Vulnerability Management Tutorials

Vulnerabilities are weaknesses in computer systems that can be exploited by malicious attackers. Whether they are caused by bugs or design flaws, vulnerabilities can allow attackers to execute code in your environment or elevate their privileges. Because vulnerabilities are such a common attack surface, understanding how each compute instance in your environment is vulnerable is crucial to assessing potential risks.

The following scenarios provide step-by-step instructions to help you understand a few key ways we want you to use Wiz to address your vulnerability risks:

- Identify common critical exploitable vulnerabilities
- View how many CVEs are associated with a VM
- Associate vulnerabilities with a container execution context
- Identify vulnerabilities validated in runtime
  - Identify containers running vulnerable packages validated in runtime
  - Identify containers running vulnerable libraries validated in runtime
  - Identify critical attack paths involving vulnerabilities validated in runtime
- Prioritize remediation by the number of affected resources per vulnerability or the number of vulnerabilities per resource
- View critical exploitable network vulnerabilities
- Detect vulnerabilities in VCS repositories
- View CVEs with an exploit in the last 60 days
- Exposed compute resources with the latest exploitable vulnerabilities
- View all unresolved vulnerability Issues

For a more abstract discussion of vulnerabilities, see the How Wiz Works page Vulnerability Management.

> You can watch our webinar on vulnerabilities.

Identify common critical exploitable vulnerabilities

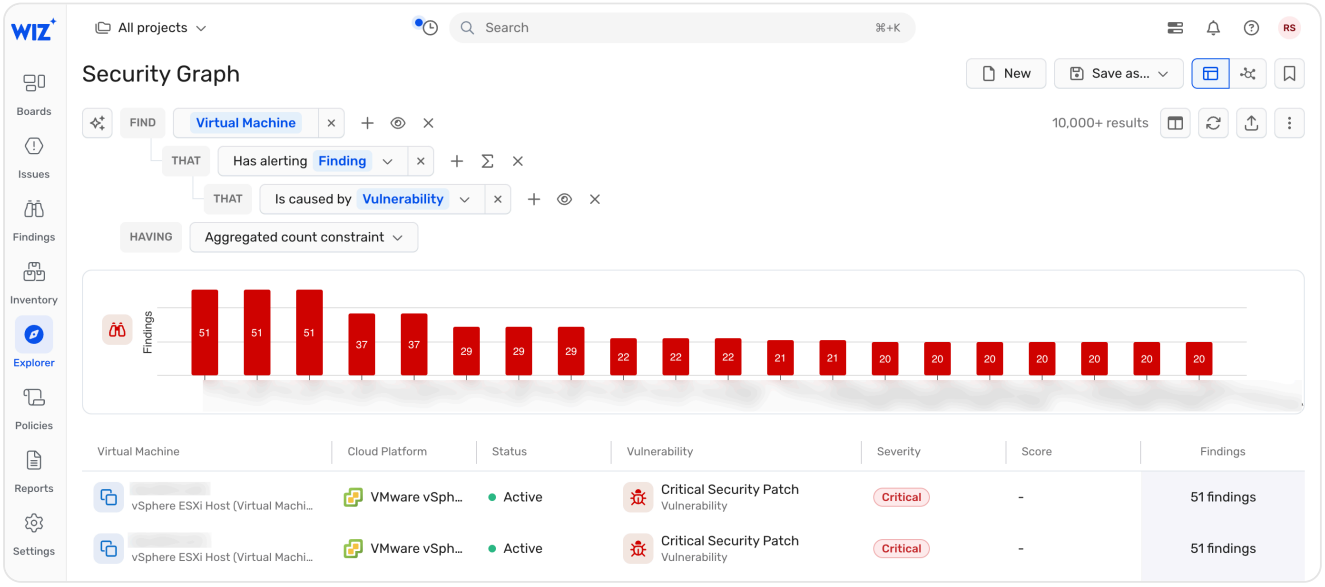
1. Go to the Vulnerabilities board.

2. Click the Most Common Critical Exploitable Vuln widget (direct link).
3. Investigate the vulnerabilities and/or the resources on which they were found to prioritize remediation efforts:
    i. Click a vulnerability (left) to read a brief description or follow the link to the official definition.
    ii. Click "X resources" (right) to query the Security Graph for affected resources. More sensitive resources should be addressed first.
4. Look for the `Fixed Version` property of a specific vulnerability:
    i. On the Finding query criterion, click 👁 > Show these entities (they were hidden to simplify the initial results table; direct link).
    ii. Click a Finding. The Finding's details drawer opens on the right.
    iii. In the Finding's details drawer, look for the `Fixed Version` property for the associated vulnerability.

View how many CVEs are associated with a VM

> The sheer number of vulnerabilities associated with a VM is rarely the best metric for risk. This is an example of the quantitative fallacy, i.e. focusing on what can be measured easily. Instead, we recommend focusing on remediating Issues generated by vulnerability assessment Controls, which correlate vulnerabilities with other risk factors, like external exposure and excessive permissions.

1. Go to the Security Graph.
2. Search for all VMs with Vulnerability Findings (direct link).
3. On the Finding query criterion, click 👁 > Aggregate these entities (direct link).
4. In large environments, this query may time out, in which case you could try optimizing it.



> The Security Graph contains only vulnerabilities that are either critical or high severity or CISA KEV exploitable. If you need to also count low and medium vendor severity and/or non-CISA KEV exploitable vulnerabilities, you have to export all vulnerabilities as a report or using the Wiz API, and then work with the downloaded CSV.

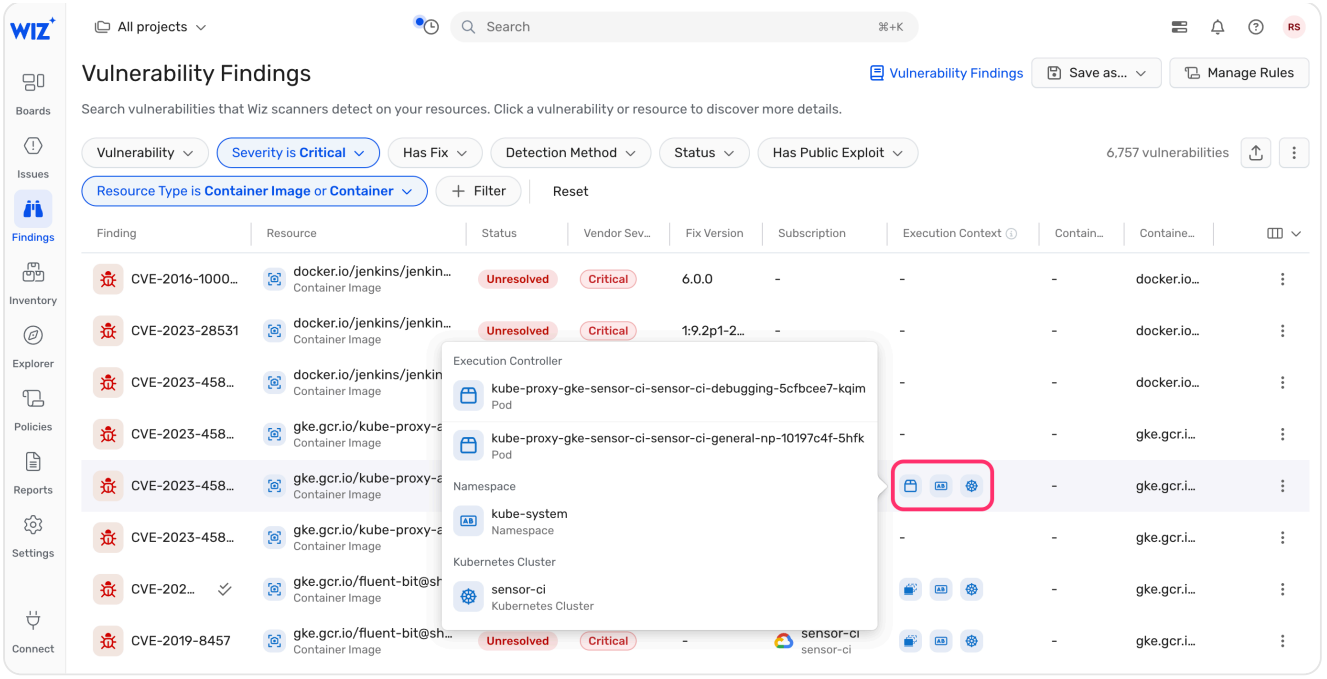
Associate vulnerabilities with a container execution context

In containerized environments, it is insufficient to solely know which container image is affected by a vulnerability. Instead, contextual information is required to determine where this container image is deployed and operational. This context encompasses Kubernetes controllers (such as deployments and daemon sets), namespaces, clusters, and other container services (e.g., ECS Service) that are utilizing the compromised container.

Learn how to use Wiz to associate Vulnerability Findings in container images and containers with their execution context to accurately identify the specific resources impacted by the vulnerability, and attribute it to the appropriate owner for remediation.

1. In Wiz, go to the Findings > Vulnerability Findings page.
2. Narrow down the scope to only containers or container images with vulnerabilities (direct link).
3. Find the Execution Context column. Hover over that column to get a list of controllers, namespaces, and clusters where the vulnerable container image or container is running.

   > If you don't see the Execution Context column, add it by clicking Show/Hide Table Columns on the right.



4. Click a Vulnerability Finding (that has icons in the Execution Context column) to open its details drawer, then locate the Execution Controllers running the Vulnerable Container Image section to view the full execution context.

Identify vulnerabilities validated in runtime

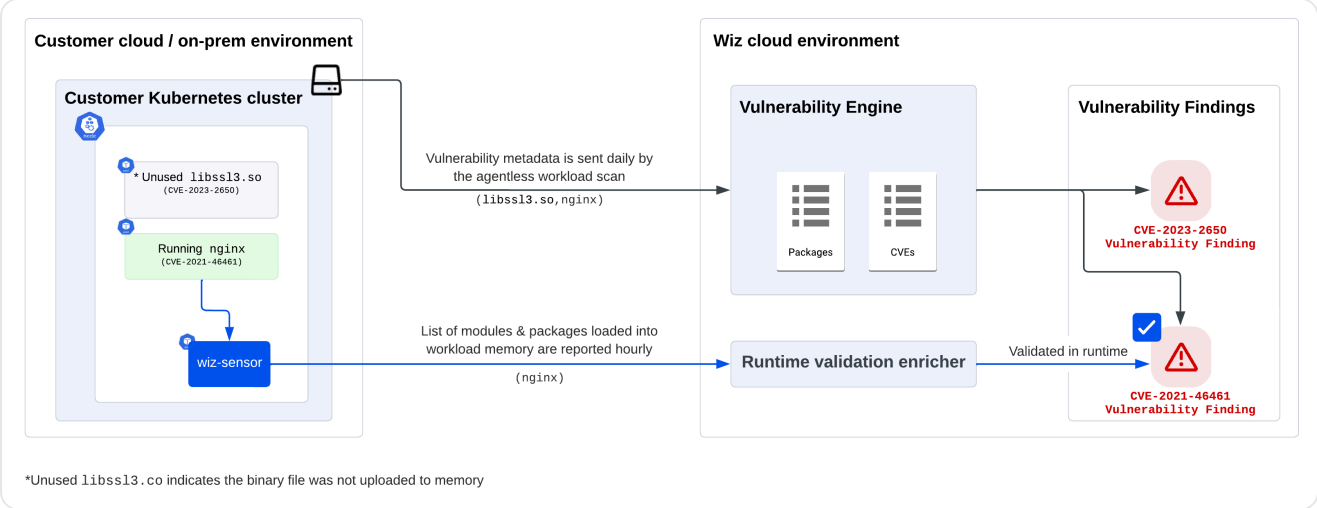
> This use case requires a Wiz Runtime Sensor.

The Wiz Runtime Sensor adds runtime signals on top of the agentless vulnerability assessment. This allows Wiz to identify vulnerabilities executed in runtime so you can focus your remediation efforts on active vulnerable packages and their associated resources.

For each container where the Sensor is deployed, it generates a comprehensive list of all the modules which are active and loaded into memory and sends the list to the Wiz backend. When Wiz's Workload Scanner identifies a vulnerable package, we look for it on the list of modules. If the vulnerable package exists on the list, then Wiz generates a Vulnerability Finding for the resource and the vulnerable package, flags the Finding with the `Validated in Runtime` property and marks it with the icon. Learn about Security Graph icons.

> The Sensor validates also ignored Vulnerability Findings. Learn about Ignore Rules.

The property is removed from the Vulnerability Finding if the Sensor does not detect the module loaded in memory within the last 48 hours.



For example, let's look at a container with 120 packages, out of which 100 have vulnerabilities and 35 of the 100 vulnerable packages are active and loaded into memory. In this case, Wiz generates 100 Vulnerability Findings for the container, while the 35 Vulnerability Findings associated with the active packages are also flagged with the `Validated in Runtime` property. Use this query to see only Vulnerability Findings that were validated in runtime.

Monitoring and reporting mechanism

Packages

When an executable module (main binary or shared object) belonging to a vulnerable package is loaded into the memory of a process, this package is considered validated in runtime.

To generate the list of all the modules which are active and loaded into memory, the Sensor:

- Enumerates all modules loaded on Sensor startup (using `/proc/<pid>/maps`) to build a coherent map of the system state.
- Monitors all process-start (`exec`) system calls.
- Watches all Linux module load events in real-time (using `mmap`). This guarantees that both the modules loaded statically and the modules loaded dynamically (using `dlopen`) are monitored.

The Sensor monitoring is continuous and not based on polling. If a process or module is not seen for more than 48 hours, then it is considered stale (i.e., not validated in runtime) and therefore excluded from the hourly report sent to Wiz.

Libraries–.NET and Go

Wiz's agentless scanning detects vulnerable libraries used inside the main executable of a .NET or Go program. The Sensor detects when such a program is

loaded into memory and marks the vulnerability as validated in runtime.

Libraries-Java

The Sensor monitors active `jar` and `war` files:

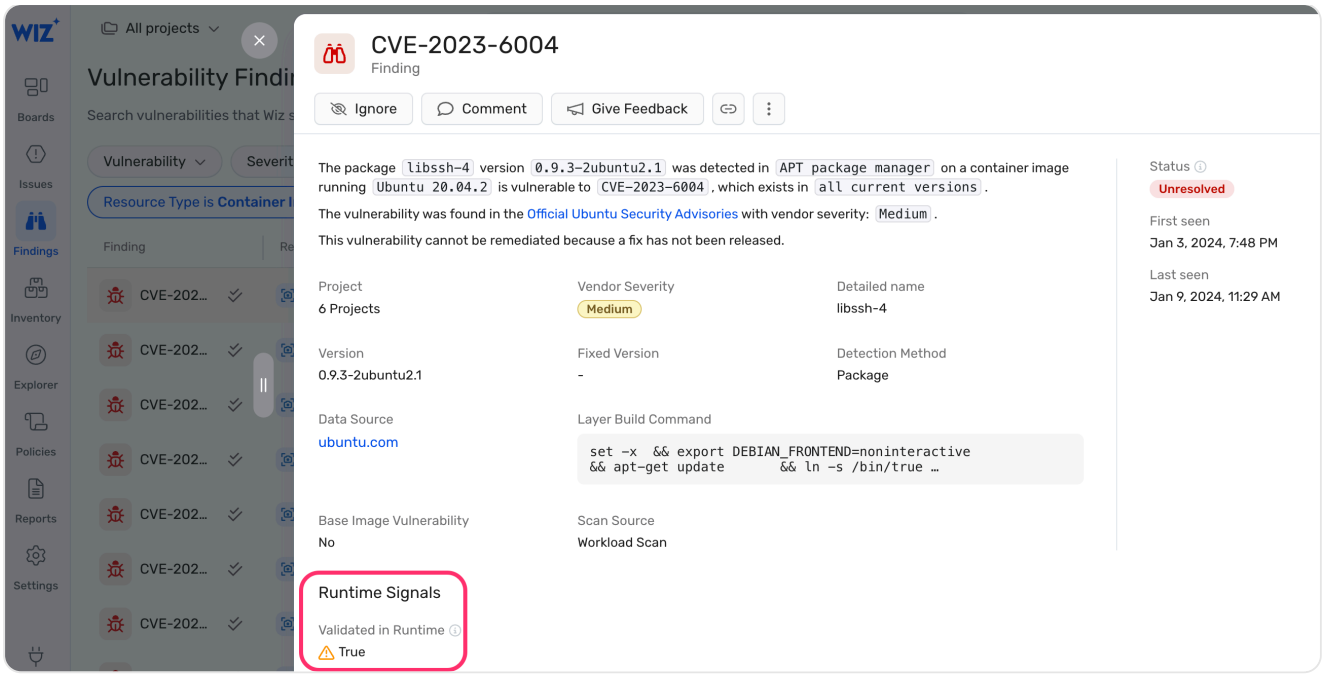
- Enumerates all such files loaded into a `java` process on Sensor startup (using the process `fd` map).
- Tracks additional files loaded into the `java` memory.

When a file path marked as active is determined to have a vulnerability by the Vulnerability Scanner, the vulnerability is marked as `Validated in runtime`.

Identify containers running vulnerable packages validated in runtime

Validating runtime risks helps you distinguish between active and dormant vulnerabilities so you can prioritize your mitigation efforts.

1. Go to the Findings > Vulnerability Findings page.
2. Narrow down the scope to only containers or container images with vulnerabilities (direct link).
3. Locate only the vulnerabilities detected in packages (direct link).
4. Click + Filter, add Validated in Runtime, and choose True (direct link).
5. The Vulnerability Finding object indicates that at least one Wiz Runtime Sensor detected the package running in at least one container running the image (this value is not set if no Runtime Sensors are deployed in any clusters running containers based on this image). Click on a Finding to view its details drawer.



Identify containers running vulnerable libraries validated in runtime

Validating runtime risks helps you distinguish between active and dormant vulnerabilities so you can prioritize your mitigation efforts.

1. Go to the Findings > Vulnerability Findings page.
2. Narrow down the scope to only containers or container images with vulnerabilities (direct link).
3. Locate only the vulnerabilities detected in libraries (direct link).
4. Click + Filter, add Validated in Runtime, and choose True (direct link).
5. The Vulnerability Finding object indicates that at least one Wiz Runtime Sensor detected the library running in at least one container running the image (this value is not set if no Runtime Sensors are deployed in any clusters running containers based on this image). Click on a Finding to view its details drawer.

Identify critical attack paths involving vulnerabilities validated in runtime

Wiz provides out-of-the-box Controls that detect toxic combinations involving a container and a validated-in-runtime vulnerability.

For example:

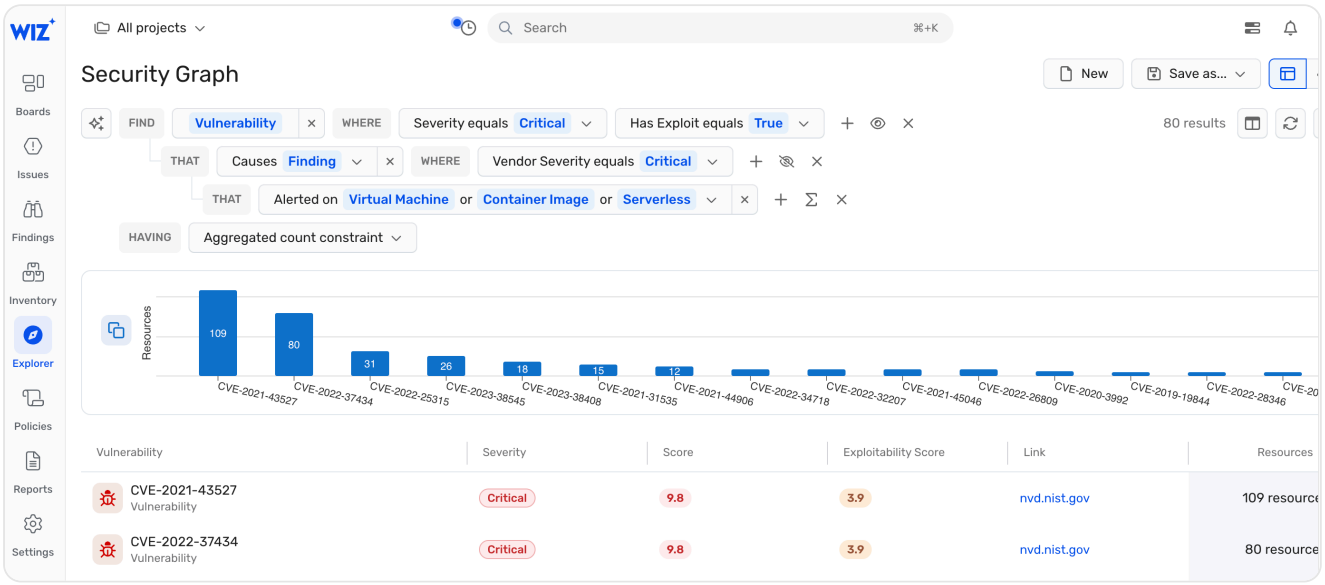
1. In Wiz, go to Policies > Controls. At the top, filter by Category > Wiz for Risk Assessment > Vulnerability Assessment (direct link).
2. Review the list of available Controls. Click a Control to see all the Issues that Wiz has detected for it. For example, Issues generated for Publicly exposed container with an image with high/critical severity vulnerabilities that were validated in runtime and a lateral movement finding to admin privileges. (direct link).

3. Open the Issue's details drawer to see its details, including remediation steps.

Prioritize remediation by the number of affected resources per vulnerability or the number of vulnerabilities per resource

1. Go to the Vulnerabilities board.
2. Scroll down and click the Most Common Critical Exploitable Vuln widget (direct link). Resources are aggregated by Vulnerability.

In this example, there are 109 resources affected by CVE-2021-43527. That can't be good.



3. Consider prioritizing remediation efforts according to the number and/or importance of the affected resources. If necessary, click X resources to query the Graph for all resources affected by the selected CVE.
4. Widen the query by clearing the severity filters from both the Vulnerability and Finding criteria (direct link).

> It is necessary to clear both, because a vulnerability and a Finding can refer to the same CVE but have different severities. See the guide on vulnerability management.

5. Re-aggregate by vulnerabilities instead of resources (direct link):
   i. On the Alerted on criterion, click ∑ Column aggregated > Show these entities.
   ii. On the Finding criterion, click ∑ Column visible > Hide these entities.
   iii. On the Vulnerability criterion, click 👁 Column visible > Aggregate these entities.
6. Consider prioritizing remediation efforts according to which resources are affected by the most Vulnerabilities.

View critical exploitable network vulnerabilities

Vulnerabilities that can be exploited remotely and are already being actively used by malicious actors in the wild pose a clear and present danger to your organization's data and cloud infrastructure.

To view critical exploitable network vulnerabilities:

1. Go to the Vulnerabilities board.
2. Scroll down and click the Critical Exploitable Network Vuln widget (direct link).
3. On the left, click a CVE to view its description and other key information.
4. On the right, click X resources to query the Security Graph for all resources affected by the selected CVE.

Detect vulnerabilities in VCS repositories

> To block pull requests that contain vulnerabilities, enable event-triggered scanning for the version control Connector; follow the instructions of the first solution here. You can also apply custom CI/CD policies.

1. Go to the Findings > Vulnerability Findings page.
2. At the top, click Resource Type > Repository branch (direct link).

View CVEs with an exploit in the last 60 days

1. Go to the Vulnerabilities board.
2. Scroll down and click the CVEs with an Exploit in the Last 60 Days widget (direct link).
3. On the left, click a CVE to view its description and other key information.
4. On the right, click X resources to query the Security Graph for all resources affected by the selected CVE.

Exposed compute resources with the latest exploitable vulnerabilities

1. Go to the Vulnerabilities board.
2. Click the Publicly Exposed Compute With Latest Exploitable Vulnerabilities widget (direct link).
3. In the query results, click a virtual machine. The virtual machine details drawer opens on the right.
4. Review the important properties of the VM to gauge the potential risk and attack vectors:
    - Network exposure (both internal and external)

- OS
- Privileges

5. Switch to the Vulnerability tab and scroll down to the All Vulnerabilities section.
6. On the right side, click View All to see all Findings related to the hosted technologies installed on this VM on the Vulnerability Findings page.
7. (Optional) If there is a large number of Findings, add filters such as critical/high severity.
8. Click a Finding. Its details drawer opens on the right.
9. In the details drawer, explore the Vulnerability Finding:
   - Review the description at the top
   - Follow the link to the official NIST reference

View all unresolved vulnerability Issues

Like practically everything in Wiz, vulnerabilities can be correlated with other risk factors to detect toxic combinations that also involve network exposure, identity & access, exposed secrets, and more. In order to detect the most toxic combinations that involve vulnerabilities, you should view the critical severity Issues that are generated by built-in Controls.

1. Go to the Vulnerabilities board.
2. Click the Top Vulnerability Issues widget (direct link).
3. Investigate, monitor, and follow up on Issues. See the complete guide on the Issues page.

Updated 8 days ago

← How Vulnerability Management Works

 Vulnerability Management FAQ →