



Air University Islamabad, Main Campus.

National Cyber Security Academy (NCSA)

Department of Cyber Security

Course name:

Introduction to Cyber Security

Semester Project

Submitted by:

Khadija Zia

Ayesha Mushtaq

Hasan Ahmed Mumtaz

OpenVAS Project

Contents

OpenVAS.....	4
1. Explanation of the Selected Tool:	4
1.1. Key Features:	4
1.2. Purpose:.....	4
1.3. Limitations:	4
1.4. Advancements:	5
2. Implementation of Risk Assessment:.....	5
2.1. Purpose of the Task:	5
2.2. Requirements:	5
2.3. Tool and System Versions:	5
3. Setting Up the Environment:	6
3.1. Installation and Configuration of OpenVAS:	6
3.2. Metasploitable2(Target) Setup:	9
3.3. Network Configuration Verification:	11
3.4. Connectivity Verification (Reachability Test):.....	11
3.5. Target Configuration:	13
3.6. Assessment of Risk:	13
3.7. Vulnerability Scanning Process:	14
3.8. Results and Findings:	16
3.9. Report Generation:	16
4. Challenges Faced and Solutions:	17
4.1. Installation Issues:	17
4.2. Timeout Errors:	17
4.3. Configuration Complexity:	17
4.4. Resource Limitations:	17

4.5. Lack of Documentation:.....	17
4.6. PDF Export Challenge:	17
5. Contribution Table:.....	18
6. Conclusion:	18
7. References:.....	18

OpenVAS

OpenVAS (Open Vulnerability Assessment System) is an open-source tool for identifying security issues in networks. Part of the Greenbone Vulnerability Management (GVM) suite, it is widely used by cybersecurity professionals to assess vulnerabilities in systems and applications. It is a powerful tool for vulnerability scanning that identifies and evaluates potential weaknesses within a network.

1. Explanation of the Selected Tool:

1.1. Key Features:

OpenVAS offers a comprehensive vulnerability database that is regularly updated with information about known vulnerabilities, ensuring accurate and up-to-date assessments. It supports customizable scans, allowing users to target specific systems or services based on their needs. The tool generates detailed reports, highlighting detected vulnerabilities and their criticality, aiding in prioritization and remediation.

1.2. Purpose:

- i. Identifying exploitable vulnerabilities in systems.
- ii. Providing actionable insights to improve security posture.
- iii. Supporting compliance with security standards and policies.

OpenVAS is widely utilized due to its robust features, open-source nature, and capability to uncover a broad spectrum of vulnerabilities. Key reasons include:

1. **Risk Mitigation:** By identifying vulnerabilities, organizations can implement controls to reduce the risk of exploitation.
2. **Comprehensive Scans:** It provides in-depth scans that detect issues across multiple layers of the network.
3. **Ease of Reporting:** Detailed reports help organizations prioritize remediation efforts effectively.
4. **Cost-Effectiveness:** As an open-source tool, it offers advanced vulnerability scanning capabilities without licensing costs.

1.3. Limitations:

While OpenVAS is a powerful tool, it has limitations such as being resource-intensive, often requiring significant processing power and memory. Scans may fail or timeout on large networks.

due to misconfigurations or scale. Additionally, its initial setup can be complex for beginners, and successful scans depend on proper network configuration and access.

1.4. Advancements:

Recent advancements in OpenVAS have tackled several limitations, including an improved user interface for better usability and workflow management. The tool now features regular updates to its vulnerability database, ensuring coverage of the latest threats. Enhanced scalability optimizes performance on large networks and added API integration supports seamless connectivity with other security tools and frameworks.

2. Implementation of Risk Assessment:

Risk assessment was conducted using **OpenVAS on Kali Linux** to identify vulnerabilities in **Metasploitable2**, a purposely vulnerable virtual machine. Both systems were deployed in a controlled lab environment using VirtualBox to simulate a real-world network scenario.

2.1. Purpose of the Task:

The objectives of this task were:

- i. To scan the Metasploitable2 system for known vulnerabilities using OpenVAS.
- ii. To analyze detected vulnerabilities based on severity and risk level.
- iii. To generate a vulnerability assessment report for learning and evaluation purposes.

2.2. Requirements:

- i. Kali Linux (Attacker / Scanner Machine)
- ii. OpenVAS (GVM)
- iii. Metasploitable2 (Target Machine)
- iv. VirtualBox (Virtualization Platform)

2.3. Tool and System Versions:

The following tools and systems were used during the vulnerability assessment:

- **Kali Linux:** Kali Linux 2025.x (64-bit)
- **OpenVAS (GVM):** Greenbone Vulnerability Management (GVM) version 22.x
- **Metasploitable2:** Metasploitable2 Linux (Rapid7 vulnerable VM)
- **VirtualBox:** Oracle VM VirtualBox version 7.2.4x

These versions ensured compatibility between the scanning tool and the target system and provided a stable laboratory environment for vulnerability assessment.

3. Setting Up the Environment:

Two virtual machines were configured on VirtualBox:

- i. **Kali Linux VM** – Used to install and run OpenVAS.
- ii. **Metasploitable2 VM** – Used as the vulnerable target system.

Both VMs were connected using a **Host-Only Adapter network**, enabling direct communication between them while isolating them from external networks. This configuration ensured a controlled and secure laboratory environment for vulnerability assessment.

3.1. Installation and Configuration of OpenVAS:

OpenVAS was installed on the Kali Linux virtual machine following system updates

```
sudo apt update
```

```
(kali㉿kali)-[~]
└─$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.3 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [48.8 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [110 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [262 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [196 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [876 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [10.6 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [23.2 kB]
Fetched 70.6 MB in 1min 45s (675 kB/s)
1898 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Install OpenVAS using this command

```
sudo apt install openvas
```

```
(kali㉿kali)-[~]
└─$ sudo apt install openvas
Note, selecting 'gvm' instead of 'openvas'
The following packages were automatically installed and are no longer required:
  ibverbs-providers    libcephfs2   libgwdx0    libpython3.11-dev  python3-l
  libboost-iostreams1.83.0 libgfapi0    libglusterfs0  librados2      python3.1
  libboost-thread1.83.0   libgfRPC0   libibverbs1   librdmacm1t64   python3.1
Use 'sudo apt autoremove' to remove them.

Upgrading:
  blueman      libjs-sphinxdoc  libssl3t64    onboard      python3-arc4
  gvd          libldb2        liblalloc2   onboard-common python3-dev
  gvd-common   libpq5         libltdb1     onboard-data   python3-gpg
  libgpg-error0 libpython3-dev  libwbclient0 openssl     python3-gvm
  libgpgme11t64 libpython3-stdlib  libz3-4      openvas-scanner python3-ldb
  libgvm22t64   libsmbclient0  libz3-dev    python3      python3-minimal

Installing:
  gvm
```

Most probably you will see the message asking you to select **gvm** instead of **openvas**. Options will appear write Y to continue.

```
Continue? [Y/n] Y
Get:2 http://kali.download/kali kali-rolling/main
Get:1 http://mirrors.netix.net/kali kali-rolling/
Get:18 http://mirror.cspacehostings.com/kali kali
Get:24 http://mirror.telepoint.bg/kali kali-rolling/
Get:34 http://mirror.kku.ac.th/kali kali-rolling/
Get:29 http://http.kali.org/kali kali-rolling/main
Get:3 http://kali.download/kali kali-rolling/main
Get:62 http://mirror1.sox.rs/kali/kali kali-rolling/
Get:4 http://kali.download/kali kali-rolling/main
Get:66 http://http.kali.org/kali kali-rolling/main
Get:55 http://http.kali.org/kali kali-rolling/main
Get:5 http://kali.download/kali kali-rolling/main
Get:1 http://http.kali.org/kali kali-rolling/main
```

This will take some time. After this you will see an error message when you run gvm setup. Run setup using this command

```
sudo gvm-setup
```

This gvm command is used to run the setup.

```
sudo gvm-setup
```

```
(kali㉿kali)-[~]
└─$ sudo gvm-setup
Filesystem
[>] Starting PostgreSQL service
/usr/bin/gvm-setup: line 35: [: too many arguments

[>] Creating GVM's certificate files

[>] Creating PostgreSQL database

[*] Creating database user

[*] Creating database

[*] Creating permissions
CREATE ROLE
```

If everything went well, you would get this message. This process will take a lot of time. When all the desired files and configurations are installed, now it's time to check the setup of gvm. To do this, run this command:

```
sudo gvm-check-setup
```

```
(kali㉿kali)-[~]
└─$ sudo gvm-check-setup
[sudo] password for kali:
gvm-check-setup 23.11.0
  Test completeness and readiness of GVM-23.11.0
Step 1: Checking OpenVAS (Scanner) ...
  OK: OpenVAS Scanner is present in version 23.13
  OK: Notus Scanner is present in version 22.6.3
  OK: Server CA Certificate is present as /var/lib/openvas/ca/certs/ca.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
  OK: the mqtt_server_uri is defined in /etc/openvas
  OK: _gvm owns all files in /var/lib/openvas/plugins/*
  OK: NVT collection in /var/lib/openvas/plugins/
  OK: The notus directory /var/lib/notus/products/
Checking that the obsolete redis database has been removed
```

If everything went well, this message would display,

```
SUGGEST: Edit the /etc/gvm/pwpolicy.conf file
Step 9: Checking greenbone-security-assistant ...
  OK: greenbone-security-assistant is installed

It seems like your GVM-23.11.0 installation is OK.
```

When you run the check setup command, check the portion where you get the local host address which is **127.0.0.1** and your password. Username is always **admin**. Copy the address and paste it on your browser, and this window will appear,

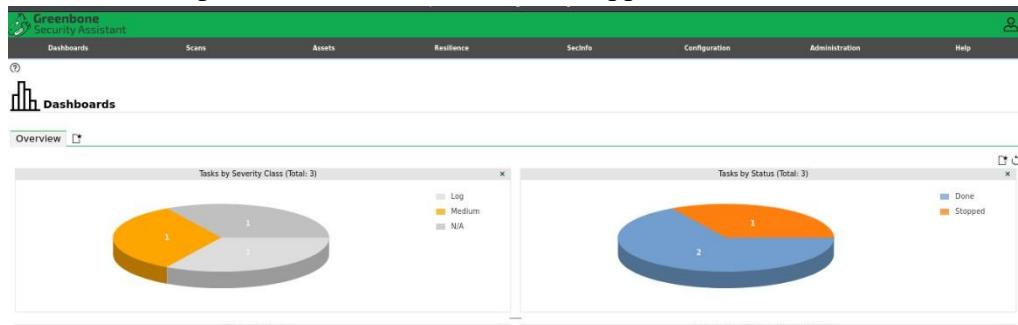


Sign in to your account

admin	<hr/>
<hr/>	
Sign In	



Enter the username and password, and Dashboard will appear.



Before starting your task, go to the configuration tab and check whether configurations are present or not. If yes then no worries, but if not then enter these three commands one by one

```
greenbone-feed-sync --type GVMD_DATA
```

```
greenbone-feed-sync --type SCAP
```

```
greenbone-feed-sync --type CERT
```

After a long wait, you will see configurations in configurations portion.

Summary of OpenVAS Setup on Kali Linux:

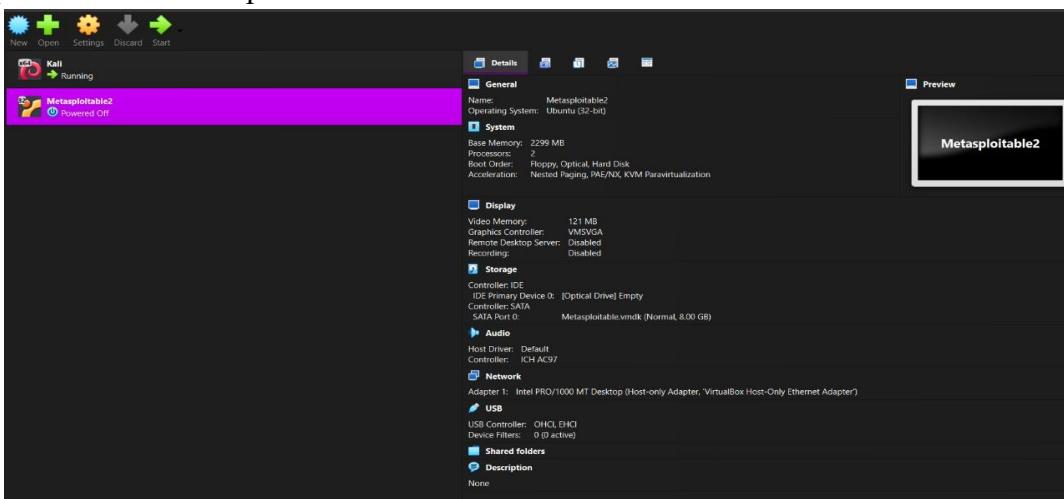
OpenVAS was installed on the Kali Linux machine by updating the system and installing the required packages. After installation, the GVM setup process was completed, and all necessary services were verified using the built-in setup check command. The OpenVAS web interface was accessed through the local host address provided during setup.

3.2. Metasploitable2(Target) Setup:

Metasploitable2 was installed using Rapid7.

Name	Type	Compressed size	Password p...	Size	Ratio	Date modified
Metasploitable	Virtual Machine Disk Form...	844,806 KB	No	1,880,512 KB	56%	20/05/2012 3:01 pm
Metasploitable.nvram	NVRAM File	2 KB	No	9 KB	79%	20/05/2012 2:56 pm
Metasploitable.vmsd	VMSD File	0 KB	No	0 KB	0%	07/05/2010 2:46 pm
Metasploitable.vmx	VMX File	2 KB	No	3 KB	62%	20/05/2012 3:00 pm
Metasploitable.vmxsf	VMXF File	1 KB	No	1 KB	32%	07/05/2010 2:46 pm

Metasploitable2 was set up on Virtual Box.



Metasploitable2 was logged in after entering the “msfadmin”, the admin ID.

```
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Thu Dec 11 06:00:25 EST 2025 on ttym1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$
```

Now checking for the IP of Metasploitable2 using “ifconfig” command.

ifconfig

```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 08:00:27:99:2e:8c
          inet addr:192.168.56.101 Bcast:192.168.56.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe99:2e8c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2564 (2.5 KB) TX bytes:3924 (3.8 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:96 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21437 (20.9 KB) TX bytes:21437 (20.9 KB)

msfadmin@metasploitable:~$
```

The IP address of the Metasploitable2 machine was identified (192.168.56.101) using network configuration command.

This IP address was then added manually as a target in OpenVAS under the **Targets** configuration section.

3.3. Network Configuration Verification:

Before performing connectivity tests and scans, it was verified that both VMs were on the **same host-only network**:

- VirtualBox network settings were checked to confirm both machines used the **same Host-Only Adapter**.
- IP addresses were verified using ifconfig (Metasploitable2) and ip a (Kali Linux) to ensure both systems were on the **same subnet**.

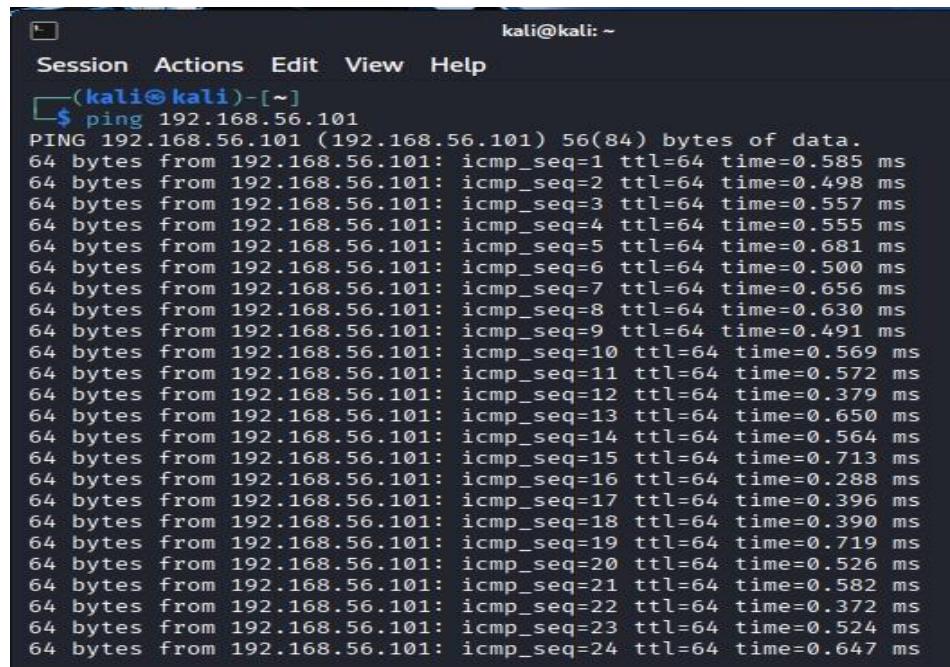
This configuration allowed proper communication between Kali Linux and Metasploitable2, making them ready for vulnerability scanning using OpenVAS

3.4. Connectivity Verification (Reachability Test):

Ping Test:

The ping command was used from the Kali Linux machine to verify basic network connectivity with the Metasploitable2 target system.

```
ping 192.168.56.101
```



A screenshot of a terminal window titled "kali@kali: ~". The window shows the command \$ ping 192.168.56.101 followed by its output. The output displays 24 ICMP echo requests sent to the target IP, with each request showing the sequence number, TTL, time taken, and the source IP (192.168.56.101). The times range from 0.498 ms to 0.681 ms.

```
(kali㉿kali)-[~]
$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.585 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.498 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.557 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=0.555 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=0.681 ms
64 bytes from 192.168.56.101: icmp_seq=6 ttl=64 time=0.500 ms
64 bytes from 192.168.56.101: icmp_seq=7 ttl=64 time=0.656 ms
64 bytes from 192.168.56.101: icmp_seq=8 ttl=64 time=0.630 ms
64 bytes from 192.168.56.101: icmp_seq=9 ttl=64 time=0.491 ms
64 bytes from 192.168.56.101: icmp_seq=10 ttl=64 time=0.569 ms
64 bytes from 192.168.56.101: icmp_seq=11 ttl=64 time=0.572 ms
64 bytes from 192.168.56.101: icmp_seq=12 ttl=64 time=0.379 ms
64 bytes from 192.168.56.101: icmp_seq=13 ttl=64 time=0.650 ms
64 bytes from 192.168.56.101: icmp_seq=14 ttl=64 time=0.564 ms
64 bytes from 192.168.56.101: icmp_seq=15 ttl=64 time=0.713 ms
64 bytes from 192.168.56.101: icmp_seq=16 ttl=64 time=0.288 ms
64 bytes from 192.168.56.101: icmp_seq=17 ttl=64 time=0.396 ms
64 bytes from 192.168.56.101: icmp_seq=18 ttl=64 time=0.390 ms
64 bytes from 192.168.56.101: icmp_seq=19 ttl=64 time=0.719 ms
64 bytes from 192.168.56.101: icmp_seq=20 ttl=64 time=0.526 ms
64 bytes from 192.168.56.101: icmp_seq=21 ttl=64 time=0.582 ms
64 bytes from 192.168.56.101: icmp_seq=22 ttl=64 time=0.372 ms
64 bytes from 192.168.56.101: icmp_seq=23 ttl=64 time=0.524 ms
64 bytes from 192.168.56.101: icmp_seq=24 ttl=64 time=0.647 ms
```

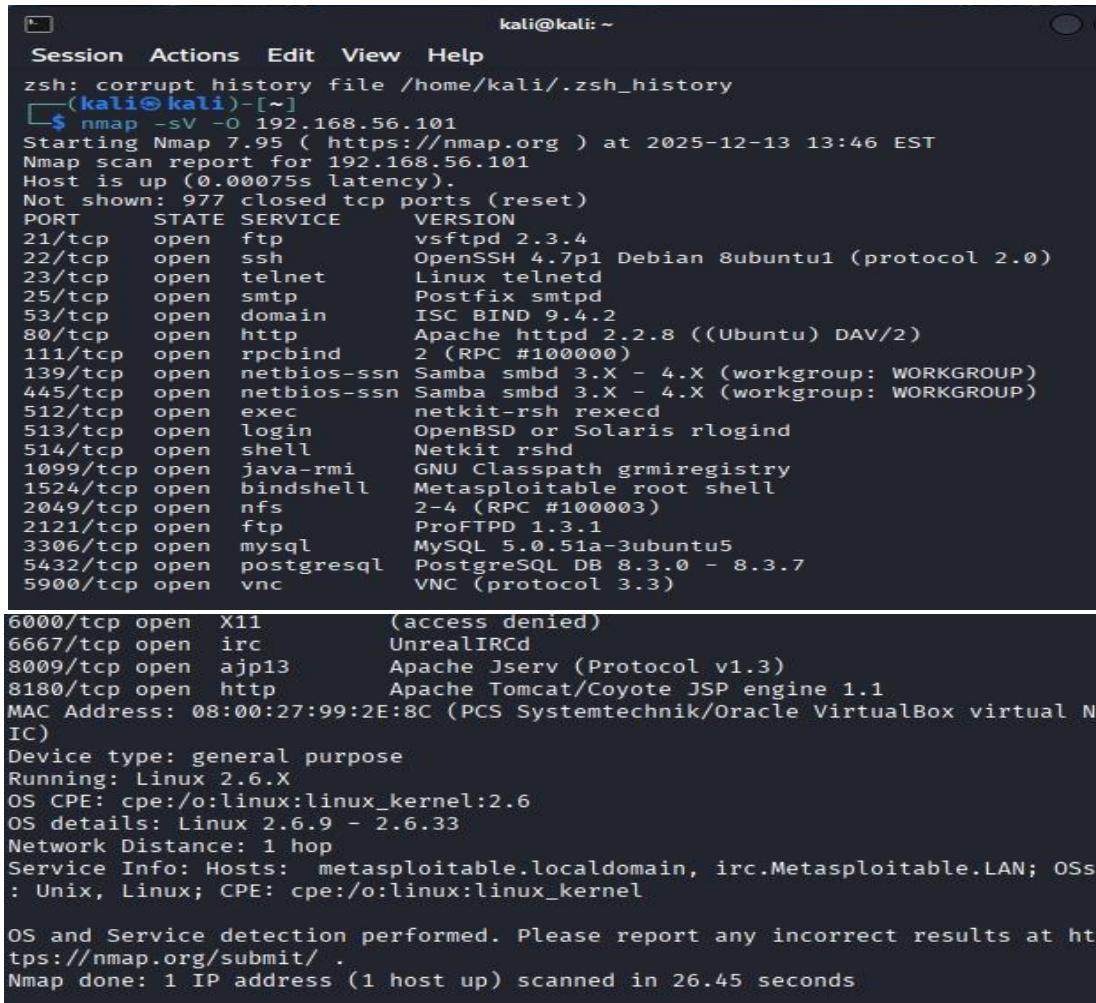
Result:

The Metasploitable2 machine responded successfully with ICMP replies and acceptable response times. This confirmed that the target system was online and reachable from the Kali Linux machine.

Nmap Scan:

The nmap tool was used to identify open ports and active services running on the Metasploitable2 system.

```
nmap -sV -O 192.168.56.101
```



```
kali@kali: ~
Session Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali㉿kali)-[~]
└─$ nmap -sV -O 192.168.56.101
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-13 13:46 EST
Nmap scan report for 192.168.56.101
Host is up (0.00075s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
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)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)

6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:99:2E:8C (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.45 seconds
```

Result:

The scan revealed multiple open ports and running services, including vulnerable services such as FTP, Telnet, HTTP, and SMB. The presence of these open ports confirmed that the Metasploitable2 system was accessible and suitable for vulnerability assessment using OpenVAS.

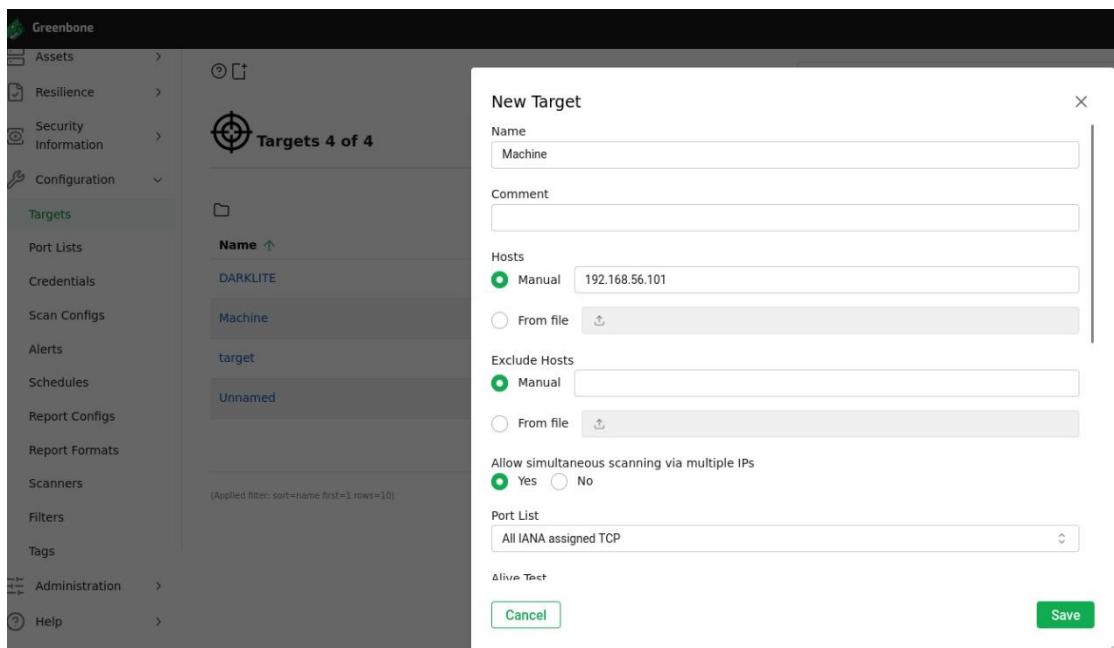
3.5. Target Configuration:

To configure the target, the **Configuration** → **Targets** option was selected, and a new target was created.

The identified IP address (192.168.56.101) of the Metasploitable2 machine was entered in the target field and the target was named as “Machine”.

Default port ranges and scan settings were used to ensure a comprehensive vulnerability assessment.

Once saved, the target became available for selection during scan task creation. This configuration enabled OpenVAS to correctly identify and assess vulnerabilities present on the Metasploitable2 system.



3.6. Assessment of Risk:

The objective of the assessment was to identify vulnerabilities in Metasploitable2 and evaluate associated risks. The assessment included:

- Scanning all open services on Metasploitable2 using OpenVAS.
- Detecting outdated, misconfigured, or vulnerable services (FTP, Telnet, Samba, HTTP).
- Categorizing vulnerabilities based on severity (Critical, High, Medium, Low).
- Generating actionable insights for risk mitigation and security improvements.

This step established the **risk context** and justified the need for a vulnerability assessment.

3.7. Vulnerability Scanning Process:

Starting the vulnerability process after running both machines on Virtual Box.
On Linux, using the command to start OpenVAS.

```
sudo gvm-start
```

```
kali@kali: ~
Session Actions Edit View Help
└──(kali㉿kali)-[~]
$ sudo gvm-start
[sudo] password for kali:
[>] Please wait for the GVM services to start.
[>]
[>] You might need to refresh your browser once it opens.
[>]
[>] Web UI (Greenbone Security Assistant): https://127.0.0.1:9392

● gsad.service - Greenbone Security Assistant daemon (gsad)
   Loaded: loaded (/usr/lib/systemd/system/gsad.service; disabled; preset: disabled)
     Active: active (running) since Sat 2025-12-13 14:02:52 EST; 24ms ago
   Invocation:
     Docs:

      Main PID: [REDACTED]

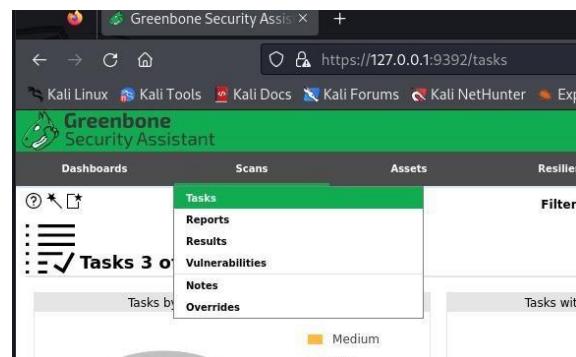
Dec 13 14:02:18 kali systemd[1]: Starting ospd-openvas.service - OSPD Wrapper
for the OpenVAS Scanner (ospd-openvas)...
Dec 13 14:02:22 kali systemd[1]: Started ospd-openvas.service - OSPD Wrapper
for the OpenVAS Scanner (ospd-openvas).

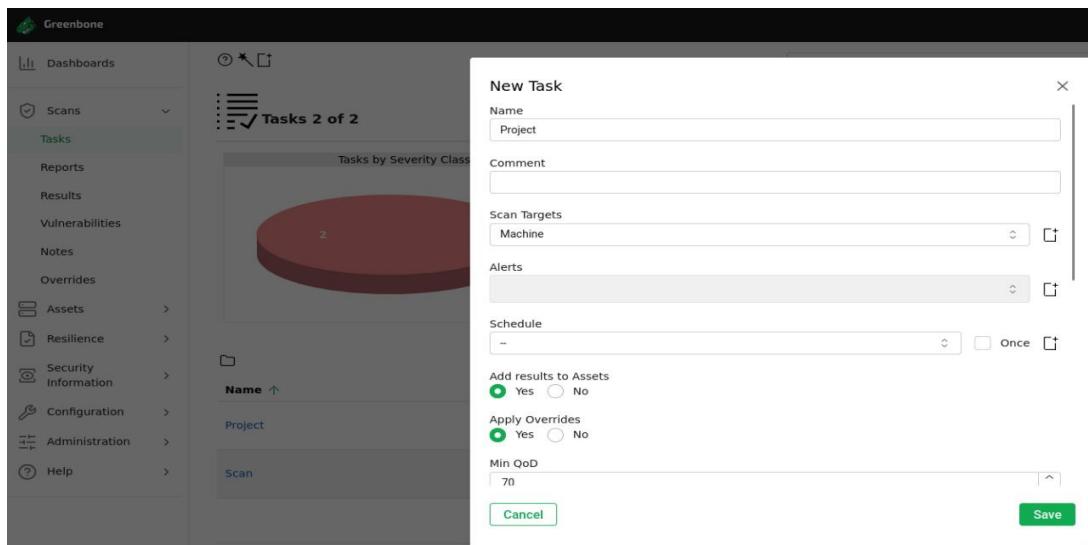
[>] Opening Web UI (https://127.0.0.1:9392) in: 5 ... 4 ... 3 ... 2 ... 1 ...
```

The command resulted in opening of Webpage of Greenbone Vulnerability Assessment.

Access OpenVAS web interface.

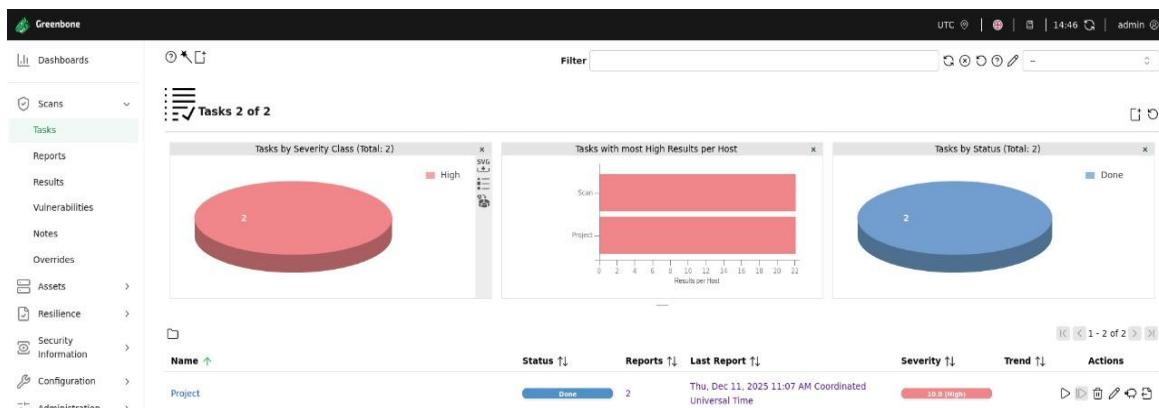
Create a new scan task using Wizard





In Scans, we had named the scanning procedure as “**Project**” and put the target the Metasploitable2(Machine).

Then save the Project and start the scanning.



The scan ran successfully and gave the required output.

Summary:

The vulnerability assessment was performed using the following steps:

- The OpenVAS web interface was accessed from the Kali Linux browser.
- A new scan task was created using the Wizard option.
- The IP address of the Metasploitable2 machine was entered as the scan target.
- The scan was initiated, and OpenVAS performed a comprehensive vulnerability assessment.
- The scan status was monitored until completion.

3.8. Results and Findings:

After completion of the scan, the results revealed multiple vulnerabilities in the Metasploitable2 system.

The screenshot displays the Greenbone Security Audit interface. On the left, a sidebar menu includes 'Dashboards', 'Scans', 'Tasks', 'Reports' (selected), 'Results', 'Vulnerabilities', 'Notes', 'Overrides', 'Assets', 'Resilience', 'Security Information', 'Configuration', 'Administration', and 'Help'. The main area shows a scan report for 'Coordinated Universal Time' on 'Thu, Dec 11, 2025 11:07 AM'. The report summary indicates 68 results found, with details for Hosts (1 of 1), Ports (19 of 23), Applications (19 of 19), Operating Systems (1 of 1), CVEs (34 of 34), Closed CVEs (0 of 0), TLS Certificates (2 of 2), Error Messages (1 of 1), and User Tags (0). Below the summary, a table lists vulnerabilities categorized by severity (e.g., Critical, High, Medium, Low) and other metrics like QoD and EPSS Score. Two specific findings are highlighted: 'The rexec service is running' and 'rlogin Passwordless Login'.

Vulnerability	Severity	QoD	Host IP	Name	Location	EPSS Score	Percentile	Created
The rexec service is running	10.0 (High)	80 %	192.168.56.101	512/tcp	N/A	N/A		Thu, Dec 11, 2025 11:30 AM Coordinated Universal Time
rlogin Passwordless Login	10.0 (High)	80 %	192.168.56.101	513/tcp	N/A	N/A		Thu, Dec 11, 2025 11:27 AM Coordinated Universal Time

These included:

- Outdated and insecure services such as FTP, Telnet, and Samba.
- Multiple open ports exposing vulnerable services.
- Critical and high-severity vulnerabilities associated with known CVEs.

The results demonstrated that Metasploitable2 is highly vulnerable and unsuitable for use in a production environment.

3.9. Report Generation:

OpenVAS generated a detailed vulnerability report categorizing issues by severity. The report was exported in **PDF format** for analysis and documentation. This report provided valuable insights into the vulnerabilities present and their potential impact.

4. Challenges Faced and Solutions:

During the installation and use of OpenVAS on Kali Linux, several challenges were encountered:

4.1. Installation Issues:

- **Problem:** Initial attempts to install OpenVAS failed due to missing dependencies and misconfigured repositories.
- **Solution:** Manually added and updated repositories, followed by troubleshooting missing packages.

4.2. Timeout Errors:

- **Problem:** After successfully running one scan, subsequent scans resulted in timeout errors.
- **Solution:** Reinstalled OpenVAS and manually adjusted the network configuration to ensure proper connectivity.

4.3. Configuration Complexity:

- **Problem:** The tool required extensive configuration, including setting up services and defining proper network parameters.
- **Solution:** Consulted documentation and forums to manually configure the network and services.

4.4. Resource Limitations:

- **Problem:** Scans were resource-intensive, slowing down the system and causing delays.
- **Solution:** Optimized scan settings by reducing the number of simultaneous checks and disabling non-critical features.

4.5. Lack of Documentation:

- **Problem:** The lack of detailed, step-by-step documentation for resolving specific issues added to the complexity.
- **Solution:** Relied on community forums and trial-and-error methods to resolve issues.

4.6. PDF Export Challenge:

- **Problem:** Exporting the vulnerability assessment report as a PDF from the virtualized Kali Linux environment to Windows was initially difficult. The PDF was generated in Kali but not automatically accessible in Windows.
- **Solution:** The report was copied from the Kali Desktop to a VirtualBox shared folder (/media/sf_linux), making it accessible on the Windows host system. This ensured the report could be submitted and reviewed without issues.

5. Contribution Table:

S#	Names	Contribution
1	Khadija Zia	Tool Research, setup, scanning & report
2	Ayesha Mushtaq	Tool Research, OpenVAS theory & scanning
3	Hasan Ahmed Mumtaz	Network setup & scanning support

6. Conclusion:

This project successfully demonstrated the use of OpenVAS on Kali Linux to detect vulnerabilities in the Metasploitable2 system. The assessment highlighted numerous critical security weaknesses, emphasizing the importance of vulnerability scanning in identifying and mitigating security risks. OpenVAS proved to be an effective and reliable tool for vulnerability assessment within a controlled laboratory environment.

7. References:

- [1] Greenbone Networks, *Greenbone Vulnerability Management Documentation*. [Online]. Available: <https://greenbone.github.io/docs/>
 - [2] Kali Linux, *Official Kali Linux Documentation*. [Online]. Available: <https://www.kali.org/docs/>
 - [3] Rapid7, *Metasploitable2 Documentation*. [Online]. Available: <https://docs.rapid7.com/metasploit/metasploitable-2/>
 - [4] Oracle Corporation, *Oracle VM VirtualBox User Manual*. [Online]. Available: <https://www.virtualbox.org/manual/>
-