

[← All articles](#)

Vulnerability Scanning: How it Works, Why it Matters, and What You'll Find



Wiz Experts Team
September 30, 2024

6 minute read

Free Vulnerability Assessment

Get a Demo

What is vulnerability scanning?

Vulnerability scanning is the process of detecting and evaluating security flaws in IT systems, networks, and software. **Vulnerability scanners** are automated tools

that continuously search systems for known security vulnerabilities, including missing security updates, misconfigurations, and exposed secrets.

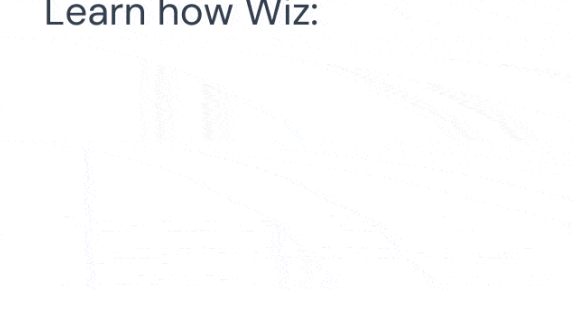
Vulnerability scanning cuts across all verticals of the organization's IT ecosystem:

- Networks
- Endpoints
- APIs
- Dependencies
- In-house and third-party apps

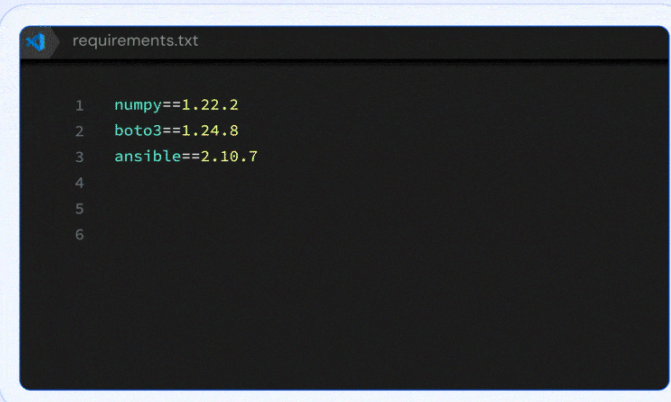
These scans are done to identify weaknesses across your systems and networks. Scanners are also usually purpose-based: they can have network-based, host-suited, or database-suited specializations. Depending on an organization's specific security needs, vulnerability scanning can be limited to individual systems or expanded to include entire network infrastructures.

See **Wiz Code** in Action

Learn how Wiz:



[Get a Demo](#)



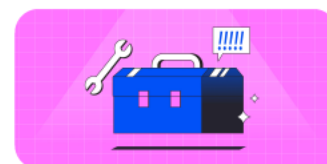
Why vulnerability scanning is important for security

Vulnerability scanning is essential for organizations as it proactively identifies and mitigates security weaknesses, preventing potential breaches and reducing associated costs. It also ensures compliance with **security frameworks** and provides valuable insights into the security posture of digital assets for more efficient resource allocation.

Benefit	Description
Proactive Security	Vulnerability scanning allows organizations to identify and address security weaknesses before attackers can exploit them. This proactive approach helps in preventing potential security breaches, reducing the risk of data loss, financial damage, and reputational harm.
Compliance and Regulatory Requirements	Vulnerability scanning helps organizations achieve data and software security, to better align with compliance frameworks such as SOC 2, ISO 27001, and NIST 800-53.
Cost savings	Identifying and remedying vulnerabilities early can significantly reduce the costs associated with a security breach. The financial implications of a breach often extend beyond immediate remediation efforts to include legal fees, fines, and lost business. Regular scanning helps organizations allocate resources more efficiently by prioritizing vulnerabilities based on their severity.
Asset Visibility and Management	Vulnerability scanning provides an inventory of all devices and software on a network, offering valuable insights into the security posture of an organization's digital assets. This visibility is crucial for effective asset management, ensuring that all parts of the IT infrastructure are up-to-date and secure.
Improved security posture	Regular scanning enables organizations to continuously assess and improve their security posture. By identifying and tracking vulnerabilities over time, organizations can measure the effectiveness of their security strategies and make informed decisions about where to invest in security improvements.

Vulnerability Management Buyer's Guide

Everything you need to know when assessing a vulnerability management solution: actionable reports, remediation steps, and RFP templates.



[Get the Guide >](#)

Types of vulnerability scanning use cases and common use cases

Techniques used for vulnerability scanning can be active or passive:

- **Active scanning**, also called non-credentialed scanning, involves sending simulated attacks, queries, or requests to the target to identify potential vulnerabilities, such as buffer overflows, unencrypted data, and broken authentication processes.
- **Passive scanning**, also called credentialed scanning, involves unobtrusively analyzing (without actively probing) network traffic to detect vulnerabilities that attackers can leverage to spread malware or steal/manipulate data.

Vulnerability scanning can also be categorized into different use cases, such as:

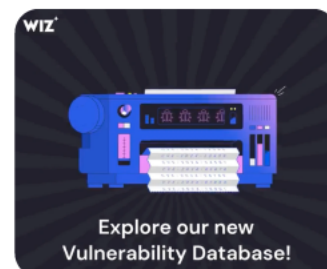
- **Network vulnerability scanning:** Scans the network for vulnerabilities, including open ports, unpatched software, and weak network protocols.
- **Web application vulnerability scanning:** Looks for security flaws such as SQL injection, **cross-site scripting (XSS)**, and other vulnerabilities that are unique to web applications.
- **Database vulnerability scanning:** Concentrates on identifying vulnerabilities within databases, such as misconfigurations, weak authentication, and permissions that are too permissive.

- **Host vulnerability scanning:** Performed on individual hosts (servers or workstations) to identify vulnerabilities at the operating system level or in installed software.
- **Container and Virtualized Environment Scanning:** Designed to identify vulnerabilities in **containerized applications** and virtual environments. This includes scanning for vulnerabilities in the container images and the management of containers and virtual machines.

The CVE Database: Curated Vulnerability Intelligence by Wiz

Wiz's CVE Database curates CVE data to create easy-to-navigate profiles that cover the entire vulnerability timeline, exploit scenarios, and mitigation steps.

[Explore database >](#)



How does vulnerability scanning work? 8 steps

The vulnerability scanning process involves eight steps:

1. **Scoping**
2. **Tool selection**
3. **Configuration**
4. **Scan initiation**
5. **Vulnerability detection**
6. **Vulnerability analysis**
7. **Remediation and rescanning**
8. **Continuous monitoring**

Here's a simple breakdown of each step:

Stage 1: Scoping

Before scanning, you must determine the target networks and applications, map out endpoints, and identify dependencies. Scoping also involves determining if internal devices, external-facing systems, or a combination of both are to be scanned.

Stage 2: Tool selection

You must choose a solution—from the pool of available commercial and **open-source tools**—that aligns with your organization's security requirements. The solution should also have a user-friendly console for easy vulnerability scanning and function optimally across distributed, hybrid networks to ease risk identification across all your environments.

Pro tip

Agentless scanning solutions typically have quicker setup and deployment and require less maintenance. They can scan all workloads using cloud native APIs and connects to customer environments with a single org-level connector. If the approach is agent-based, this type of deployment will require ongoing agent installation, update, and maintenance effort.

Stage 3: Configuration

The scanning tool should be configured to scan according to your desired parameters. Configuration details can include specifying target IP addresses or

domain names, setting scanning intensity or speed, and defining scanning techniques.

No organization should resort to using default policy configurations. This is because default policy configurations rarely address an organization's nuanced business-, region-, and industry-specific requirements.

Stage 4: Scan initiation

Initiate the process via commands or using the options provided by the tool of choice, such as a GUI. Some resources will allow you to schedule your scans, which makes this step automatic once you select your preferences.

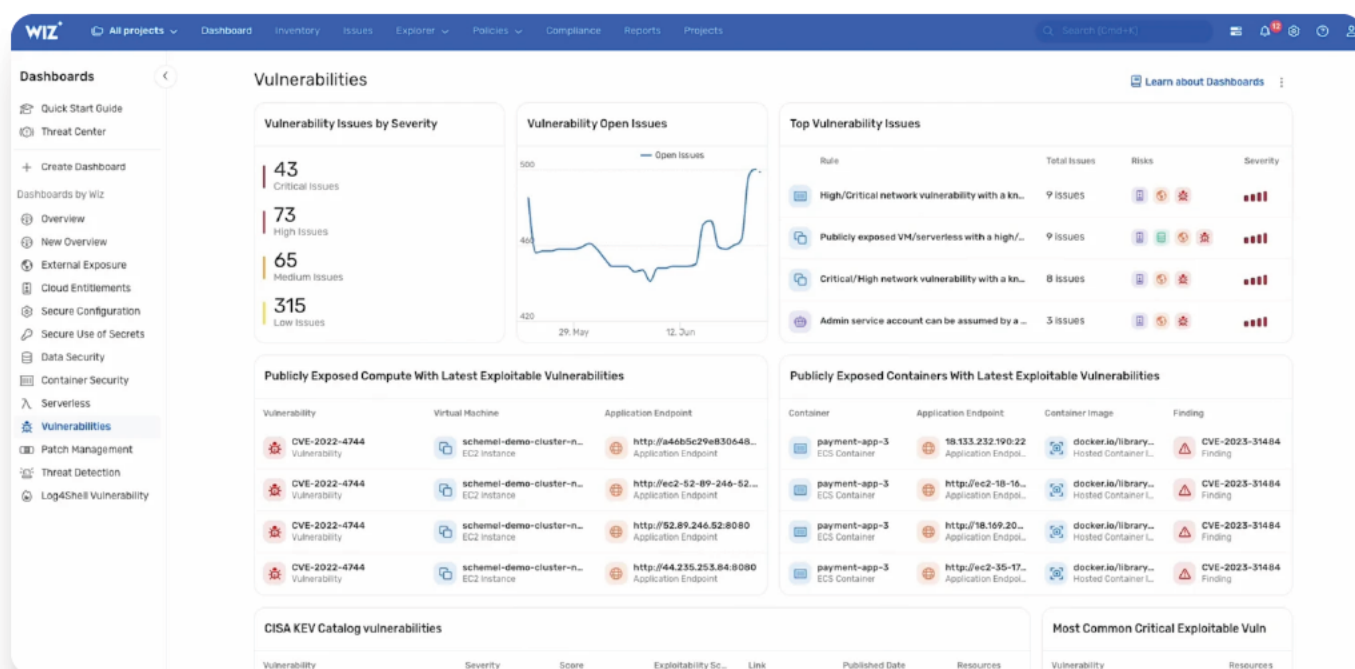
Stage 5: Vulnerability detection

CISA KEV Catalog vulnerabilities						
Vulnerability	Severity	Score	Exploitabi...	Link	Published...	Resources
 CVE-2021-30661 Vulnerability	 Hi...	8.7	2.8	nvd.nist...	Wed, Sep ...	75 resources
 CVE-2021-3156 Vulnerability	 Hi...	7.7	1.8	nvd.nist...	Tue, Jan ...	44 resources
 CVE-2020-15999 Vulnerability	 Me...	6.4	2.8	nvd.nist...	Tue, Nov ...	41 resources
 CVE-2021-4034 Vulnerability	 Hi...	7.7	1.8	cve.mitre...	Fri, Jan 2...	41 resources
 CVE-2021-22600 Vulnerability	 Hi...	7.7	1.8	nvd.nist...	Wed, Jan ...	36 resources

Example of vulnerability detections aligned with the CISA KEV catalog

Scanners probe for common vulnerability types or compare the system's attack surface with parameters saved in the **vulnerability database** in use. The vulnerabilities being scanned for will usually align with the scanner's speciality, whether that's databases, networks, etc.

Stage 6: Vulnerability analysis



Example vulnerability dashboard that prioritizes issues by contextual severity

After scanning, the tool will generate a comprehensive list of identified vulnerabilities, order them based on severity, filter away false positives, and provide options for remediation.

Stage 7: Remediation and rescanning

Vulnerability Findings

Search vulnerabilities that Wiz scanners detect on your resources. Click a vulnerability or resource to discover more details.

Vulnerability Severity Has Fix Detection Method Status Has Public Exploit equals

Findings

Finding	Resource	Details
CVE-2023-28841	test-fsck Virtual Machine	github.com
CVE-2023-28840	test-fsck Virtual Machine	github.com
CVE-2020-9283	test-fsck Virtual Machine	golang
CVE-2020-8554	test-fsck Virtual Machine	k8s.io
CVE-2020-8554	test-fsck Virtual Machine	k8s.io
CVE-2020-8912	test-fsck Virtual Machine	github.com
CVE-2020-8911	test-fsck Virtual Machine	github.com
CVE-2020-9283	yanivsv-temp-vm Virtual Machine	golang
CVE-2020-8912	yanivsv-temp-vm Virtual Machine	github.com
CVE-2020-8911	yanivsv-temp-vm Virtual Machine	github.com
CVE-2020-8554	yanivsv-temp-vm Virtual Machine	k8s.io
CVE-2020-8554	yanivsv-temp-vm Virtual Machine	k8s.io
CVE-2020-9283	yanivsv-temp-vm Virtual Machine	golang

CVE-2023-28841 Finding

Ignore Comment Share Feedback

Description

The library `github.com/docker/docker` version `20.10.8+incompatible` was detected in `Golang` binary located at `[/Partition611fda27]/home/kubecrates/bin/crictl` and is vulnerable to `CVE-2023-28841`, which exists in versions `>= 1.12.0, < 20.10.24`.

The vulnerability was found in the [GitHub Security Advisory](#) with vendor severity: `Medium` (NVD severity: `Medium`).

This vulnerability has a known exploit available. Source: [GitHub](#).

The vulnerability can be remediated by updating the library to any of the following versions: 23.0.3, 20.10.24 or higher: using `go get -u github.com/docker/docker`

Project `5 Projects` **Severity** `Medium` **Detailed name** `github.com/docker/docker`

Version `20.10.8+incompatible` **Fixed Version** `23.0.3, 20.10.24` **Detection Method** `Library`

Data Source `github.com`

Primary Resource [GCP](#) [JSON](#) [Add note](#) [View](#)

Virtual Machine **Cloud Platform** **Status** **External ID** **Internet exposure** **Operating System**

`test-fsck` `Compute Instance` `Google Cloud` `Active` `555930431847658...` `Yes` `Linux`

Vulnerability [View in catalog](#)

Description

Moby is an open source container framework developed by Docker Inc. that is distributed as Docker, Mirantis Container Runtime, and various other downstream projects/products. The Moby daemon component (`dockerd`), which is developed as moby/moby is commonly referred to as Docker Swarm Mode, which is compiled in and delivered by default in `dockerd`, and is thus present in most major Moby downstreams, is a simple, built-in container orchestra...

[View More](#)

Name `CVE-2023-28841` **Severity** `Medium`

Score `6.8` **Exploitability Score** `3.2`

Link [nvd.nist.gov](#) **Published Date** `April 4th, 2023 at 3:15 PM`

Impact Score `4.0` **References** [View item](#)

Attack Complexity (CVSS V3) `High` **Privileges Required (CVSS V3)** `None`

Confidentiality Impact (CVSS V3) `High` **Integrity Impact (CVSS V3)** `None`

Example vulnerability detection with easy-to-follow remediation instructions

Based on the scan results, your security team will resolve identified vulnerabilities by deploying security patches, updating software versions, or re-configuring security settings, depending on recommendations in the vulnerability report.

After remediation, rescanning the target systems should take place to verify that the vulnerabilities have been successfully resolved.

Stage 8: Continuous monitoring

New vulnerabilities can always surface. A **vulnerability management program** ensures scans are scheduled at intervals to identify and address emerging threats promptly.

WIZ ACADEMY

What is Vulnerability Management? Steps and Components to Implement Vulnerability Management



Read more >

What vulnerabilities does a scan uncover?

These are common areas of weaknesses that a vulnerability scan typically identifies:

- **Network:** Open ports, weak passwords, misconfigured firewalls, and unauthorized devices or connections.
- **System:** Missing patches, outdated software, misconfigurations, and vulnerable operating systems.
- **Applications:** Security flaws, cross-site scripting and SQL injection vulnerabilities, and misconfigured settings.
- **Cloud-specific:** Misconfigured cloud services and settings and improper identity access and authentication.

The specific vulnerabilities uncovered will depend on the type of scan performed (e.g. network, application, database) and whether it's an internal or external scan.

Vulnerability scanning databases

Vulnerability scanners use a database of known vulnerabilities to identify weaknesses. One such database is the **National Vulnerability Database (NVD)**.

These databases contain information such as vulnerability severity, potential impact, and recommended mitigation techniques. The scanner compares its discoveries in the target environment and matches them with those in the database, then flags, reports, and provides remediation options to any matches.

Vulnerability scanning vs Penetration testing: what's the difference?

Vulnerability scanning is an automated process that identifies potential security weaknesses in systems, while penetration testing is a more comprehensive, manual approach that simulates real-world attacks to exploit vulnerabilities and assess an organization's overall security posture.

Action	Outcome	Input	Time	Frequency
Vulnerability scanning	A list of potential vulnerabilities	Automated	• Under an hour for a simple scan • Up to 72 hours for a complex scan	Up to daily
Penetration testing	Results of a real-world simulated cyberattack	Manual	Up to several weeks	At least once a year

Common vulnerability scanning challenges

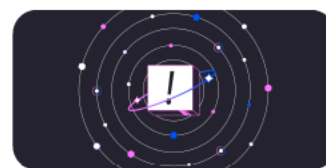
A vulnerability scan may be ineffective if these issues are present:

Challenge	Description
Resource sharing	Vulnerability scanning requires significant network bandwidth and computing resources. Production (in the IT environment) is also resource intensive. When both processes share resources provided by the organization's infrastructure, resource contention occurs, and can negatively impact the scan's efficiency.
False positives	The vulnerability scanning tool could incorrectly identify a non-existent vulnerability, wasting time and effort. For instance, a developer could be patching a dependency in the source code, and the tool might alert that malicious activity is taking place. Misconfiguring the vulnerability scanner usually leads to these kinds of false positives.
Alert Fatigue	Vulnerability scanning generates quintillions of alerts, making it overwhelming for the security team to painstakingly track and address each alert, and that can lead to neglecting critical vulnerabilities.
Siloed tooling	Using vulnerability scanning tools with other security solutions across different environments or departments can create data silos and distort vulnerability management. That can hinder collaboration and make it difficult to have an end-to-end view of the organization's security posture.
Inability to contextualize vulnerability impact	Vulnerability scanning tools may be ineffective for risk management as they're often ignorant of asset criticality, business processes, and system dependencies. They also likely won't understand the impact of vulnerabilities across individual organizations.
High ownership costs	Vulnerability scanning tools and the associated infrastructure can be expensive to procure, deploy, and maintain. Organizations may also need to invest in staff training and dedicated personnel employment. All of that translates to increased costs.
Ongoing maintenance efforts	Some vulnerability scanning solutions require agents to be installed on target systems for continuous scanning. Managing the installation, updates, and maintenance of these agents across many systems can be challenging and time consuming.
Blind spots	This occurs when vulnerabilities in certain assets are missed during scanning, and may be caused by a tool's inability to detect vulnerabilities on specific asset types, such as cloud infrastructure, mobile devices, or IoT devices.

Challenge	Description
Software development delays	Traditional vulnerability scanning practices require extensive scans and manual verification, causing delays in the development of applications and the release of software updates. These kinds of delays ultimately hurt an organization's bottom line.

WIZ ACADEMY

8 All-Too-Common Cloud Vulnerabilities

[Read more >](#)

Key features to look for in a vulnerability scanning tool

To effectively address and mitigate the challenges described above, choose a vulnerability scanning tool with the following key features:

1. Continuous scanning capability

Continuous monitoring is the last but crucial stage of vulnerability scanning. Choose a tool that can continuously scan and detect vulnerabilities as they emerge so your organization can be consistently vulnerability free.

2. Agentless approach

[Your vulnerability scanning tool should be agentless](#), eliminating the need to install and manage scanning agents on target systems. Such tools utilize network-

based scanning techniques, consume fewer resources, and erase the possibilities of incompatibility.

Pro tip

It's important to be able to scan virtual machines or containers even if the workload is offline. Security teams can remediate the vulnerability before the workload is online and effectively at risk.

But with an agent-based scanner, since an agent is part of the runtime of the workload, the scanning can only happen while the workload is online. This also applies for authenticated scanning, which means you can test applications in their ready-to-run configuration both in staging and production environments.

[Learn more](#) >

3. Risk-based prioritization



Example visualization of the risk-based context that a vulnerability scanner should provide

Choose a tool that provides risk-based prioritization of vulnerabilities, considering factors such as severity, exploitability, and asset criticality alongside elements such as external exposure, **cloud entitlements**, secrets, misconfigurations, and malware presence. A tool with this functionality can correlate your vulnerabilities that have numerous risk factors to mitigate the amount of alert fatigue you experience.

4. Cross-cloud/cross-technology support

The screenshot displays the Wiz Vulnerability Scanning interface. The top navigation bar includes the 'WIZ' logo and various menu items: 'All projects', 'Dashboard', 'Inventory', 'Issues', 'Explorer', 'Policies', 'Compliance', 'Reports', and 'Projects'. Below the navigation bar, there's a 'Technologies' tab and a 'Registry Container Images' link. The main section is titled 'Inventory' and shows a list of technologies with columns for Technology, Resource Count, Org. Usage, Type, and Status. The list includes items like AWS CodeCommit, AWS CodePipeline, AWS Recovery Point, Amazon CloudWatch Alarm, Amazon EBS Snapshot, Azure API Connection, Azure Action Group, Azure Alerts Management Smart Detector Ale..., Azure Application Insights, and Azure Arc-enabled Kubernetes. A sidebar on the left shows categories like Code, CI/CD & Management, Compute Platforms, Application & Data, Security, and Cloud Entitlements with their respective counts.

A cloud vulnerability scanner must be able to run across your entire technology inventory

Environments are becoming more hybridized and distributed. Select a technology-agnostic tool that can scan different storage environments and cloud providers, including AWS, GCP, Azure, OCI, and Alibaba Cloud, regardless of underlying OS or programming language to ensure software compatibility.

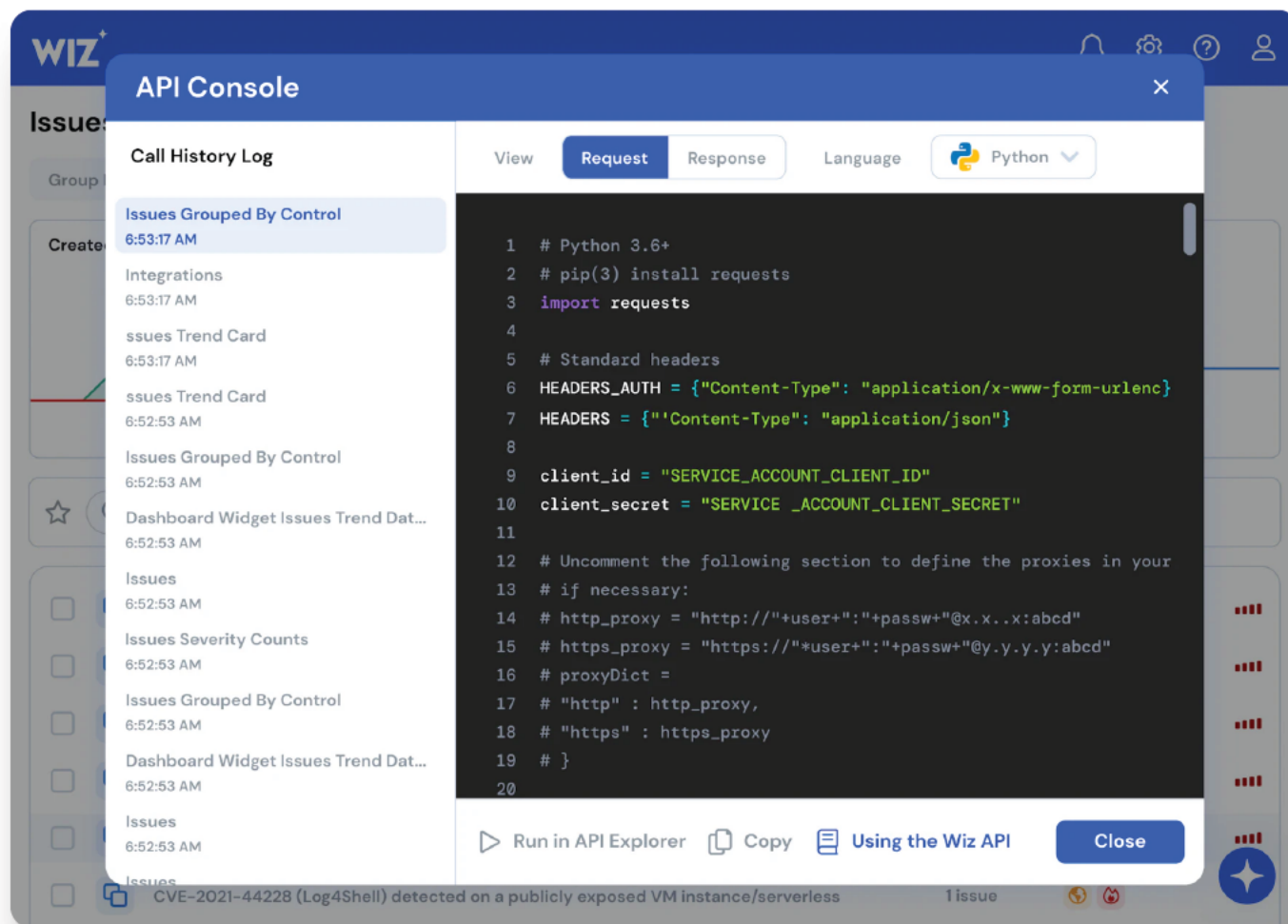
Pro tip

The cloud poses unique challenges that traditional vulnerability management solutions may struggle to address. **Cloud vulnerability management** is a proactive security solution that can keep up with the speed and scale of the cloud.

Traditional scanning tools were able to identify and remediate vulnerabilities but often flagged vulnerabilities that were non-critical and

irrelevant. Furthermore, traditional vulnerability management had a significant deficiency: context.

5. Scanning before deployment



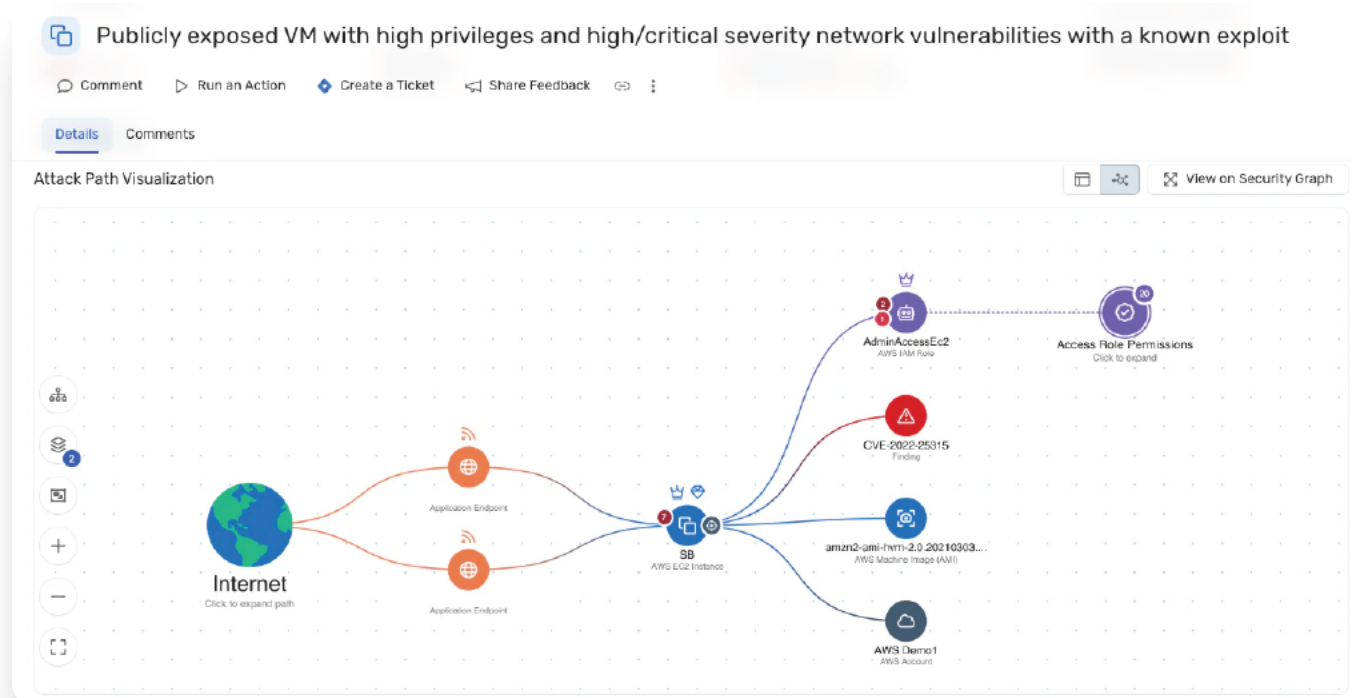
Vulnerability scanning should be extended into the CI/CD pipeline to scan VM and container images and catch issues before they ever reach production

Opt for a tool that can scan virtual machines (VMs) and containers and detect potential vulnerabilities in them before their deployment. This will help avoid spreading vulnerabilities across the entire production environment.

6. Comprehensive workload coverage

Your scanning tool needs to be able to simultaneously scan various systems and **workloads**—servers, endpoints, databases, and web applications—to allow for proactive and efficient vulnerability remediation.

7. Data-Based visualization reports



Visual representation of vulnerability data in various formats—such as tables, graphs, and charts—are key to decision making and remediation. The tool must provide that level of visualization in the scan results and make them easily shareable.

8. Integration

The tool must seamlessly integrate with tools for SIEM (Security Information and Event Management), log management, and SCM (Security Configuration Management) to enable better threat detection and **incident response**, and provide cohesive security management.

WIZ BLOG

Security industry call to action: we need a cloud vulnerability database

Read more >



Go beyond vulnerability scanning with vulnerability management

Wiz performs agentless vulnerability scanning across cloud environments based on multiple updated vulnerability databases and public sources to ensure accurate and up-to-date vulnerability detection. The scan analyzes a variety of metadata, including installed packages, programming language libraries, operating system information, and file hashes to identify vulnerabilities.

Wiz scans both cloud resources and code repositories without requiring elevated privileges, and prioritizes vulnerabilities based on contextual risk factors like external exposure and cloud entitlements. But a scan is only one step in [Wiz's vulnerability management solution](#).

“With our old platform, we were getting thousands of alerts for every one problem that we’d solve. Wiz allows us to understand vulnerabilities much more efficiently. Now, we can concentrate our efforts on problems rather than simply identifying them”.

Alex Steinleitner, President & CEO, Artisan

To experience first-hand how a unified vulnerability management solution can boost your organization's security posture, [request a live demo of Wiz](#). You'll get the opportunity to understand why and how using Wiz's unified vulnerability management solution can boost your organization's security posture.

Agentless Scanning = Complete Visibility Into Vulnerabilities

Learn why CISOs at the fastest growing companies choose Wiz to identify and remediate vulnerabilities in their cloud environments.

Get a demo >



Explore more on this topic

Fundamentals of Code Security

- [Code Security](#) >
- [Cloud Application Security: Basics and Best Practices](#) >
- [The Secure Software Development Framework \(SSDF\)](#) >
- [What is Security by Design?](#) >
- [What Is Secure Coding? Overview and Best Practices](#) >
- [Secure SDLC](#) >

PLATFORM

[Wiz CNAPP](#)[Wiz Code](#)[Wiz Cloud](#)[Wiz Defend](#)[Integrations](#)[Environments](#)[Documentation](#)

LEARN

[Customer stories](#)[Train Your Team For Cloud](#)[Blog](#)[CloudSec Academy](#)[Resources Center](#)[Cloud threat landscape](#)[Cloud Security Assessment](#)[Vulnerability Database](#)

COMPANY

[About Wiz](#)[Join the team](#)[Newsroom](#)[Events](#)[Contact us](#)[Trust Center](#)[Our partners](#)

 English (US) ▾



[Status](#)

[Privacy Policy](#)

[Your California Privacy Rights](#)

[Terms of Use](#)

[Modern Slavery Statement](#)

[Cookie Settings](#)

© 2025 Wiz, Inc.