

Lab - Vulnerability Scanning with Kali Tools

Objectives

In this lab, you will explore network vulnerability scanning tools and use them to perform a vulnerability scan on a target host.

- Perform network scans with Nmap
- Use Greenbone Vulnerability Management to perform a vulnerability scan.

Background / Scenario

In a previous lab, you used Nmap to enumerate a host computer that was creating unusual traffic on the network. In this lab, you will use Nmap and Greenbone Vulnerability Management (GVM) to scan the system to identify potential vulnerabilities.

Required Resources

= Kali VM customized for the Ethical Hacker course

= Internet access

Step 1: Start and login to the Kali virtual machine. a. Start and log into the Kali virtual machine. b. Start a terminal session.

```
(kali㉿kali)-[~]  
$ ping -c5 10.6.6.23  
PING 10.6.6.23 (10.6.6.23) 56(84) bytes of data.  
64 bytes from 10.6.6.23: icmp_seq=1 ttl=64 time=0.059 ms  
64 bytes from 10.6.6.23: icmp_seq=2 ttl=64 time=0.051 ms  
64 bytes from 10.6.6.23: icmp_seq=3 ttl=64 time=0.103 ms  
64 bytes from 10.6.6.23: icmp_seq=4 ttl=64 time=0.059 ms  
64 bytes from 10.6.6.23: icmp_seq=5 ttl=64 time=0.053 ms  
  
— 10.6.6.23 ping statistics —  
5 packets transmitted, 5 received, 0% packet loss, time 4127ms  
rtt min/avg/max/mdev = 0.051/0.065/0.103/0.019 ms
```

Identify open ports and services. Review the results of a Nmap scan on the host with the IP address 10.6.6.23

```
(kali㉿kali)-[~]  
$ nmap -sV 10.6.6.23  
Starting Nmap 7.94 ( https://nmap.org ) at 2025-10-08 13:20 UTC  
Nmap scan report for gravemind.vm (10.6.6.23)  
Host is up (0.00018s latency).  
Not shown: 994 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 3.0.3  
22/tcp    open  ssh          OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)  
53/tcp    open  domain      ISC BIND 9.11.5-P4-5.1+deb10u5 (Debian Linux)  
80/tcp    open  http         nginx 1.14.2  
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)  
Service Info: Host: GRAVEMIND; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Identify the operating system running on the target computer using the nmap -O command.

```
(kali㉿kali)-[~]
└─$ sudo nmap -O 10.6.6.23
[sudo] password for kali:
Starting Nmap 7.94 ( https://nmap.org ) at 2025-10-08 13:21 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00010s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 02:42:0A:06:06:17 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.96 seconds
```

Use the Nmap Vulners script to scan for vulnerabilities.

Use the nmap -sV --script vulners command to launch the vulners script. The syntax for the command is nmap -sV -sV --script vulners [--script-args mincvss=] where the script argument mincvss restricts the output to only those CVEs that have a higher CVSS score than the one specified in the argument.

```
(kali㉿kali)-[~]
└─$ nmap -sV --script vulners --script-args mincvss=4 10.6.6.23
Starting Nmap 7.94 ( https://nmap.org ) at 2025-10-08 13:22 UTC
Nmap scan report for gravemind.vm (10.6.6.23)
Host is up (0.00012s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 3.0.3
| vulners:
| vsftpd 3.0.3:
|   CVE-2021-30047  7.5  https://vulners.com/cve/CVE-2021-30047
|   CVE-2021-3618  7.4  https://vulners.com/cve/CVE-2021-3618
22/tcp    open  ssh          OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| vulners:
| cpe:/a:openbsd:openssh:7.9p1:
|   PACKETSTORM:173661  9.8  https://vulners.com/packetstorm/PACKETSTORM:173661 *EXPLOIT*
|   F0979183-AE88-53B4-86CF-3AF0523F3807  9.8  https://vulners.com/githubexploit/F0979183-AE88-53B4-86CF-3AF0523F3807 *EXPLOIT*
|   CVE-2023-38408  9.8  https://vulners.com/cve/CVE-2023-38408
|   B8190CDB-3EB9-5631-9828-8064A1575B23  9.8  https://vulners.com/githubexploit/B8190CDB-3EB9-5631-9828-8064A1575B23 *EXPLOIT*
28-8064A1575B23 *EXPLOIT*
|   8FC9C5AB-3968-5F3C-825E-E8DB5379A623  9.8  https://vulners.com/githubexploit/8FC9C5AB-3968-5F3C-825E-E8DB5379A623 *EXPLOIT*
5E-E8DB5379A623 *EXPLOIT*
|   8AD01159-548E-546E-AA87-2DE89F3927EC  9.8  https://vulners.com/githubexploit/8AD01159-548E-546E-AA87-2DE89F3927EC *EXPLOIT*
87-2DE89F3927EC *EXPLOIT*
|   2227729D-6700-5C8F-8930-1EEAFD4B9FF0  9.8  https://vulners.com/githubexploit/2227729D-6700-5C8F-8930-1EEAFD4B9FF0 *EXPLOIT*
30-1EEAFD4B9FF0 *EXPLOIT*
|   0221525F-07F5-5790-912D-F4B9E2D1B587  9.8  https://vulners.com/githubexploit/0221525F-07F5-5790-912D-F4B9E2D1B587 *EXPLOIT*
2D-F4B9E2D1B587 *EXPLOIT*
|   BA3887BD-F579-53B1-A4A4-FF49E9531C0  8.1  https://vulners.com/githubexploit/BA3887BD-F579-53B1-A4A4-FF49E9531C0 *EXPLOIT*
A4-FF49E9531C0 *EXPLOIT*
|   4FB01B00-F993-5CAF-BD57-D7E290D10C1F  8.1  https://vulners.com/githubexploit/4FB01B00-F993-5CAF-BD57-D7E290D10C1F *EXPLOIT*
57-D7E290D10C1F *EXPLOIT*
|   CVE-2020-15778  7.8  https://vulners.com/cve/CVE-2020-15778
|   CVE-2019-16905  7.8  https://vulners.com/cve/CVE-2019-16905
|   C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3  7.8  https://vulners.com/githubexploit/C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3 *EXPLOIT*
EE-0DAF45EEFFE3 *EXPLOIT*
|   2E719186-2FED-58A8-A150-762EFBAA523  7.8  https://vulners.com/gitee/2E719186-2FED-58A8-A150-762EFBAA523
```

GVM is part of the Open Source Vulnerability Management suite of products produced by Greenbone Networks GmbH. The GVM scanner is one of the most widely used open-source vulnerability scanners. Unlike Nmap, GVM uses a graphical user interface to initiate scans and report vulnerability scan results.

Step 1: Verify the GVM Product Installation.

Before beginning any scan, it is important to verify that GVM is correctly installed and that the files it uses to identify vulnerabilities are up-to-date. a. Verify the setup of the GVM service using the sudo gvm-check-setup command. This command verifies that the setup completed correctly and the

necessary files are available. The verification will flag any issues that need fixing and will provide the commands to use to fix the issues

```
➔ sudo gvm-check-setup
gvm-check-setup 22.5.0
Test completeness and readiness of GVM-22.5.0
Step 1: Checking OpenVAS (Scanner) ...
  OK: OpenVAS Scanner is present in version 22.7.3.
  OK: Notus Scanner is present in version 22.5.0.
  OK: Server CA Certificate is present as /var/lib/gvm/CA/servercert.pem.
Checking permissions of /var/lib/openvas/gnupg/*
  OK: gvm owns all files in /var/lib/openvas/gnupg
  OK: Redis-server is present.
  OK: scanner (db_address setting) is configured properly using the redis-server socket: /var/run/redis-openvas/redis-server.sock
  OK: the mqtt_server_url is defined in /etc/openvas/openvas.conf
  OK: gvm owns all files in /var/lib/openvas/plugins
  OK: NVT collection in /var/lib/openvas/plugins contains 86225 NVTs.
  OK: The notus directory /var/lib/notus/products contains 440 NVTs.
Checking that the obsolete redis database has been removed
  OK: No old Redis DB
  OK: ospd-openvas service is active.
  OK: ospd-OpenVAS is present in version 22.5.3.
Step 2: Checking GVM Manager ...
  OK: GVM Manager (gvm) is present in version 22.5.5.
Step 3: Checking Certificates ...
  OK: GVM client certificate is valid and present as /var/lib/gvm/CA/clientcert.pem.
  OK: Your GVM certificate infrastructure passed validation.
Step 4: Checking data ...
  OK: SCAP data found in /var/lib/gvm/scap-data.
  OK: CERT data found in /var/lib/gvm/cert-data.
Step 5: Checking Postgresql DB and user ...
  Starting postgresql service
  OK: Postgresql version and default port are OK.
  OK: ospd-openvas is active.
  OK: gvm | gvm | UTF8 | en_US.UTF-8 | en_US.UTF-8 | | libc | |
16435|pg-gvm|10|2200|f|22.5||
  OK: At least one user exists.
Step 6: Checking Greenbone Security Assistant (GSA) ...
  OK: Greenbone Security Assistant is present in version 22.05.1-git.
Step 7: Checking if GVM services are up and running ...
  Starting gvm service
  Waiting for gvm service
  OK: gvm service is active.
  Starting gsad service
  Waiting for gsad service
  OK: gsad service is active.
Step 8: Checking few other requirements ...
```

Just for this activity, stop the GVM service so you can observe the startup output.

```
(kali@kali)~$ sudo gvm-stop
[+] Stopping GVM services
o gsad.service - Greenbone Security Assistant daemon (gsad)
  Loaded: loaded (/lib/systemd/system/gsad.service; disabled; preset: disabled)
  Active: inactive (dead)
  Docs: man:gsad(8)
        https://www.greenbone.net

Oct 08 13:32:21 Kali systemd[1]: Starting gsad.service - Greenbone Security Assistant daemon (gsad)...
Oct 08 13:32:21 Kali systemd[1]: Started gsad.service - Greenbone Security Assistant daemon (gsad)...
Oct 08 13:40:30 Kali systemd[1]: Stopping gsad.service - Greenbone Security Assistant daemon (gsad)...
Oct 08 13:40:30 Kali systemd[1]: gsad.service: Deactivated successfully.
Oct 08 13:40:30 Kali systemd[1]: Stopped gsad.service - Greenbone Security Assistant daemon (gsad).

o gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)
  Loaded: loaded (/lib/systemd/system/gvmd.service; disabled; preset: disabled)
  Active: inactive (dead)
  Docs: man:gvmd(8)

Oct 08 13:32:09 Kali systemd[1]: Starting gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)...
Oct 08 13:32:09 Kali systemd[1]: gvmd.service: Can't open PID file /run/gvmd/gvmd.pid (yet?) after start: No such file or directory
Oct 08 13:32:10 Kali systemd[1]: Started gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)...
Oct 08 13:40:30 Kali systemd[1]: Stopping gvmd.service - Greenbone Vulnerability Manager daemon (gvmd)...
Oct 08 13:40:30 Kali systemd[1]: gvmd.service: Deactivated successfully.
Oct 08 13:40:30 Kali systemd[1]: Stopped gvmd.service - Greenbone Vulnerability Manager daemon (gvmd).
Oct 08 13:40:30 Kali systemd[1]: gvmd.service: Consumed 1.510s CPU time.

o ospd-openvas.service - OSPd Wrapper for the OpenVAS Scanner (ospd-openvas)
  Loaded: loaded (/lib/systemd/system/ospd-openvas.service; disabled; preset: disabled)
  Active: inactive (dead)
  Docs: man:ospd-openvas(8)
        man:openvas(8)
```

Start the GVM scanner using the `sudo gvm-start` command. You can also access the `gvm-start` script

using the Applications menu on the Kali desktop, Kali -> 02-Vulnerability Analysis -> gvm start. It is possible that GVM may already be running as a result of the check setup process.

The output of the command should be similar to what is shown below. At the end of the output, a message that the scanner is loading in Firefox will appear.



Greenbone

Sign in to your account

Username

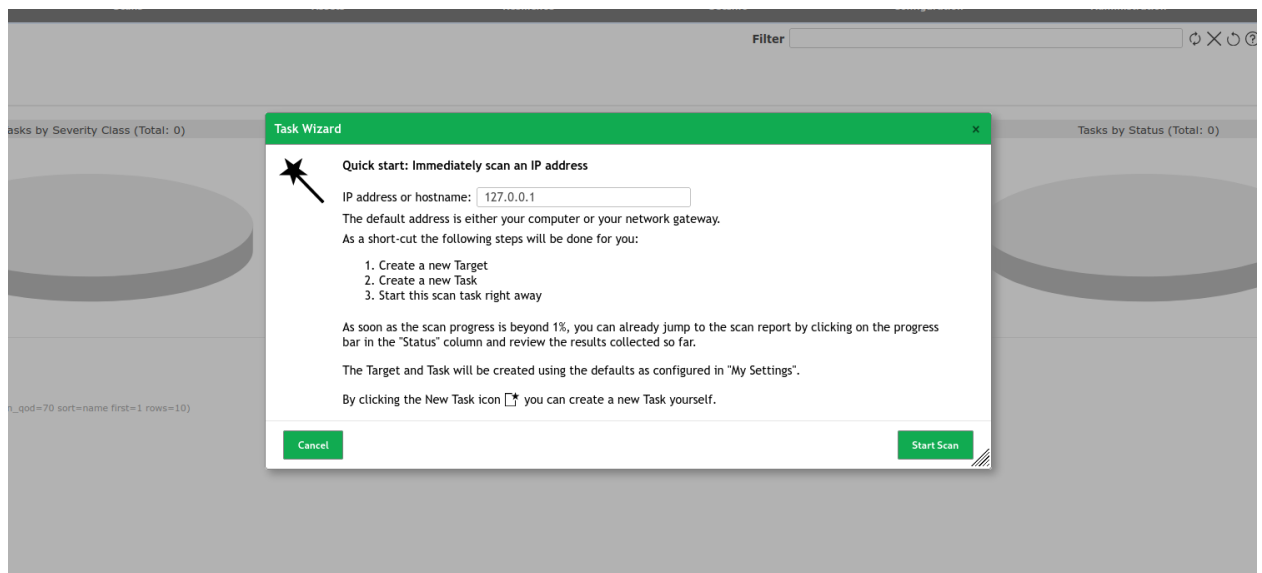
Password

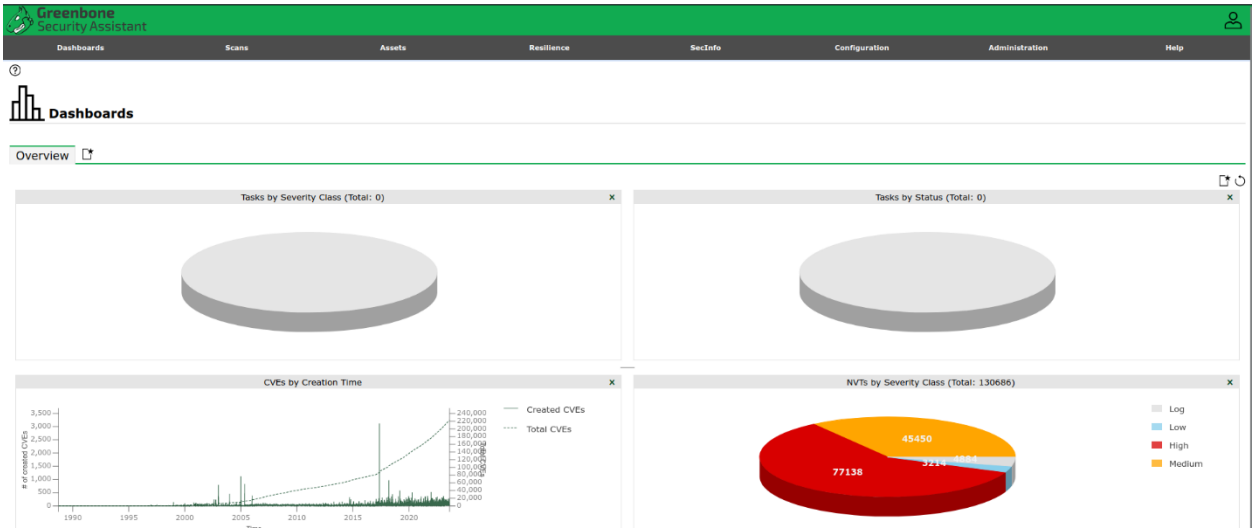
Sign In



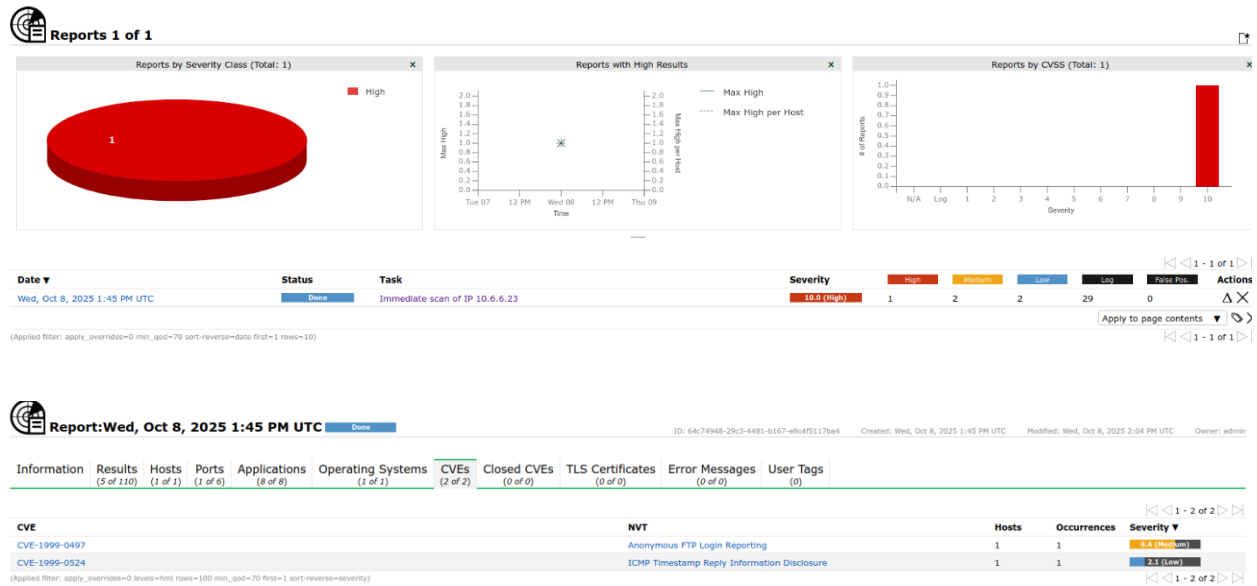
In the Greenbone Security Assistant login box, enter admin as the username and kali as the password. Username: admin Password: kali

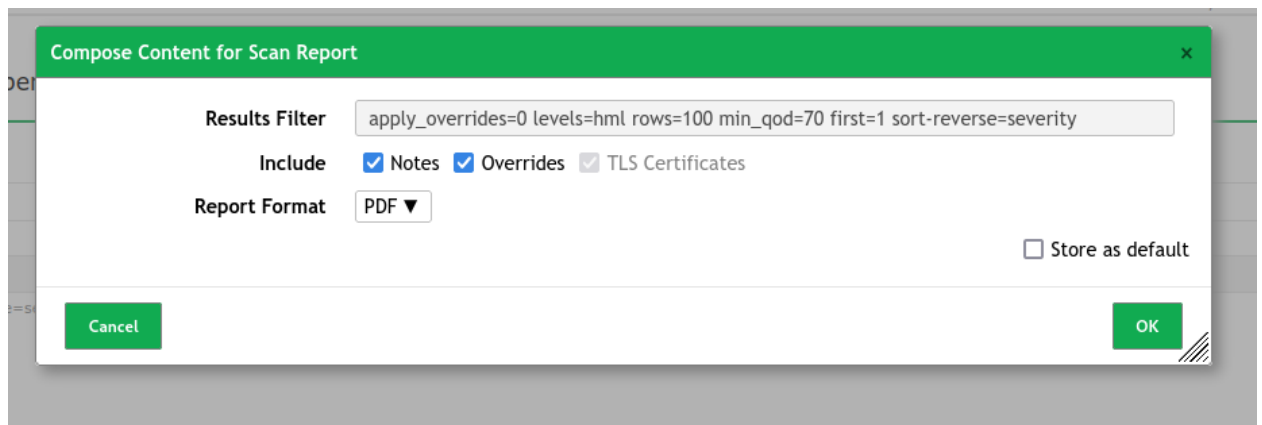
In the IP address or hostname box, enter the IP address 10.6.6.23 or gravemind.vm. Click the Start Scan button at the bottom of the screen. The scan will take a few minutes, so wait for it to complete. The status and percent complete are displayed on the screen. The scan will be finished when the status changes to Done





Download the report by clicking the Download Filtered Report button from the menu in the upper left of the report page. It has a downward-pointing arrow icon. In the settings box, choose to download the report in PDF format. After a brief delay, the PDF file should open in your browser





Scan Report

October 8, 2025

Summary

This document reports on the results of an automatic security scan. All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC". The task was "Immediate scan of IP 10.6.6.23". The scan started at Wed Oct 8 13:46:11 2025 UTC and ended at Wed Oct 8 14:04:46 2025 UTC. The report first summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.

Contents

1 Result Overview

2

When you are done with GVM services, use the following command to stop GVM.

```
(kali@kali)-[~]
└─$ sudo gvm-stop
[sudo] password for kali:
[>] Stopping GVM services
o gsad.service - Greenbone Security Assistant daemon (gsad)
   Loaded: loaded (/lib/systemd/system/gsad.service; disabled; preset: disabled)
   Active: inactive (dead)
   Docs: man:gsad(8)
         https://www.greenbone.net

Oct 08 13:32:21 Kali systemd[1]: Started gsad.service - Greenbone Security Assistant daemon (gsad).
Oct 08 13:40:30 Kali systemd[1]: Stopping gsad.service - Greenbone Security Assistant daemon (gsad)...
Oct 08 13:40:30 Kali systemd[1]: gsad.service: Deactivated successfully.
Oct 08 13:40:30 Kali systemd[1]: Stopped gsad.service - Greenbone Security Assistant daemon (gsad).
```

In your opinion, which tool is easier to use? Explain.

In my opinion, GVM is easier to use because it has a clear web interface, flexible configuration options, and does not require a commercial license.² It is recommended to keep the databases of vulnerabilities updated every few days. Research on the internet

It is recommended to keep the databases of vulnerabilities updated every few days. Research on the internet the necessary commands to update the GVM CVE database. Why do you think it is necessary to keep a database of all CVEs (current and past) for use by vulnerability scanners?

It is necessary to keep the full CVE database (past and current) because old vulnerabilities may still exist in systems. A complete database ensures accurate scans, proper detection, and historical analysis.