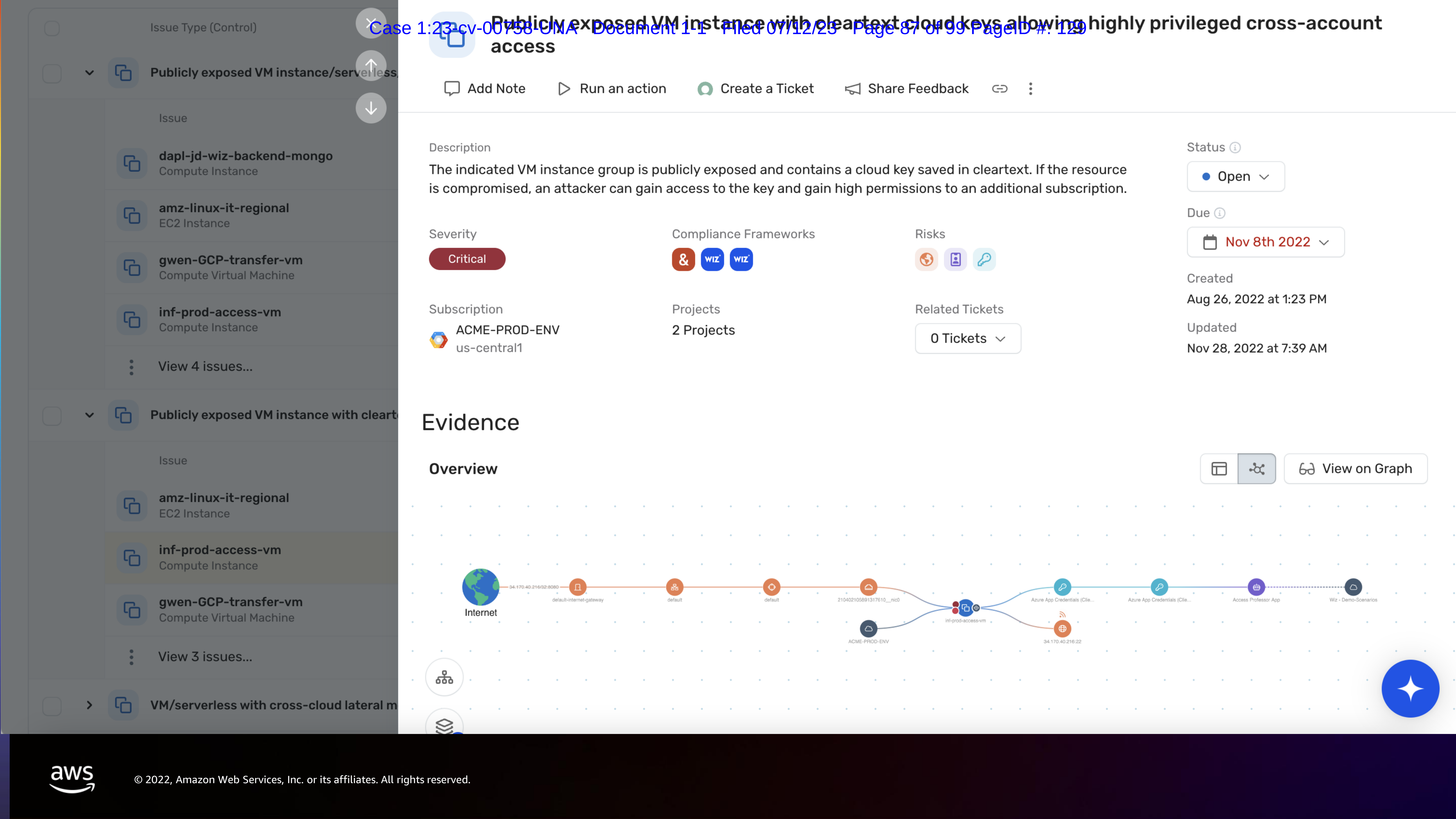
Due
Oct 23rd 2022

Created
Jul 18, 2022 at 3:46 PM

Updated
Nov 28, 2022 at 5:29 AM

Evidence

Overview



Issue Type (Control)

Publicly exposed VM instance/serverless

Issue

dapl-jd-wiz-backend-mongo
Compute Instance

amz-linux-it-regional
EC2 Instance

gwen-GCP-transfer-vm
Compute Virtual Machine

inf-prod-access-vm
Compute Instance

View 4 issues...

Publicly exposed VM instance with cleart

Issue

amz-linux-it-regional
EC2 Instance

inf-prod-access-vm
Compute Instance

gwen-GCP-transfer-vm
Compute Virtual Machine

View 3 issues...

VM/serverless with cross-cloud lateral m

Publicly exposed VM instance with cleartext cloud keys allowing highly privileged cross-account access

Description

The indicated VM instance group is publicly exposed and contains a cloud key saved in cleartext. If the resource is compromised, an attacker can gain access to the key and gain high permissions to an additional subscription.

Severity: Critical

Compliance Frameworks: & WIZ WIZ

Risks

Subscription: ACME-PROD-ENV
us-central1

Projects: 2 Projects

Related Tickets: 0 Tickets

Status: Open

Due: Nov 8th 2022

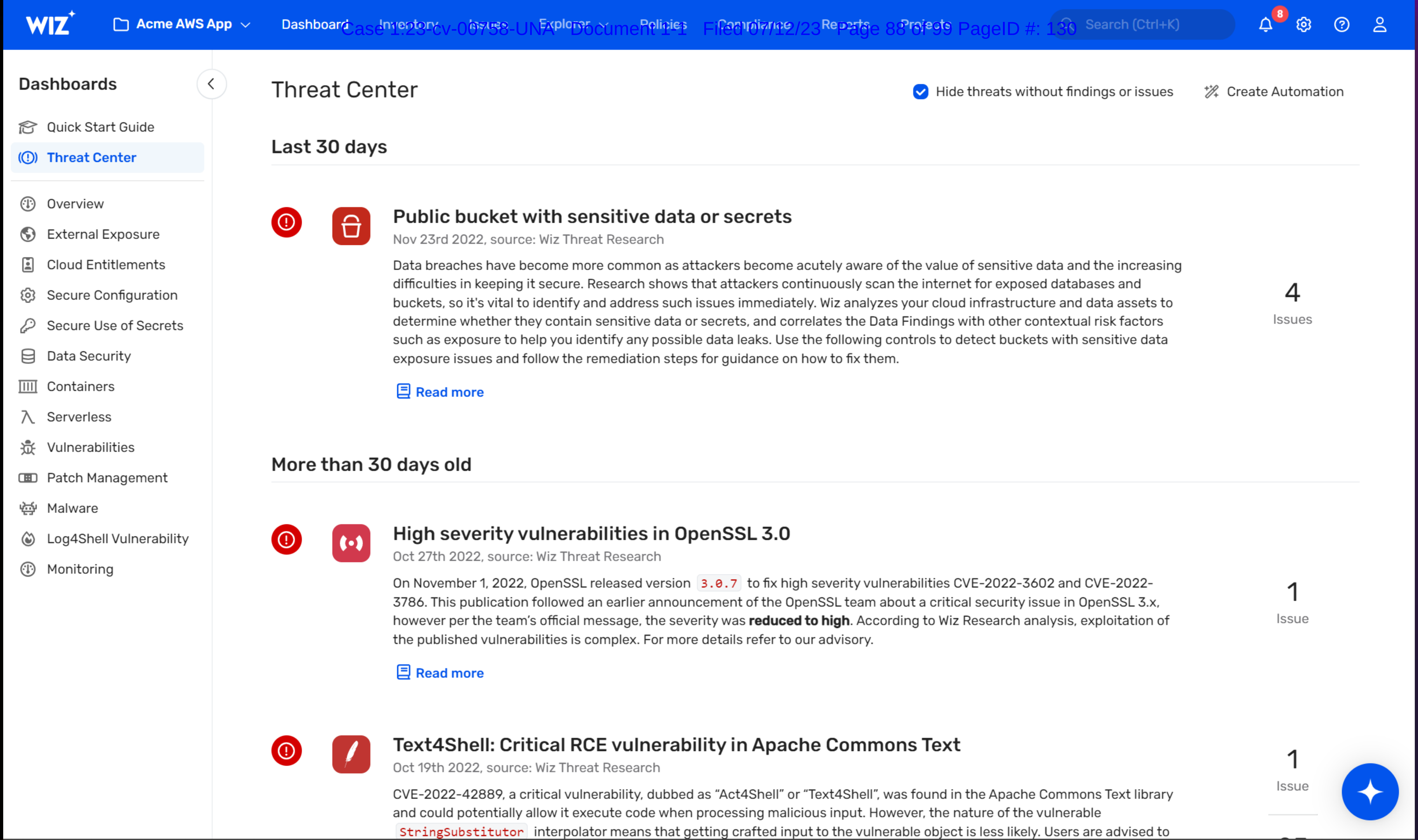
Created
Aug 26, 2022 at 1:23 PM

Updated
Nov 28, 2022 at 7:39 AM

Evidence

Overview

aws



Threat Center

☑ Hide threats without findings or issues

Create Automation

Last 30 days

Public bucket with sensitive data or secrets

Nov 23rd 2022, source: Wiz Threat Research

Data breaches have become more common as attackers become acutely aware of the value of sensitive data and the increasing difficulties in keeping it secure. Research shows that attackers continuously scan the internet for exposed databases and buckets, so it's vital to identify and address such issues immediately. Wiz analyzes your cloud infrastructure and data assets to determine whether they contain sensitive data or secrets, and correlates the Data Findings with other contextual risk factors such as exposure to help you identify any possible data leaks. Use the following controls to detect buckets with sensitive data exposure issues and follow the remediation steps for guidance on how to fix them.

Read more

4 Issues

More than 30 days old

High severity vulnerabilities in OpenSSL 3.0

Oct 27th 2022, source: Wiz Threat Research

On November 1, 2022, OpenSSL released version `3.0.7` to fix high severity vulnerabilities CVE-2022-3602 and CVE-2022-3786. This publication followed an earlier announcement of the OpenSSL team about a critical security issue in OpenSSL 3.x, however per the team's official message, the severity was **reduced to high**. According to Wiz Research analysis, exploitation of the published vulnerabilities is complex. For more details refer to our advisory.

Read more

1 Issue

Text4Shell: Critical RCE vulnerability in Apache Commons Text

Oct 19th 2022, source: Wiz Threat Research

CVE-2022-42889, a critical vulnerability, dubbed as "Act4Shell" or "Text4Shell", was found in the Apache Commons Text library and could potentially allow it execute code when processing malicious input. However, the nature of the vulnerable `StringSubstitutor` interpolator means that getting crafted input to the vulnerable object is less likely. Users are advised to

1 Issue

Where Wiz fits in the security stack today

Wiz CNAPP

CSPM

- Secure cloud and Kubernetes configuration
- Cloud external attack surface management
- Cloud Infrastructure Entitlement Management (CIEM)

CWPP

- Vulnerability and secure configuration assessment
- Container and serverless security
- Secrets scanning
- Malware scanning

CNAPP

- Asset Inventory and Security Graph
- Attack Path Analysis (APA)
- Toxic Combination and risk prioritization
- Cloud Detection and Response (CDR)
- Data Security Posture Management (DSPM)

Wiz Guardrails (CI/CD integration)

DevSecOps

- Container image scanning
- VM Image Gallery scans

- Registry scanning
- IaC Scanning (TF, CF, ARM, YAML, …)
- Kubernetes Admission Controllers

Wiz complements

- SaaS security (CASB, SASE, SSPM)
- Identity Management, Access and Zero trust (IAM, ZTNA)
- Application Security (SAST, DAST, WAF)
- Runtime Protection and behavior analysis (EDR, UEBA)
- Orchestration and event management (SIEM, SOAR)

Wiz provides immediate business value for AWS customers

Risk mitigation

- ✓ Eliminate blind spots
- ✓ Prevent issues in production
- ✓ Ensure readiness
- ✓ Scale in the cloud seamlessly

Operational efficiency

- ✓ Eliminate overhead of agents
- ✓ Reduce noise and manual effort
- ✓ Automate governance
- ✓ Speed remediation

Cost reduction

- ✓ Reduce license, deployment, integration, and support costs of point security products
- ✓ Identify unnecessary cloud usage

Accelerate the business

- ✓ Keep your devs focused on building
- ✓ Prevent issues in production
- ✓ Ensure readiness
- ✓ Scale in the cloud seamlessly

aws

Fireside chat

John Visneski
CISO – MGM Studios

Q&A

Visit us at booth #419

Learn more at www.wiz.io | Find Wiz in the AWS Marketplace

Thank you!

Please complete the session survey in the **mobile app**

EXHIBIT 6

Datasheet

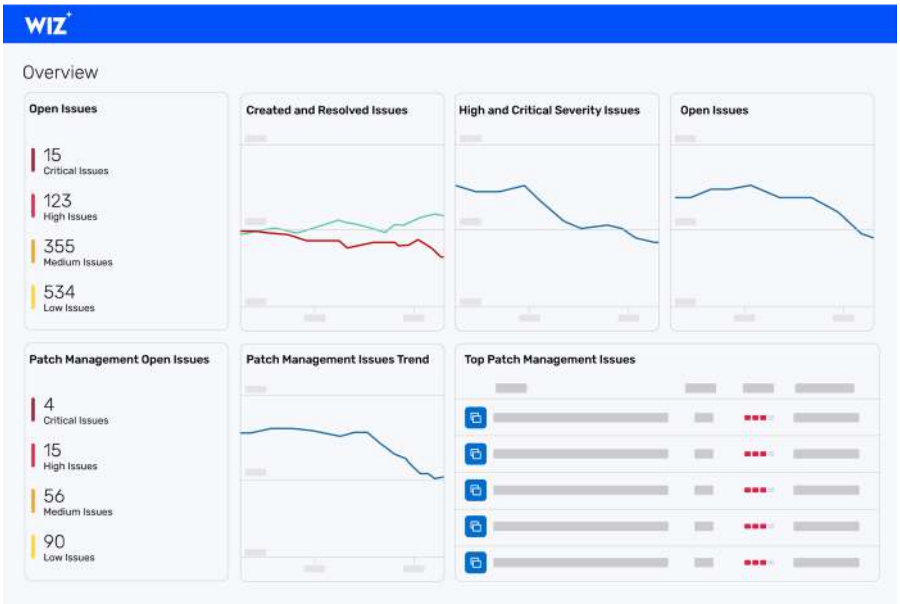
Wiz Cloud Security Platform

Take control of your cloud infrastructure security

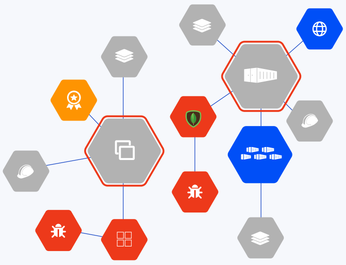
Wiz analyzes all layers of the cloud stack to reveal actionable insights about high-risk attack vectors in your cloud so you can prioritize and fix them.

Key use cases

- Get a complete and up-to-date inventory of all cloud resources: PaaS, VMs, containers, etc.
- Correlate issues across the cloud stack that together create high-risk infiltration vectors
- Route high-risk issues to the right teams to fix them and track resolution
- Create a cloud governance practice that manages risk within a defined risk budget



What makes Wiz different



You don't deploy Wiz, you connect it

With no agents or sidecars to deploy, Wiz begins delivering security value in minutes after you connect Wiz to your cloud environment API.

Actionable insights without the noise

Wiz combines the functionality of a CSPM, vulnerability scanner, container security, and CIEM into a single graph to correlate risks without the noise.

Total coverage ofyour environment

Wiz analyzes the full cloud stack without the limits of agents—every VM, every container, and every cloud service across AWS, Azure, GCP, Kubernetes and OpenShift.

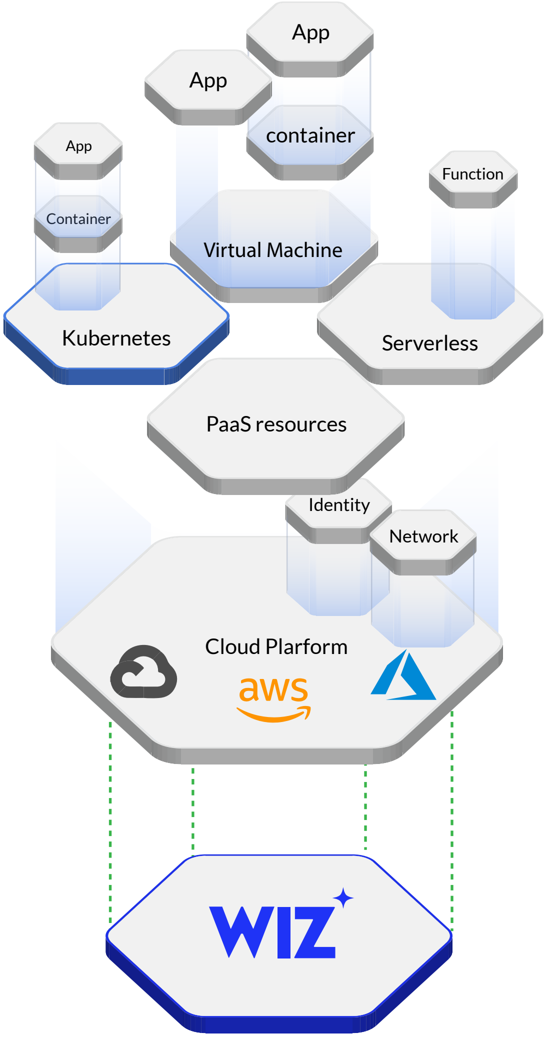
Datasheet

The first full-stack multi-cloud security platform

Depth into the full cloud stack

As soon as you connect Wiz to your cloud environment API, Wiz scans your entire cloud stack, not just the infrastructure layer. Inside workloads, Wiz analyzes the operating system, applications, code libraries, and secrets. Wiz also scans your cloud configuration and metadata.

- **Virtual machines**
- **Containers**
- **Serverless**
- **PaaS services**



Breadth across multiple clouds

Wiz gives you a unified view and ability to perform security with a common tool set across all your cloud environments.

- **Public cloud** – Amazon Web Services, Microsoft Azure, and Google Cloud Platform.
- **On premises** – Container environments deployed with OpenShift.
- **Every flavor of Kubernetes** – Self-managed Kubernetes clusters and managed container services from cloud

Agentless coverage of everything

Wiz scans all the resources and workloads in your cloud environment using a unique snapshot technology that covers more than an agent can.

- **Complete coverage** – Complete coverage of all VMs and containers, not just the ones with the agent or sidecar installed.
- **Short-lived resources** – Analyze short-lived resources created on the fly for autoscaling, which agents can't scan.
- **Managed instances** – Preconfigured virtual machine templates from third parties and marketplaces you can't install agents on.

Datasheet

Identify high-risk attack vectors

Until now, cloud security tools have created thousands of low-priority alerts because they look at vulnerabilities or misconfigurations in isolation. Wiz uses the full context of your cloud and combines this information in a single graph in order to correlate related issues that together create an infiltration vector, giving you actionable information about the highest risks so you can fix what matters most.

Secure use of secrets

Identify all keys located on your workloads cross referenced with the privileges they have in your cloud environment.

Identity and access

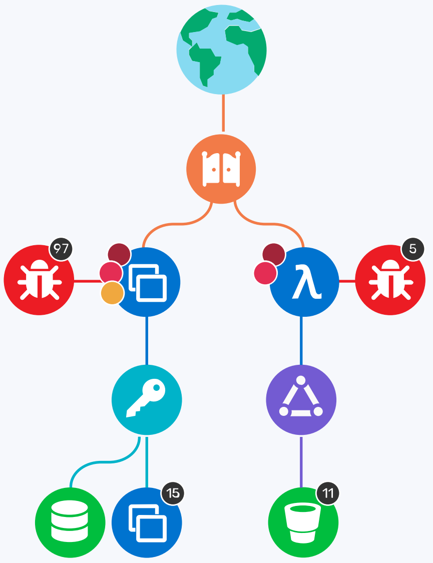
Map the identity structure of every resource to the role it can assume, taking into account mitigating controls such as service control policies (SCP) and permissions boundaries.

External exposure

See which resources are publicly exposed to the internet based on a full analysis of your cloud network, even those behind multiple hops.

Cloud Security Graph

Wiz combines all of the data about your cloud and workloads into a single graph, making it possible to correlate related issues that create attack vectors.



Lateral movement

Remove lateral movement risks such as private keys used to access both development and production environments.

Vulnerability and patch management

Scan for vulnerable and unpatched operating systems, installed software, and code libraries in your workloads prioritized by risk.

Secure configuration

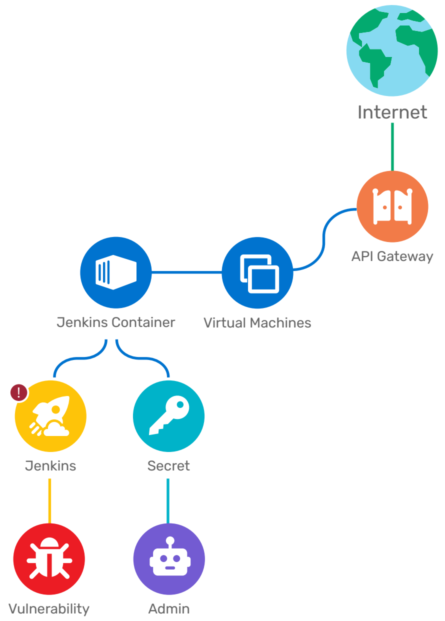
Assess the configuration of cloud infrastructure, Kubernetes, and VM operating systems against your baselines and industry best practices.

Real-world example

Unpatched Jenkins container running on a VM exposed to the internet with exploitable vulnerabilities and high-privilege secrets that give access to the production environment.

Only Wiz is able to pinpoint this kind of high-risk situation because it understands the full cloud context:

- Scans the workloads inside the container to determine the version of Jenkins and its vulnerabilities.
- Analyzes networking in your cloud to identify internet exposure on a machine with no public IP.
- Finds private keys (secrets) on the container and analyzes the permissions they have in your environment.



For more information visit www.wiz.io

Datasheet

Where Wiz fits in the security stack

Wiz includes many cloud security capabilities typically found in standalone products in one platform:

- ✓ Patch and vulnerability assessment
- ✓ Cloud security posture management (CSPM)
- ✓ Cloud inventory and asset management
- ✓ Container and serverless security
- ✓ Cloud network visibility – configuration analysis
- ✓ Cloud identity and entitlement management (CIEM)
- ✓ Secrets scanning and analysis in cloud workloads

There are also some cloud security features adjacent to Wiz that we don't cover:

- ○ Cloud access security broker (CASB)
- ○ Secure access service edge (SASE)
- ○ Zero trust network access (ZTNA)
- ○ Runtime protection
- ○ Netflow analysis
- ○ Secrets mangement

Key features

Snapshot scanning

Takes a snapshot of each VM system volume and analyzes its operating system, application layer, and data layer statically with no performance impact.

Secrets scanning and analysis

Finds cleartext keys stored on VMs and containers, parses the key to understand it, and maps the permissions it has within your environment.

Remediation workflow

Creates tickets directly in service tracking products like Jira and ServiceNow and sends alerts via email or messaging applications like Slack.

Inventory and asset management

Creates a complete and up-to-date inventory of all services and software in your cloud environment including the application version and package.

Noise-cancelling alerts

Collapses alerts for related resources into one alert (e.g. multiple VMs part of an instance group or containers from the same image).

Project-based cloud governance

Gives role-based access to Wiz's security capabilities, so developers and other teams can track risk in their projects and stay under a defined risk budget.

Supported Platforms

Cloud platforms

Get deep visibility into your cloud environment.

- ○ Amazon Web Services
- ○ Microsoft Azure
- ○ Google Cloud Platform (GCP)

Containers

Get deep visibility into your Kubernetes clusters.

- ○ OpenShift
- ○ Kubernetes
- ○ Google Kubernetes Engine (GKE)
- ○ Amazon Elastic Kubernetes Service (EKS)
- ○ Azure Kubernetes Service (AKS)
- ○ Standalone Containers

Integrations

CI/CD Pipelines

Shift-left your security scans with Wiz CLI that integrates seamlessly to the leading CI/CD pipelines.

Remediation workflow

Send risks to the right people to fix using built-in integrations to Slack, ServiceNow, Jira, and more...

Extensibility

Build upon Wiz's robust and fully documented webhooks and APIs that enable easy integration to any data platform you need.

JS 44 (Rev. 06/17)

CIVIL COVER SHEET

The JS 44 civil cover sheet and the information contained herein neither replace nor supplement the filing and service of pleadings or other papers as required by law, except as provided by local rules of court. This form, approved by the Judicial Conference of the United States in September 1974, is required for the use of the Clerk of Court for the purpose of initiating the civil docket sheet. *(SEE INSTRUCTIONS ON NEXT PAGE OF THIS FORM.)*

I. (a) PLAINTIFFS

Orca Security Ltd.

(b) County of Residence of First Listed Plaintiff __________
(EXCEPT IN U.S. PLAINTIFF CASES)

(c) Attorneys *(Firm Name, Address, and Telephone Number)*

Jack B. Blumenfeld; Morris, Nichols, Arsht & Tunnell LLP
1201 North Market Street; P.O. Box 1347; Wilmington, DE 19899
(302) 658-9200

DEFENDANTS

Wiz, Inc.

County of Residence of First Listed Defendant __________
(IN U.S. PLAINTIFF CASES ONLY)

NOTE: IN LAND CONDEMNATION CASES, USE THE LOCATION OF THE TRACT OF LAND INVOLVED.

Attorneys *(If Known)*

II. BASIS OF JURISDICTION *(Place an "X" in One Box Only)*

□ 1 U.S. Government Plaintiff
☒ 3 Federal Question *(U.S. Government Not a Party)*
□ 2 U.S. Government Defendant
□ 4 Diversity *(Indicate Citizenship of Parties in Item III)*

III. CITIZENSHIP OF PRINCIPAL PARTIES *(Place an "X" in One Box for Plaintiff and One Box for Defendant)*

(For Diversity Cases Only)

	PTF	DEF		PTF	DEF
Citizen of This State	□ 1	□ 1	Incorporated *or* Principal Place of Business In This State	□ 4	□ 4
Citizen of Another State	□ 2	□ 2	Incorporated *and* Principal Place of Business In Another State	□ 5	□ 5
Citizen or Subject of a Foreign Country	□ 3	□ 3	Foreign Nation	□ 6	□ 6

IV. NATURE OF SUIT *(Place an "X" in One Box Only)*

Click here for: Nature of Suit Code Descriptions.

CONTRACT

□ 110 Insurance
□ 120 Marine
□ 130 Miller Act
□ 140 Negotiable Instrument
□ 150 Recovery of Overpayment & Enforcement of Judgment
□ 151 Medicare Act
□ 152 Recovery of Defaulted Student Loans (Excludes Veterans)
□ 153 Recovery of Overpayment of Veteran's Benefits
□ 160 Stockholders' Suits
□ 190 Other Contract
□ 195 Contract Product Liability
□ 196 Franchise

REAL PROPERTY

□ 210 Land Condemnation
□ 220 Foreclosure
□ 230 Rent Lease & Ejectment
□ 240 Torts to Land
□ 245 Tort Product Liability
□ 290 All Other Real Property

TORTS

PERSONAL INJURY

□ 310 Airplane
□ 315 Airplane Product Liability
□ 320 Assault, Libel & Slander
□ 330 Federal Employers' Liability
□ 340 Marine
□ 345 Marine Product Liability
□ 350 Motor Vehicle
□ 355 Motor Vehicle Product Liability
□ 360 Other Personal Injury
□ 362 Personal Injury - Medical Malpractice

PERSONAL INJURY

□ 365 Personal Injury - Product Liability
□ 367 Health Care/ Pharmaceutical Personal Injury Product Liability
□ 368 Asbestos Personal Injury Product Liability

PERSONAL PROPERTY

□ 370 Other Fraud
□ 371 Truth in Lending
□ 380 Other Personal Property Damage
□ 385 Property Damage Product Liability

CIVIL RIGHTS

□ 440 Other Civil Rights
□ 441 Voting
□ 442 Employment
□ 443 Housing/ Accommodations
□ 445 Amer. w/Disabilities - Employment
□ 446 Amer. w/Disabilities - Other
□ 448 Education

PRISONER PETITIONS

Habeas Corpus:
□ 463 Alien Detainee
□ 510 Motions to Vacate Sentence
□ 530 General
□ 535 Death Penalty
Other:
□ 540 Mandamus & Other
□ 550 Civil Rights
□ 555 Prison Condition
□ 560 Civil Detainee - Conditions of Confinement

FORFEITURE/PENALTY

□ 625 Drug Related Seizure of Property 21 USC 881
□ 690 Other

LABOR

□ 710 Fair Labor Standards Act
□ 720 Labor/Management Relations
□ 740 Railway Labor Act
□ 751 Family and Medical Leave Act
□ 790 Other Labor Litigation
□ 791 Employee Retirement Income Security Act

IMMIGRATION

□ 462 Naturalization Application
□ 465 Other Immigration Actions

BANKRUPTCY

□ 422 Appeal 28 USC 158
□ 423 Withdrawal 28 USC 157

PROPERTY RIGHTS

□ 820 Copyrights
☒ 830 Patent
□ 835 Patent - Abbreviated New Drug Application
□ 840 Trademark

SOCIAL SECURITY

□ 861 HIA (1395ff)
□ 862 Black Lung (923)
□ 863 DIWC/DIWW (405(g))
□ 864 SSID Title XVI
□ 865 RSI (405(g))

FEDERAL TAX SUITS

□ 870 Taxes (U.S. Plaintiff or Defendant)
□ 871 IRS—Third Party 26 USC 7609

OTHER STATUTES

□ 375 False Claims Act
□ 376 Qui Tam (31 USC 3729(a))
□ 400 State Reapportionment
□ 410 Antitrust
□ 430 Banks and Banking
□ 450 Commerce
□ 460 Deportation
□ 470 Racketeer Influenced and Corrupt Organizations
□ 480 Consumer Credit
□ 490 Cable/Sat TV
□ 850 Securities/Commodities/ Exchange
□ 890 Other Statutory Actions
□ 891 Agricultural Acts
□ 893 Environmental Matters
□ 895 Freedom of Information Act
□ 896 Arbitration
□ 899 Administrative Procedure Act/Review or Appeal of Agency Decision
□ 950 Constitutionality of State Statutes

V. ORIGIN *(Place an "X" in One Box Only)*

☒ 1 Original Proceeding
□ 2 Removed from State Court
□ 3 Remanded from Appellate Court
□ 4 Reinstated or Reopened
□ 5 Transferred from Another District *(specify)*
□ 6 Multidistrict Litigation - Transfer
□ 8 Multidistrict Litigation - Direct File

VI. CAUSE OF ACTION

Cite the U.S. Civil Statute under which you are filing ***(Do not cite jurisdictional statutes unless diversity)***:
35 U.S.C. § 271

Brief description of cause:
Patent Infringement

VII. REQUESTED IN COMPLAINT:

□ CHECK IF THIS IS A **CLASS ACTION** UNDER RULE 23, F.R.Cv.P.

DEMAND $

CHECK YES only if demanded in complaint:
JURY DEMAND: ☒ Yes □ No

VIII. RELATED CASE(S) IF ANY

(See instructions): JUDGE __________ DOCKET NUMBER __________

DATE
07/12/2023

SIGNATURE OF ATTORNEY OF RECORD
/s/ Jack B. Blumenfeld

FOR OFFICE USE ONLY

RECEIPT # _____ AMOUNT _____ APPLYING IFP _____ JUDGE _____ MAG. JUDGE _____