A comprehensive benchmark of network vulnerability scanners 2024

Contents

Benchmark overview

- The analysis included the following network vulnerability scanners:
  - Nessus Professional
  - Nmap vulnerability scripts
  - Nuclei
  - OpenVAS
  - Pentest-Tools.com Network Vulnerability Scanner
  - Qualys
  - Rapid7 Nexpose
- The tests were performed against **all 167** vulhub vulnerable environments.
- vulhub was used to keep the evaluation both comprehensive and impartial and make sure anyone can independently validate the results. vulhub, an openly accessible repository, offers a wide array of vulnerable environments, making it ideal for testing security tools under specific conditions.
- For clarity, the analyzed CVEs are categorized into those detectable remotely (128 environments) and those that are not (39 environments).
- For those wishing to independently confirm the findings, it is essential to acknowledge that all scanners were updated with the latest detections as of January 2024.
- Most tools were initiated with their default settings, targeting the entire TCP port range (1-65535).
- All scanning activities unfolded throughout January 2024.
- To guarantee a uniform and impartial evaluation, the analysis relied on two main performance indicators:
  - Detection availability
  - Detection accuracy

Key findings from the benchmark

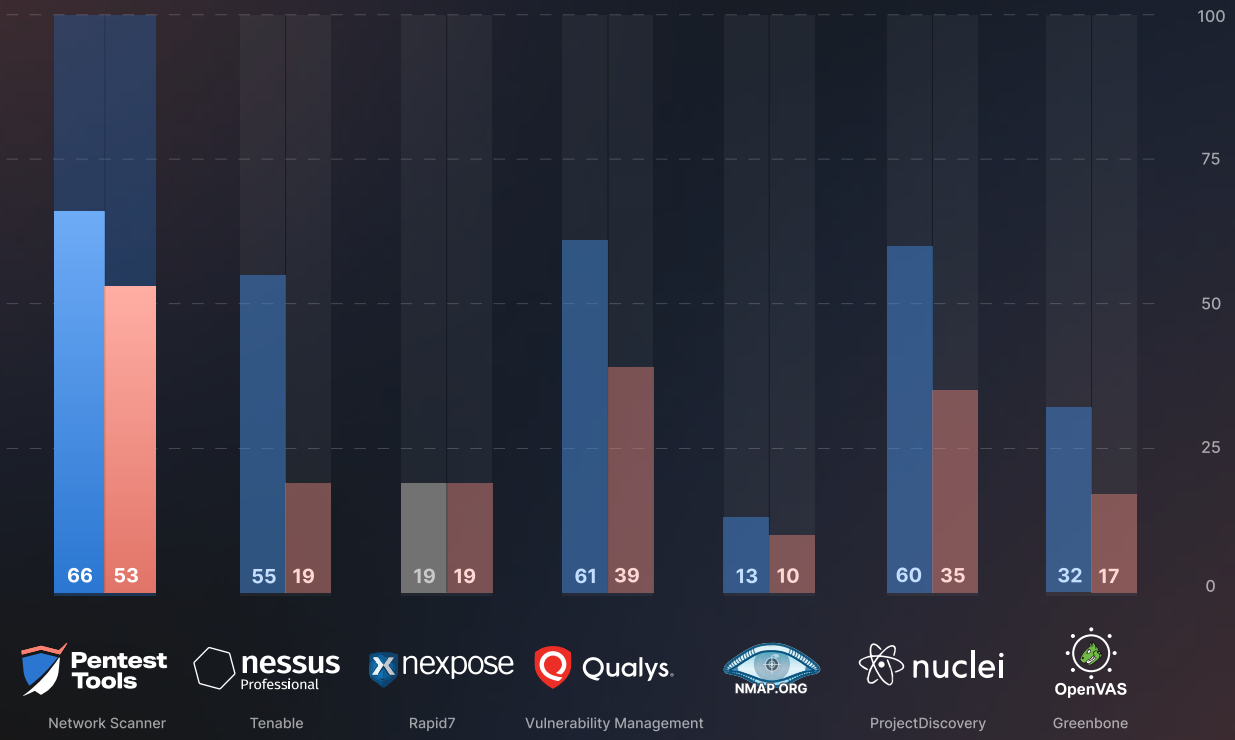
The benchmark reveals a similar level of detection availability among the major commercial key players (except for Rapid7 Nexpose, where it wasn't possible to differentiate between local and remote checks in their vulnerability database).

This is relevant in the context of commercial vulnerability scanning solutions stating that they have detections for the majority of vulnerabilities in testing environments.

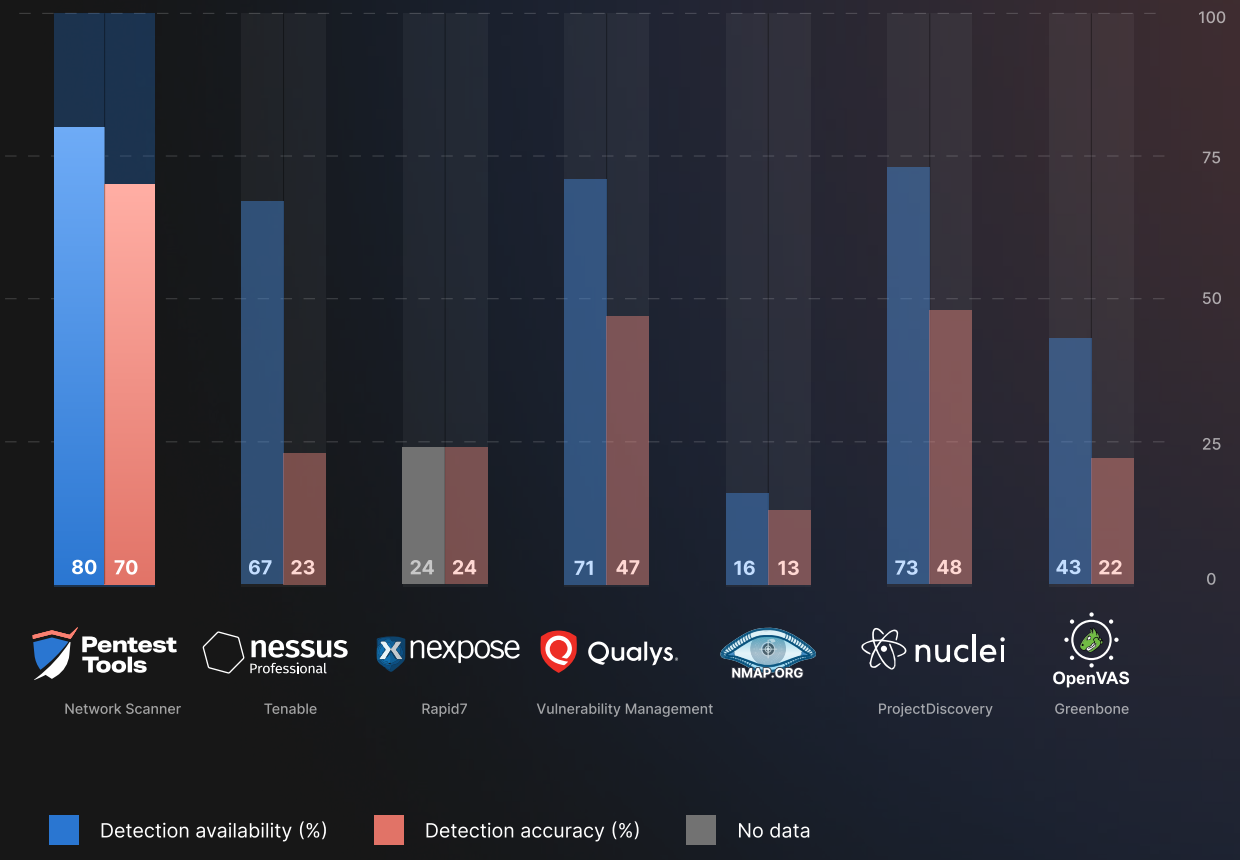
A notable disparity exists between the detection availability and the actual accuracy of certain tools. The most pronounced discrepancy was observed in Nessus, which reports having detections for 55.09% of all vulnerable environments tested but only successfully identified 18.56% of them. Similarly, it claims to have templates for 67.19% of all remotely detectable vulnerabilities, yet it only accurately detects 22.66% of these. Following this, both Qualys and Nuclei show lower variance, with their actual detection rates being about 25% lower than what their vulnerability database suggests.

A subtle shift is noted between the overall and remotely detectable classifications: for all templates, Qualys secures the second position, with Nuclei following in third. However, when focusing solely on vulnerabilities that can be detected remotely, Nuclei moves up to second place, pushing Qualys down to third. This indicates Nuclei has a slightly broader scope in terms of detection for vulnerabilities that can be remotely detected.

Benchmark results against all Vulhub vulnerabilities



Benchmark results against all Vulhub remotely detectable vulnerabilities



Detection availability for all environments was calculated as

$$= \frac{\text{count of the detection existence}}{\text{total number of vulnerabilities}} * 100$$

Detection availability for vulnerabilities that can be detected remotely was calculated as

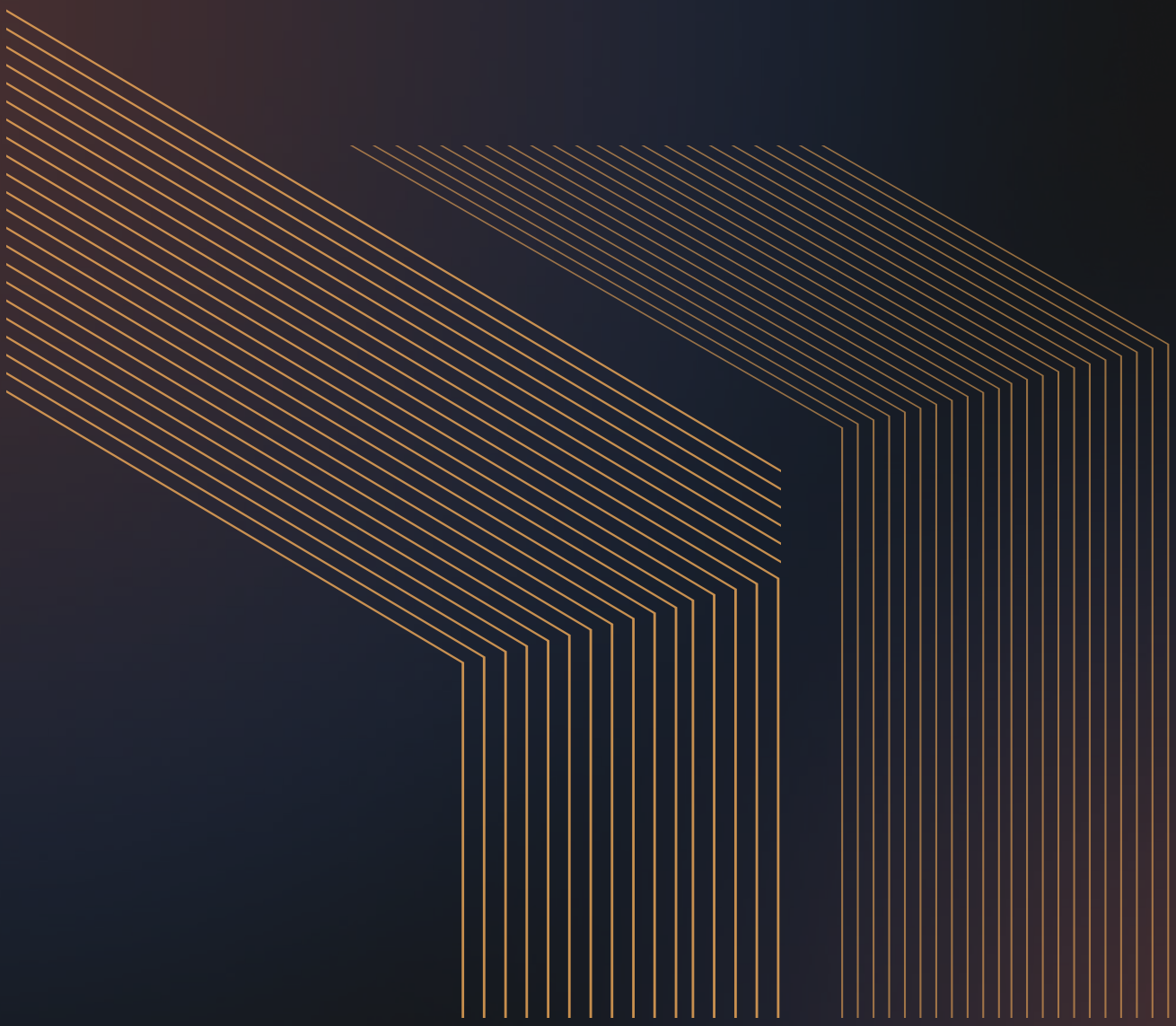
$$= \frac{\text{count of the detection existence}}{\text{total number of vulnerabilities analyzed that can be detected remotely}} * 100$$

Detection accuracy for all environments was calculated as

$$= \frac{\text{count of the vulnerabilities detected}}{\text{total number of vulnerabilities}} * 100$$

Detection accuracy for vulnerabilities that can be detected remotely was calculated as

$$= \frac{\text{count of the vulnerabilities detected}}{\text{total number of vulnerabilities analyzed that can be detected remotely}} * 100$$



Why this benchmark exists

Ethical hackers and other security specialists have to invest considerable time and effort to compare tools using information from vendors, conversations with peers, forums, and communities, and their own experience - if available.

For instance, one of the few extensive benchmarks in the industry dates from 2017, when Shay Chen evaluated 10 web application vulnerability scanners, publishing the results after a 2-year long effort. Similarly, benchmarks for network vulnerability scanners are rare and far in between, for several noticeable reasons.

One of the primary difficulties is the rapid evolution of cybersecurity threats, which means that vulnerability scanners must continuously update to detect new vulnerabilities. This dynamic nature of threats makes it hard to establish a static benchmark that remains relevant for an extended period. What's more, the diversity in vulnerabilities requires a benchmark to be adaptable and comprehensive enough to cover a wide range of scenarios, making it challenging to design a one-size-fits-all evaluation metric.

Despite the moving pieces a project like this involves, the cybersecurity community is at a stage where it needs this benchmark.

Target network vulnerability scanners

Network vulnerability scanners are complex tools designed to scan systems accessible through a network (e.g. servers, workstations, etc.) for security vulnerabilities and weaknesses.

Network vulnerability scanners sit in two primary categories based on their **licensing type**: **open-source** and **commercial**.

Open-source network vulnerability scanners,

such as OpenVAS, Nuclei, and Nmap vulnerability scripts, are freely available and can be modified and distributed under their respective licenses.

These scanners are particularly appealing to organizations with strong technical teams capable of customizing and extending the tools to fit specific requirements.

Key features often include comprehensive scanning capabilities, community-supported updates for the latest vulnerabilities, and flexibility in integrating with other security tools. However, they often require more setup and maintenance effort compared to commercial products.

Commercial network vulnerability scanners,

such as the Pentest-Tools.com Network Vulnerability Scanner, Nessus Professional, Rapid7 Nexpose, and Qualys, are proprietary tools that come with a price tag.

These scanners are known for their ease of use, professional support, and continuous updates the vendor provides.

They often feature a more user-friendly interface, extensive vulnerability databases, and advanced scanning options. Organizations typically choose commercial scanners because they are ready-to-use, reliable, regularly updated, offloading effort their teams need to use on more business-critical work.

Based on their **deployment model**, network vulnerability scanners also fall into two categories: **on-premise** and **cloud-based**.

On-premise network vulnerability scanners,

such as Nessus Professional, Rapid7 Nexpose, and OpenVAS, are security tools that are installed and run within an organization's local network environment. Since data does not leave the organization's premises, there's a higher level of control and privacy, crucial for sensitive or regulated data.

Cloud-based network vulnerability scanners,

such as the Pentest-Tools.com Network Vulnerability Scanner, Tenable.io, and Rapid7 InsightVM, are hosted on the cloud, enabling organizations to scan their networks without the need for extensive on-premise hardware.

Key features include almost zero maintenance costs, no setup time, real-time updates, scalability, reduced infrastructure costs, and the ability to scan public, private, and hybrid cloud environments.

Each type of scanner has its own set of advantages and considerations, and a security specialist will likely base their choice on factors such as budget, technical expertise, specific organizational needs, and the complexity of the environment to be scanned.

Selecting benchmark criteria and why they matter

Defining benchmark criteria for evaluating tools like network vulnerability scanners, especially in cybersecurity, is essential for several reasons.

First, clear, transparent criteria ensure a fair and standardized comparison across different tools, whether they are open-source or commercial. This is why this benchmark focuses on remote detections. Here is the full picture behind this choice.

In the vulnerability scanning space, the distinction between local checks and remote checks is similar to the difference between having an inside scoop versus observing from the outside.

Local checks require access to the scanned system, including credentials necessary to do a deep dive into the system's health. This level of access allows for a thorough inspection of the operating system, installed software, and configurations for vulnerabilities that might be invisible from the outside.

Remote checks don't require direct access to the system but, instead, scan network services exposed to them. This means they can quickly identify potential vulnerabilities like open ports or services susceptible to known exploits without needing insider access.

In the current digital landscape, the majority of infrastructure sits behind Intrusion Prevention Systems (IPSs) and firewalls, which detect and block malicious packets. This creates a challenge for scanners that use remote checks as they may miss vulnerabilities because of the blind spots these protective measures create. Despite these defenses, a target remains at risk of compromise when determined and skilled attackers focus on it.

Local and remote scans complement each other and paint a complete picture of a system's security posture, blending the depth of local insights with the breadth of remote observation.

This benchmark focuses exclusively on remote checks or assessments from a black-box perspective. While certain tools might be capable of local checks for the analyzed CVEs, this aspect is not within the scope of this benchmark.

The emphasis on remote detections aligns with simulating an external attacker's viewpoint and offers a realistic assessment of the attack surface visible and accessible from the outside. Vulnerabilities that can be exploited remotely are particularly attractive targets for attackers and pose a substantial risk to organizations.

By spotlighting remote detections, the aim is to pinpoint these high-risk exposed vulnerabilities which, when addressed, strengthen the network's perimeter against unauthorized intrusions.

To maintain uniform and objective metrics, two specific performance indicators shape the contents of this benchmark:

- **Detection availability** - a scanner reports, through a vulnerability database, that it has detection for a specific vulnerability
- **Detection accuracy** - a scanner identifies the specific vulnerability

Benchmark methodology

To streamline environment deployment and ensure system integrity, the analysis strategically involves a single type of environment: **virtual machines**.

These machines were hosted on the Vultr cloud platform, protected by a firewall that operates on an IP whitelist mechanism. This approach provided the necessary safeguard against unauthorized access while keeping the setup process straightforward.

This benchmark uses data from vulhub to keep the assessment thorough, unbiased, and to open it up to independent testing and validation. vulhub is a publicly accessible repository that houses a comprehensive collection of environments known for their vulnerabilities, offering a rich resource for testing security tools and methodologies in a controlled setting.

Important: If you are interested in verifying the results independently, please note that all scanners were updated to the latest plugins available as of January 2024. Also, most of the tools were started with default configurations, selecting the full port range (TCP 1-65535).

As part of this carefully structured setup, every vulnerable Docker container available was deployed on the Vultr cloud platform in **December 2023**. This deployment included **167 distinct environments**, spread across **17 instances**, with each instance hosting around **10 vulnerable services**.

To manage this complex array of services efficiently, each instance was equipped with its own Docker Compose file, allowing for streamlined configuration and management of the services. This meticulous setup allowed a detailed exploration and evaluation of the included network scanners and provided a solid foundation for assessing their effectiveness.

Consequently, the list of CVEs analyzed is categorized into two main groups: those that are remotely detectable and those that are not.

Under the "**remotely detectable**" category, the benchmark also includes vulnerabilities that require specific conditions, such as:

- **Path Traversal** (e.g. CVE-2021-43798) - arbitrary files can be read from the server
- **Broken Access Control** (e.g. CVE-2023-22515) - adding a user to the application
- **Arbitrary Command Injection** (e.g CVE-2020-35476, CVE-2022-46169) - command injection in arbitrary parameters
- **OGNL Injection** (e.g CVE-2021-26084) - assessing expressions that haven't been verified on the value stack, which enables to alter system variables or run commands
- **JNDI injection** (e.g CVE-2021-44228) - exploits the Java Naming and Directory Interface to execute arbitrary code
- **Arbitrary File Write** (e.g CVE-2016-3088) - remote code execution via a file upload
- **Server-Side Request Forgery** (e.g CVE-2021-21311) - it causes the server-side application to make requests to an unintended location
- **Authentication Bypass** (e.g CVE-2020-17526) - default session keys
- **SQL Injection** (e.g CVE-2014-3704) - tampering queries that an application makes to its database
- **Prototype Pollution** (e.g CVE-2019-7609) - injection of properties into a programming language's prototype object, potentially leading to remote code execution
- **Information Disclosure** (e.g CVE-2023-28432) - obtaining environment variables or disclosing sensitive information

Under the "**not remotely detectable**" category, the benchmark includes vulnerabilities that require specific conditions, such as:

- **Authentication requirements** (e.g. CVE-2022-41678, CVE-2019-14234, etc.) - Credential brute force needs to be performed, in order to obtain valid credentials
- **User interaction**, such as 1-click RCEs (CVE-2019-6341), file uploads (CVE-2019-6339), CVE-2023-38633, Electron App vulnerabilities (CVE-2018-15685, CVE-2018-1000006) - these are hard to automate because the tool needs to interact with the application to trigger the vulnerability
- **Privilege escalation instances** (e.g. CVE-2021-4034) - which are impossible to detect, because they require access to the target
- **Vulnerabilities that need a custom setup** - which makes automation of exploitation difficult or impossible as there are too many variables
- **Created Django custom vulnerable pages** (CVE-2021-35042, CVE-2020-9402, CVE-2018-1273, etc.) - these are some custom Django pages created to demonstrate the vulnerabilities (/vuln/ endpoint)
- **Complex scenarios requiring additional information like usernames** - which is challenging to obtain these and the process involves OSINT
- **Local vulnerabilities** (CVE-2020-29599) - that require local access to a password-protected PDF
- **Vulnerabilities that require fuzzing**, like ImageMagick (CVE-2022-44268), Ghostscript (CVE-2018-16509), ffmpeg (CVE-2016-1897) - which are challenging to automate but can be achievable. The process involves having the video payload prepared in advance or automating its creation through scripting. The next step requires locating the input form on the website, which may involve fuzzing as part of the challenge. Upon finding the form, the payload is sent. This vulnerability might be remotely detectable, though it falls outside the scope of the tools selected for this evaluation.

Benchmark results

When looking at the numbers, it is important to consider that the number of identified vulnerabilities does not certify the quality of a vulnerability scanner's performance.

The vulnerabilities included in this analysis are a very small subset of the coverage each vulnerability scanner is capable of. Factors such as user-friendliness, the ability to integrate with other systems, or the quality of support services can be just as relevant for an organization's context, but there is no impartial way to compare them.

All the data behind the results in this benchmark are in this publicly available Google Sheet: Public Comparison - Network vulnerability scanners benchmark data - 2024

Benchmark results against all Vulhub vulnerabilities (167)

	Pentest-Tools.com Network Vulnerability Scanner	Nessus	Rapid7 Nexpose	Qualys	Nmap	Nuclei	OpenVAS
Detection availability (%)	**65.87**	55.09	18.56	61.08	12.57	59.88	32.32
Count	**111**	93	31	103	21	101	55
Detection accuracy (%)	**53.29**	18.56	18.56	39.92	9.58	35.33	16.77
Count	**90**	31	31	65	16	60	28

Benchmark results against Vulhub remotely detectable vulnerabilities (128)

	Pentest-Tools.com Network Vulnerability Scanner	Nessus	Rapid7 Nexpose	Qualys	Nmap	Nuclei	OpenVAS
Detection availability (%)	**80.47**	67.19	24.22	71.09	16.41	73.44	42.97
Count	**103**	86	31	91	21	94	55
Detection accuracy (%)	**70.31**	22.66	24.22	46.88	12.50	47.66	21.88
Count	**90**	29	31	60	16	61	28

Insights from the benchmarking process

- With Rapid7 Nexpose, some CVEs were found in their Vulnerability & Exploit database, but it was challenging to determine whether they were detected remotely, resulting in a "**?**" in their "has remote detection" column. If the vulnerability was detected, a "✓" was automatically added to the "has remote detection" column. Therefore, for Rapid7 Nexpose, the detection availability is the same value as detection accuracy.

- Some containers presented issues, such as being broken (e.g. CVE-2020-17526 - redirect loop after injecting the session cookie) or having a faulty Dockerfile (e.g. CVE-2017-17405, CVE-2017-5223).

- All vulnerable environments were deployed on a Linux instance. Consequently, if a tool's detection capability for a vulnerability was only available for a Windows instance, its corresponding "has remote detection" column features an "×".

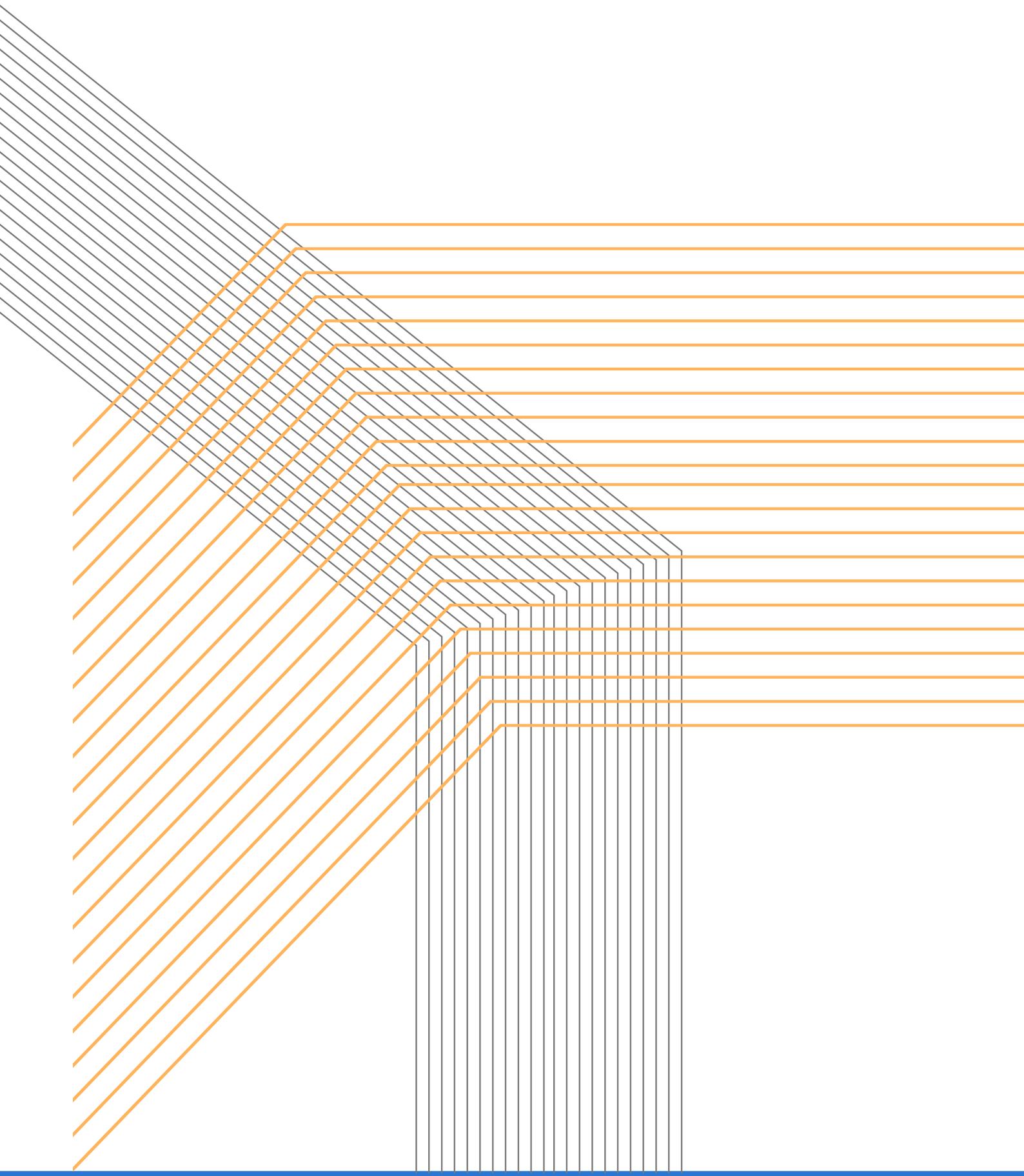
- When using Nessus and OpenVAS with the default full port range for port discovery, there were scans in which they didn't discover all the open ports. Nmap successfully identified all the open ports, and a custom list of these was then configured in the interface of Nessus and OpenVAS.

- An unusual error appeared when attempting to directly scan the IP from Nessus; in some cases, the DNS resolution for the vultrusercontent.com domain (default DNS for Vultr hosts) pointed to localhost, leading to scans that showed no results. To resolve this issue, DNS hostnames were assigned to each benchmark machine within the pentest-ground.com domain (e.g., `bench1.pentest-ground.com`, `bench2.pentest-ground.com`, etc.).

- Some Nuclei templates, such as CVE-2017-17562, fall into the fuzz category and are excluded by default. Therefore, activating the "fuzz" option is necessary to detect these vulnerabilities.

- In some instances, Nmap scripts resulted in "Script execution failed,"

leading to undetected vulnerabilities despite the availability of a vulnerability script. Nevertheless, Nmap's vulners script can detect certain vulnerabilities based on the reported version, even if there's no dedicated script for those vulnerabilities.



Network vulnerability scanners benchmark FAQs

1. **What accounts for the higher detection rate of the Pentest-Tools.com Network Vulnerability Scanner compared to Nuclei and OpenVAS, despite it incorporating these scanning engines?**

   The Pentest-Tools.com Network Vulnerability Scanner uses a comprehensive approach to detection by combining four distinct engines: version-based detection, Sniper, Nuclei, and OpenVAS, which collectively offer an extensive range of vulnerability detection capabilities. Before initiating the vulnerability scanning phase, the Pentest-Tools.com Network Vulnerability Scanner conducts reconnaissance to identify open ports and services. It then uses this information to inform the scanning engines. Additionally, the Network Vulnerability Scanner is enhanced with a suite of custom detections developed by the security research team at Pentest-Tools.com, which further broadens its ability to identify vulnerabilities.

2. **What differentiates the Nuclei engine in the Pentest-Tools.com Network Vulnerability Scanner from the standalone Nuclei release?**

   The Nuclei engine in the Pentest-Tools.com Network Vulnerability Scanner integrates a range of custom templates and fixes that are not available on the standalone Nuclei release.

3. **Why does this benchmark focus only on remote detections?**

   Focusing on remote detections is more effective for identifying and mitigating high-risk vulnerabilities, which strengthens the network's defenses against unauthorized intrusion.

   Additionally, the necessity for local access, credentials, or the deployment of agents on individual devices complicates matters, given the reluctance to provide broad access to external parties.

Europe, Romania, Bucharest
48 Bvd. Iancu de Hunedoara

E: support@pentest-tools.com
pentest-tools.com

Join our community of ethical hackers!

Pentest Tools