WEB SERVER HACKING

Module-13

ATHARVA KATKAR

Table of Contents

- 2.2.5 Finding Open Ports using Zenmap
- 2.2.6 Finding Service Versions using Nmap
- 2.2.7 Finding Service Versions using Zenmap

2.3 Vulnerability Analysis

- 2.3.1 Definition of Vulnerability Analysis
- 2.3.2 Purpose of Vulnerability Analysis
- 2.3.3 Finding Vulnerabilities using Nmap Scripts
- 2.3.4 Finding Vulnerabilities using Nikto
- 2.3.5 Finding Vulnerabilities using Acunetix

2.4 Gaining Access

- 2.4.1 Password Cracking using Hydra
- 2.4.2 Anonymous Login using FTP Port
- 2.4.3 Gaining Access using Rlogin (Port 514)

2.5 Exploitation

- 2.5.1 Exploitation using Metasploit

Extra Activity

Windows Server 2019

- 3.1 Definition
- 3.2 Why Windows Server 2019 is Used

3.3 Footprinting

- 3.3.1 Footprinting using Ping
- 3.3.2 Footprinting using Nmap

3.4 Vulnerability Scanning

---

5. Defense Section

- 5.1 How to Defend Against Web Server Attacks

WEB SERVER HACKING

Most people think a web server is just hardware, but a web server also includes software applications. In general, a client initiates the communication process through HTTP requests. When a client wants to access any resource such as web pages, photos, or videos, then the client's browser generates an HTTP request to the web server. Depending on the request, the web server collects the requested information or content from data storage or the application servers and responds to the client's request with an appropriate HTTP response. If a web server cannot find the requested information, then it generates an error message. Ethical hackers or pen testers use numerous tools and techniques to hack a target web server.

Types of Web Servers

1. **Apache HTTP Server** – Most widely used open-source server.
2. **Nginx** – High-performance, lightweight, and popular for reverse proxy/load balancing.
3. **Microsoft IIS** – Windows-based server from Microsoft.
4. **LiteSpeed** – Commercial web server known for speed and performance.
5. **Tomcat** – Used for Java-based applications (servlets, JSP).
6. **Node.js** – JavaScript runtime often used as a lightweight web server.

Web Server Components

1. **Hardware**: The physical machine storing website files (HTML, CSS, JS).

2. **Software**: Web server software like Apache, Nginx, etc., running on OS (Linux/Windows).
3. **HTTP/HTTPS Protocol**: Used for communication between browser and server.
4. **Web Content**: Static files (HTML, CSS) and dynamic content (PHP, Python, etc.).

🔹 How a Web Server Works

1. User enters URL in browser.
2. DNS resolves domain name to IP address.
3. Browser sends an **HTTP request** to that IP address.
4. Web server receives the request.
5. Web server locates the requested file or processes it via backend code.
6. Sends back an **HTTP response** with content (HTML, images, data).
7. Browser displays the content to the user.

🔹 Features of a Web Server

- Supports **HTTP/HTTPS protocols**
- Can handle **static and dynamic content**
- Provides **authentication & access control**
- **Logging and monitoring**
- **Load balancing**
- **Virtual hosting** (hosting multiple websites on a single server)
- **Compression (gzip)** to optimize bandwidth
- **SSL/TLS support** for secure communication

🔹 Common Directories in Web Server

- /var/www/html/ – Default web root in Linux (Apache)
- htdocs/ – Default in XAMPP
- wwwroot/ – Default in IIS

🔹 Web Server vs Application Server

Feature	Web Server	Application Server
Content	Static (HTML, CSS)	Dynamic (JSP, PHP, Python)
Protocol	HTTP/HTTPS	HTTP, TCP/IP
Example	Apache, Nginx	Tomcat, JBoss

🔹 Security for Web Servers

1. **Use HTTPS with SSL/TLS**
2. **Disable directory listing**
3. **Enable firewall rules**
4. **Limit server exposure**
5. **Use Web Application Firewall (WAF)**
6. **Regular patching and updates**
7. **Log access and monitor for intrusion**
8. **Use secure headers** (CSP, XSS protection)

🔹 Web Server Attacks

- **DoS/DDoS attacks**
- **Directory traversal**
- **Misconfiguration**

- **Information disclosure**
- **File inclusion**
- **Remote Code Execution (RCE)**
- **Cross-site scripting (XSS)**
- **SQL injection (if server runs dynamic scripts)**

🔹 Web Server Logs

- **Access Logs** – Info about client requests.
- **Error Logs** – Info about issues/errors.
- Used for **monitoring, troubleshooting, and incident response**.

🔹 Commands to Start/Stop Common Web Servers

Apache (Linux):

bash

CopyEdit

sudo systemctl start apache2

sudo systemctl stop apache2

sudo systemctl restart apache2

Nginx (Linux):

bash

CopyEdit

sudo systemctl start nginx

sudo systemctl stop nginx

sudo systemctl restart nginx

🔹 Tools to Test Web Servers

- **Nikto** – Vulnerability scanner
- **Nmap** – Port & service scanning
- **Burp Suite** – Manual penetration testing
- **OWASP ZAP** – Automated vulnerability scanning
- **curl / wget** – Testing HTTP responses

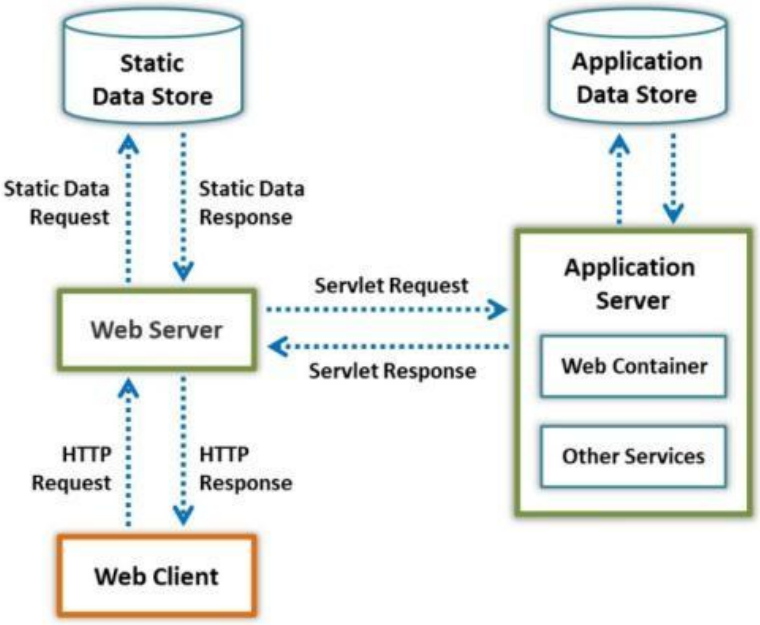
🔹 Popular Web Server File Types

- .html, .css, .js, .jpg, .png – Static files
- .php, .jsp, .asp, .py – Dynamic scripts

🔹 Web Server Optimization Tips

- Enable **caching** (browser/server-side)
- Use **CDN** for static content
- Enable **gzip compression**
- Configure **load balancing**
- Minimize resource usage via **tuning** (threads, buffer size)

Typical client-server communication in web server operation



t Web Server Security Issues -:

1. Unpatched Server Software

- Issue: Running outdated versions of Apache, Nginx, IIS, etc.
- Risk: Vulnerable to known CVEs (exploits).
- Solution: Regular updates and patch management.

---

2. Misconfigurations

- Issue: Incorrect permissions, enabled directory listing, open ports, exposed config files.
- Risk: Information disclosure or full server compromise.
- Solution: Use tools like Lynis, Nikto, and secure server configurations.

---

3. Default Settings

- Issue: Using default credentials (e.g., admin:admin), open test pages (like /test.php).
- Risk: Attackers exploit default access.

- Solution: Change all default settings and remove unused apps/scripts.

---

4. Directory Traversal

- Issue: Improper input validation allows access to server directories (e.g., ../../../etc/passwd).
- Risk: Leaks sensitive system files.
- Solution: Input sanitization and disabling unnecessary directory access.

---

5. Information Disclosure

- Issue: Error messages, banner grabbing (e.g., Apache 2.4.6), directory listings.
- Risk: Reveals server type, version, OS, paths.
- Solution: Hide server banners (ServerTokens Prod), suppress error messages.

---

6. Denial of Service (DoS/DDoS)

- Issue: Flooding server with requests to exhaust resources.
- Risk: Server becomes unresponsive.
- Solution: Use rate limiting, WAF, cloud-based protection (e.g., Cloudflare).

---

7. Insecure HTTP Methods Enabled

- Issue: Methods like PUT, DELETE, TRACE, OPTIONS allowed.
- Risk: Attackers may upload malicious files or conduct cross-site tracing.

- Solution: Disable all non-required HTTP methods.

---

8. SSL/TLS Weaknesses

- Issue: Using outdated protocols like SSLv2, weak ciphers (e.g., RC4).
- Risk: Susceptible to MITM attacks.
- Solution: Use strong ciphers and TLS 1.2/1.3 only.

---

9. Open Ports/Services

- Issue: Web server exposes unnecessary services (e.g., FTP, Telnet, SMTP).
- Risk: Increases attack surface.
- Solution: Close all unused ports, verify with Nmap.

---

10. Weak File Permissions

- Issue: World-writable directories or scripts.
- Risk: Attackers can upload or alter files.
- Solution: Use proper file ownership and least privilege permissions.

---

11. No Input Validation

- Issue: Inputs directly used in server-side logic (like PHP).
- Risk: Leads to XSS, SQLi, LFI/RFI.
- Solution: Input sanitization, parameterized queries.

---

12. Remote File Inclusion (RFI) / Local File Inclusion (LFI)

- Issue: Scripts load remote/local files without checks.
- Risk: Execute remote malicious code.

- Solution: Validate and sanitize all file inputs.

---

13. Insecure Admin Interfaces

- Issue: Admin panels accessible publicly.
- Risk: Brute force, credential stuffing.
- Solution: Restrict access by IP, use 2FA, hide admin endpoints.

---

14. Lack of Monitoring and Logging

- Issue: No real-time alerting or logs.
- Risk: Attacks go undetected.
- Solution: Enable access/error logs and use SIEM tools.

---

15. Cross-Site Scripting (XSS)

- Issue: Web app hosted on server reflects unescaped user input.
- Risk: Code injection, session hijack.
- Solution: Sanitize all output and use security headers (CSP, X-XSS-Protection).

---

16. SQL Injection

- Issue: Server-side code takes unsanitized input into SQL queries.
- Risk: Database access, data leakage.
- Solution: Use parameterized queries, WAF, and validations.

---

17. Improper Access Control

- Issue: Users can access restricted areas (e.g., /admin) without authentication.
- Risk: Privilege escalation.

- Solution: Implement Role-Based Access Control (RBAC) and session validation.

---

18. Malware in Uploaded Files

- Issue: Users upload PHP shells, JavaScript malware, etc.
- Risk: Server takeover, XSS.
- Solution: Whitelist file types, scan uploads with antivirus, store files outside web root.

---

19. Insufficient Logging & Monitoring

- Issue: No alerts for anomalies.
- Risk: Long undetected breaches.
- Solution: Integrate log monitoring tools (like ELK, Graylog).

---

20. Vulnerable Third-party Modules or Plugins

- Issue: Web servers often use add-ons (e.g., WordPress plugins).
- Risk: Vulnerabilities in plugins may affect the entire server.
- Solution: Keep all third-party modules updated.

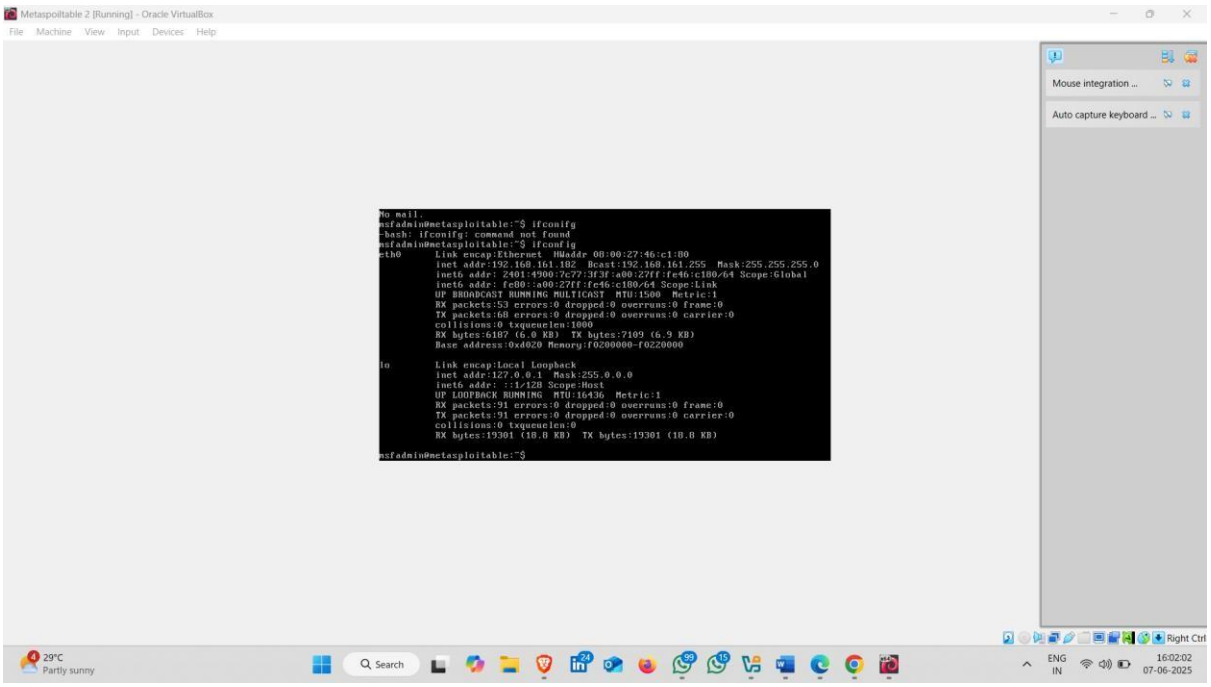
Matesploitable 2

Metasploitable 2 is a **deliberately vulnerable Linux virtual machine** created by Rapid7 for testing security tools and practicing penetration testing skills using Metasploit and other ethical hacking tools.

- Based on **Ubuntu 8.04 Server**
- Contains multiple **intentionally vulnerable services**
- Used in **labs, training, CTFs, and red team practice**

Attacker machine -: Kali linux . (ip address-:192.168.161.192)

Victim Machine -: Metasploitable 2 .(ip address- 192.168.161.182)



Reconnaissance/Footprinting

Reconnaissance (also known as **Footprinting**) is the **first phase** of ethical hacking or penetration testing, where the attacker gathers **information about the target system or network** before launching an attack.

🎯 Goal:

To **collect as much data as possible** to find potential **attack vectors** and **vulnerabilities** without alerting the target.

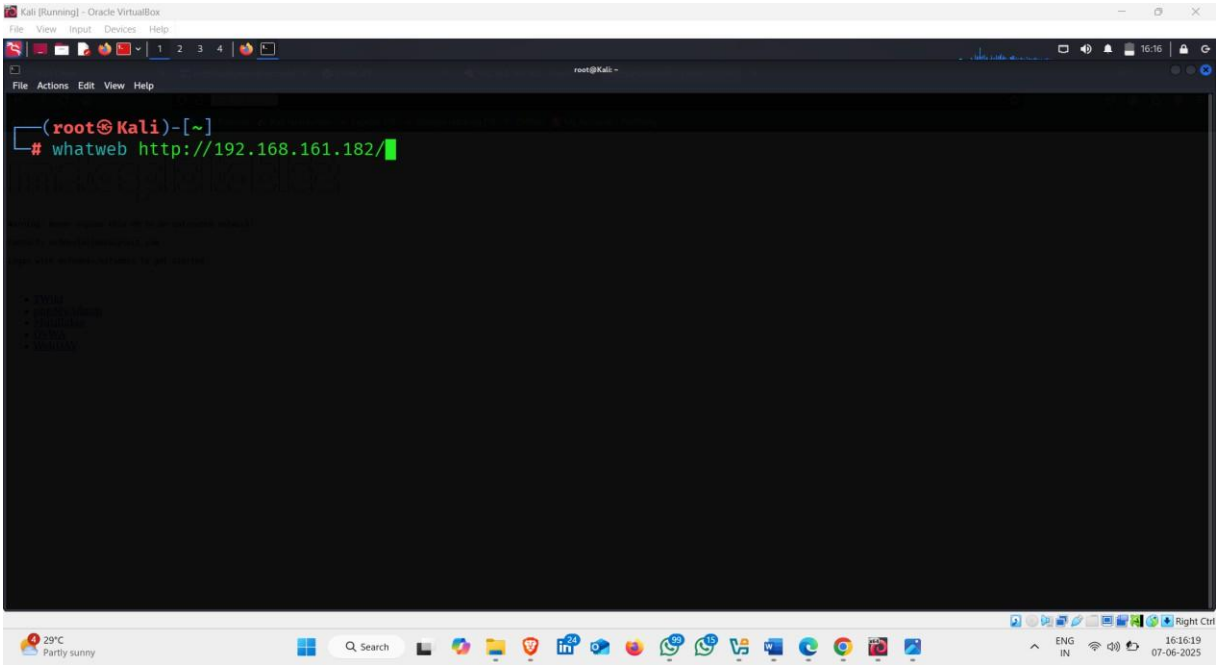
1.Perform Footprinting using whatweb tool

WhatWeb is an open-source **web scanner and fingerprinting tool** used to identify **technologies** running on a website.

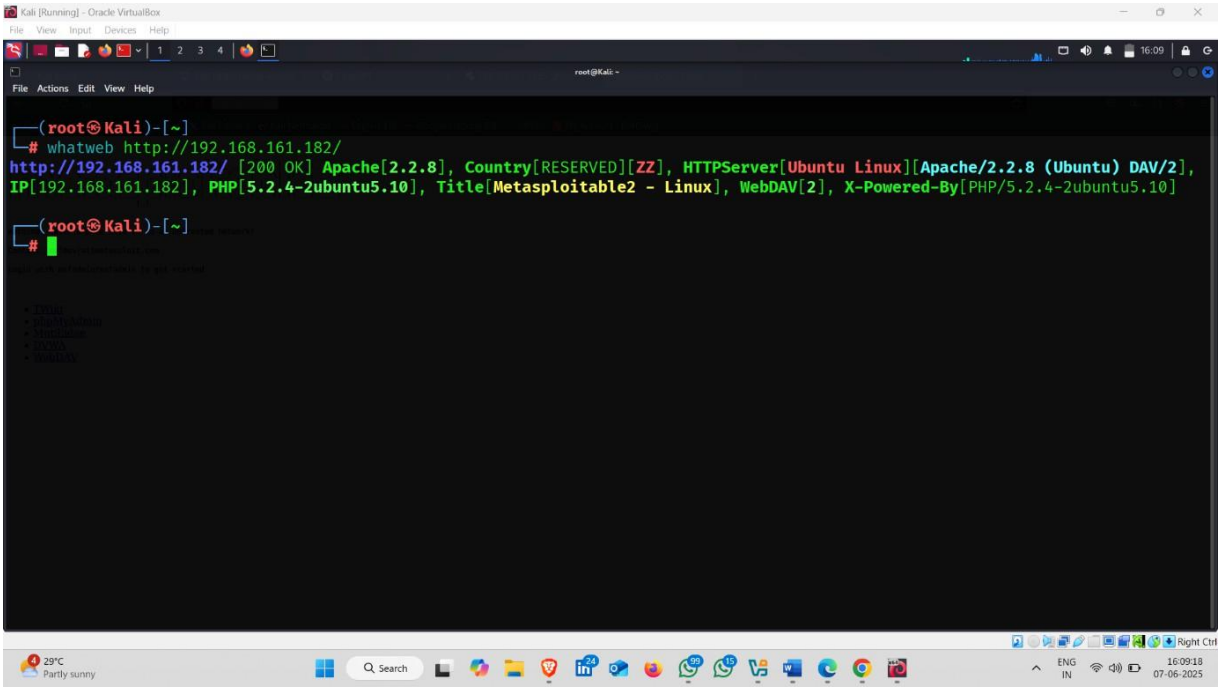
Key Purpose:

- Detect **web server software, CMS (like WordPress, Joomla), frameworks, programming languages, analytics tools, security mechanisms**, etc.

Command -: whatweb http://192.168.161.182/



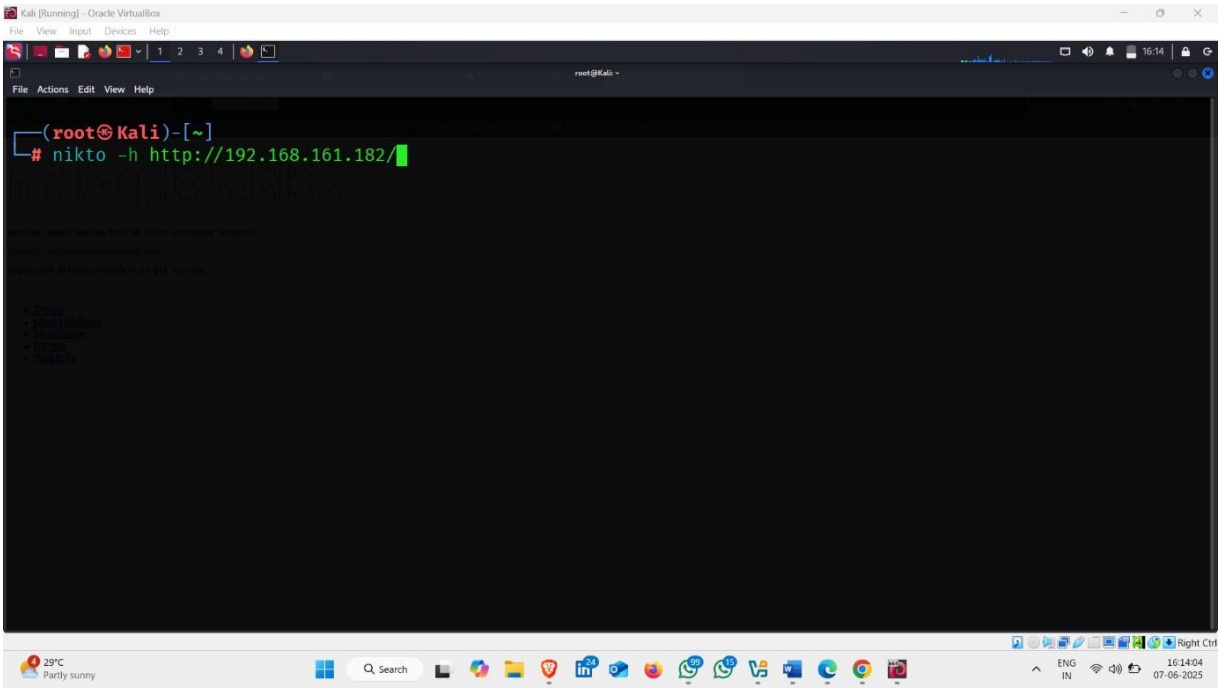
- Result



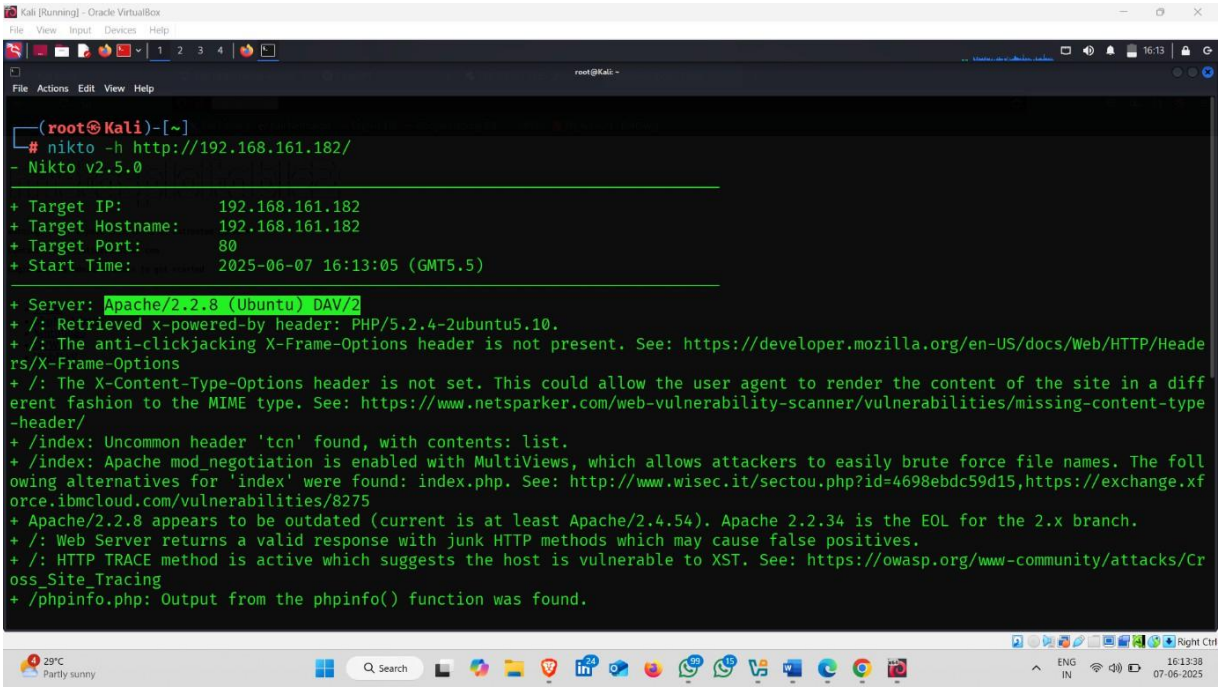
2.Perform Footprinting using Nikto

Nikto is a **web server scanner** used in the **footprinting (reconnaissance)** phase to gather detailed information about a target's web server.

Command -: nikto -h http://192.168.161.182



- Result -:



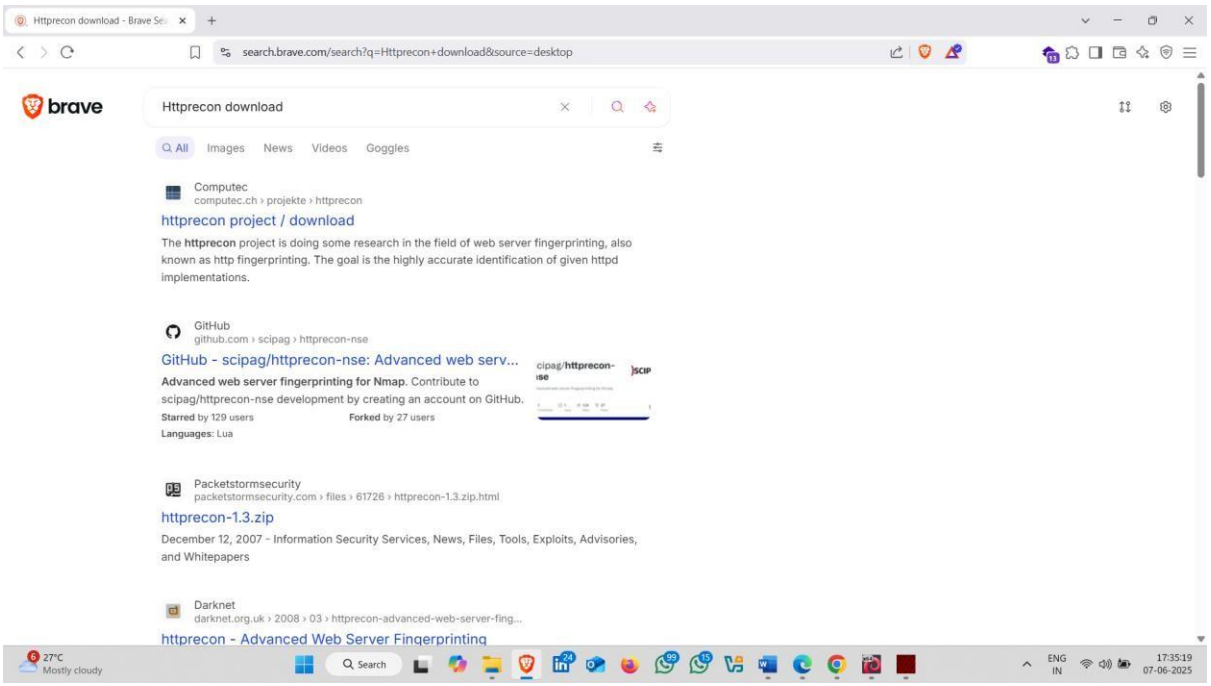
3.Perform Footprinting Using HTTPRecon

An **HTTP Recon Application** is a tool or software used in cybersecurity and penetration testing to gather detailed information about a web server or web application by interacting with its HTTP interface.

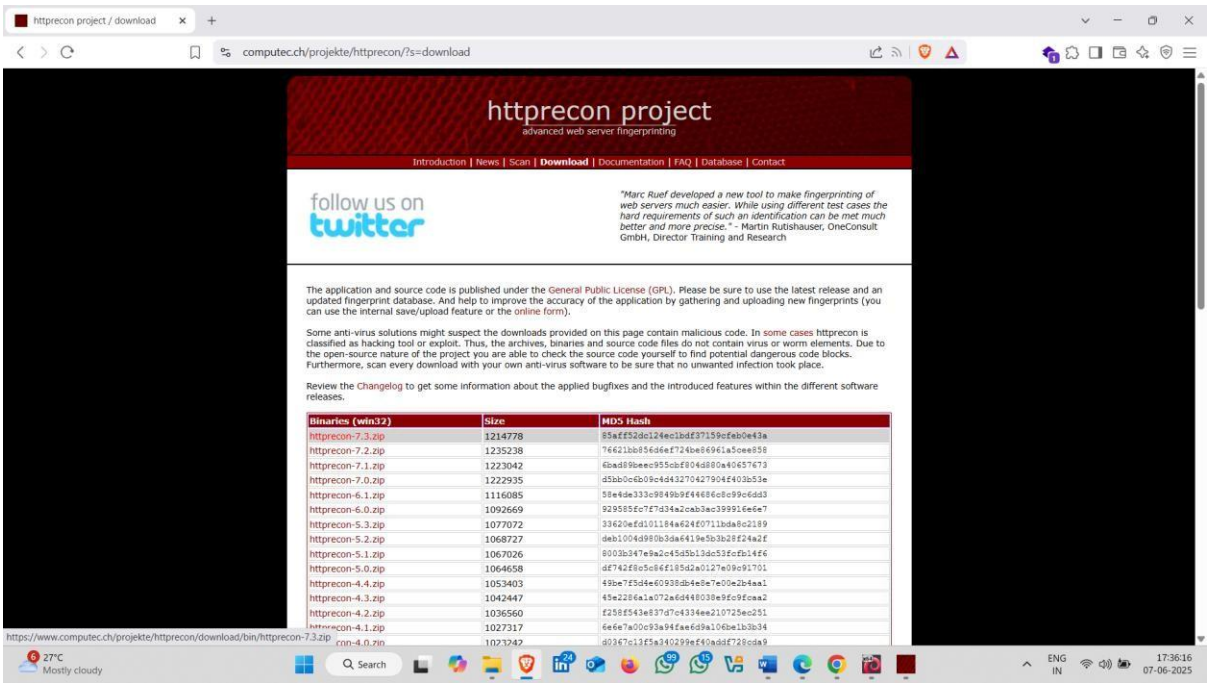
Download Link-:
https://www.computec.ch/projekte/httprecon/?s=download

How to Download it -:

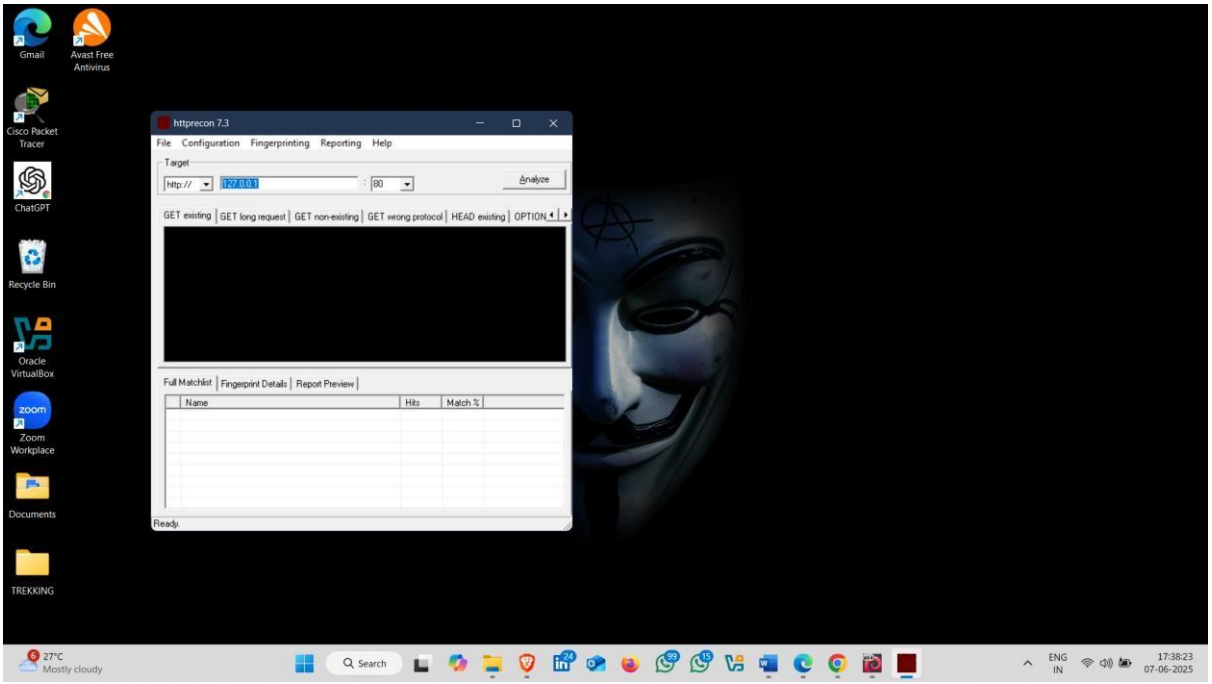
- Open Browser and Search HttpRecon Download
- Click on First Website



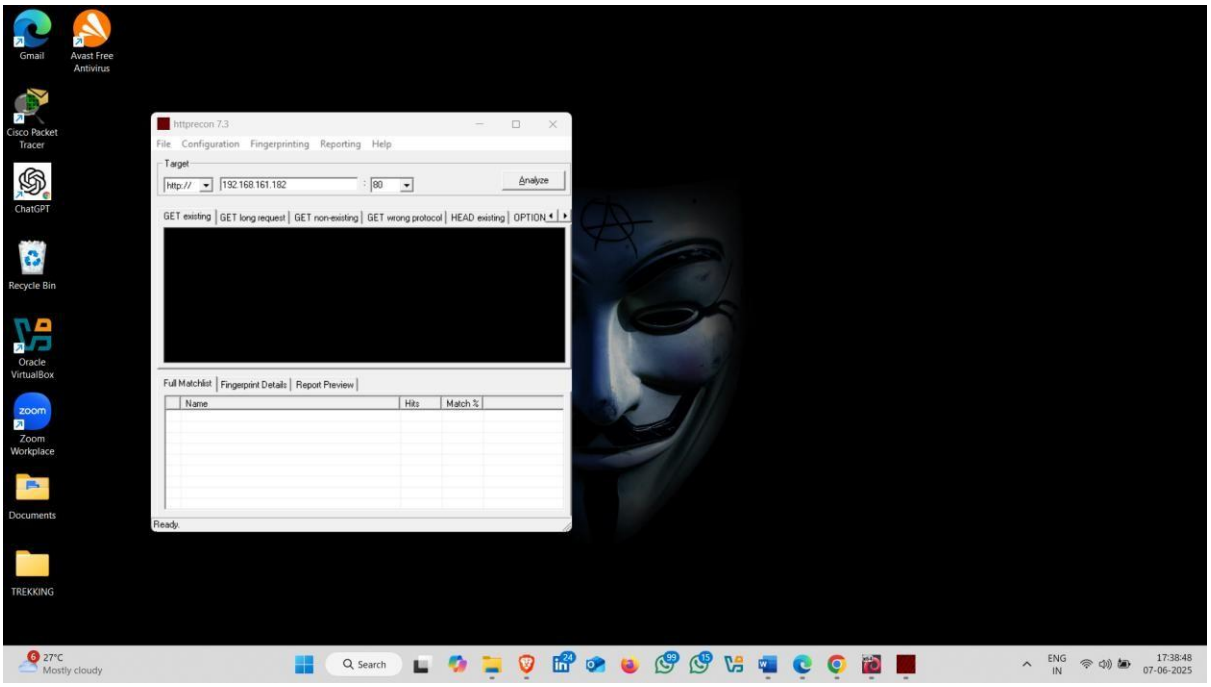
- **Download Latest Version**



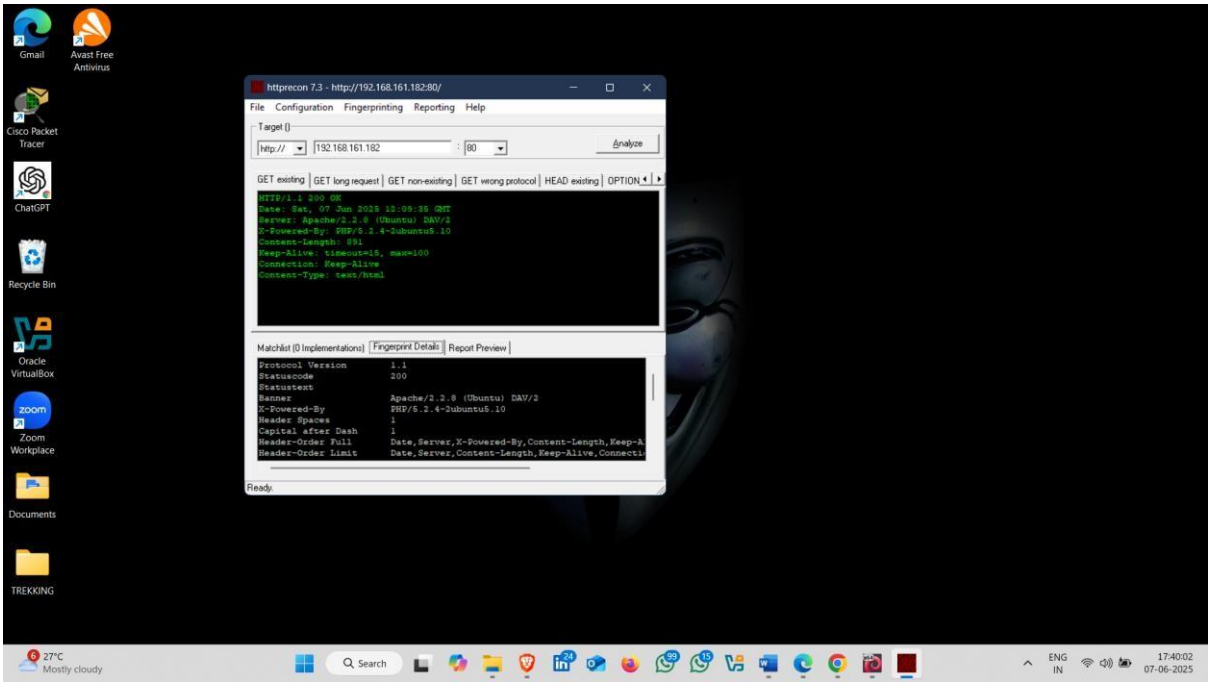
- **After Download open it**



- **Enter Target Ip Address and click on Analyse**


Gmail
Avast Free Antivirus
Cisco Packet Tracer
ChatGPT
Recycle Bin
Oracle VirtualBox
Zoom Workplace
Documents
TREKKING
httprecon 7.3
File Configuration Fingerprinting Reporting Help
Target
http://
192.168.161.182
80
Analyze
GET existing | GET long request | GET non-existing | GET wrong protocol | HEAD existing | OPTION
Full Matchlist | Fingerprint Details | Report Preview
Name
Hits
Match %
Ready.
27°C
Mostly cloudy
Search
ENG
IN
17:38:48
07-06-2025

Result


Gmail
Avast Free Antivirus
Cisco Packet Tracer
ChatGPT
Recycle Bin
Oracle VirtualBox
Zoom Workplace
Documents
TREKKING
httprecon 7.3 - http://192.168.161.182:80/
File Configuration Fingerprinting Reporting Help
Target ()
http://
192.168.161.182
80
Analyze
GET existing | GET long request | GET non-existing | GET wrong protocol | HEAD existing | OPTION
HTTP/1.1 200 OK
Date: Sat, 07 Jun 2025 13:09:36 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Length: 891
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
Matchlist (0 Implementations) | Fingerprint Details | Report Preview
Protocol Version 1.1
Statuscode 200
Statustext
Banner Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By PHP/5.2.4-2ubuntu5.10
Header Spaces 1
Capital after Dash 1
Header-Order Full Date,Server,X-Powered-By,Content-Length,Keep-A
Header-Order Limit Date,Server,Content-Length,Keep-Alive,Connecti
Ready.
27°C
Mostly cloudy
Search
ENG
IN
17:40:02
07-06-2025

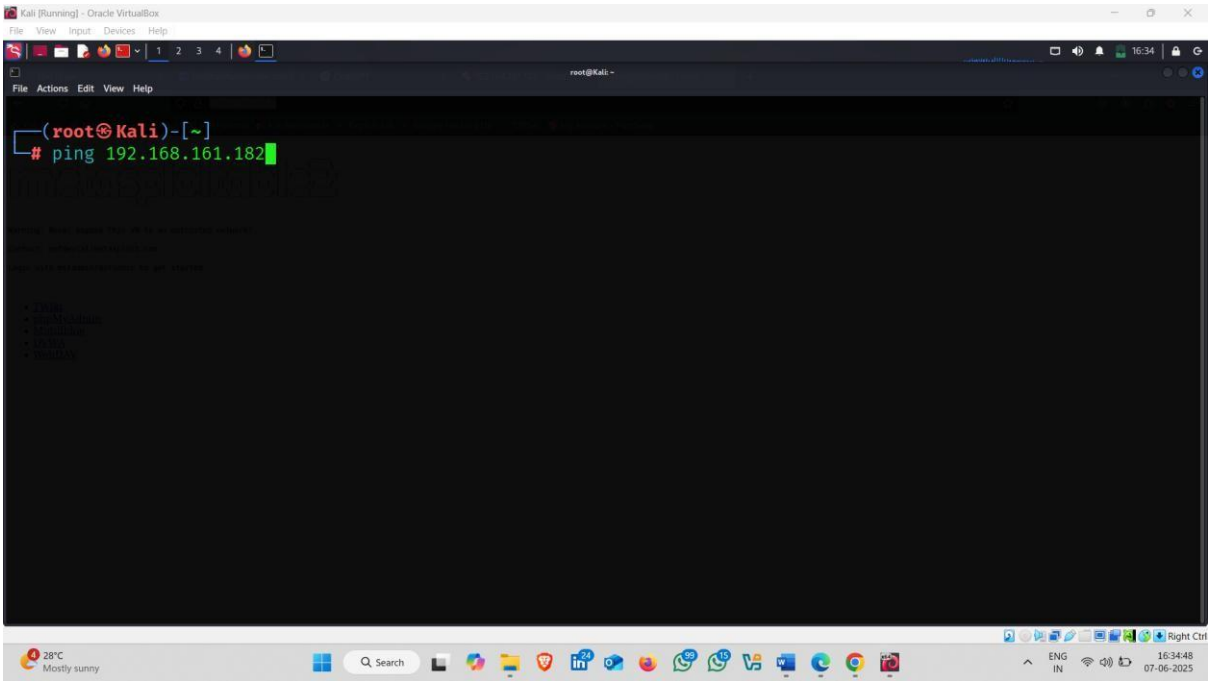
Scanning

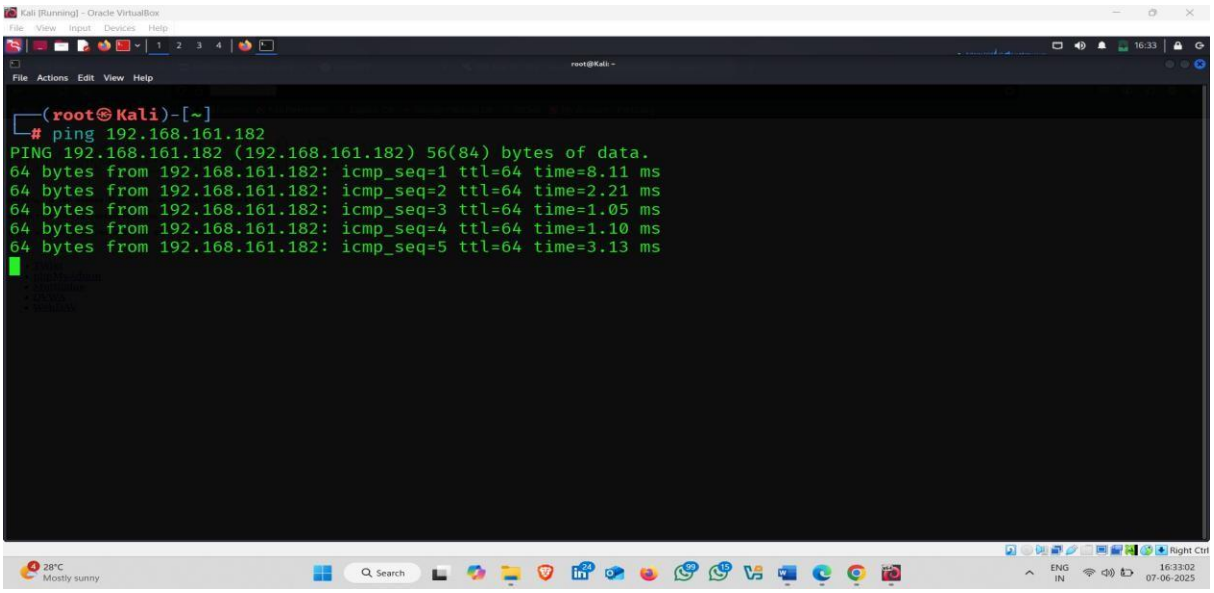
Scanning is a process used in cybersecurity and network management to actively probe and analyze a target system or network to gather information about its structure, open ports, running services, and potential vulnerabilities.

Check Host Is Alive Or Not Using Various

A)Ping

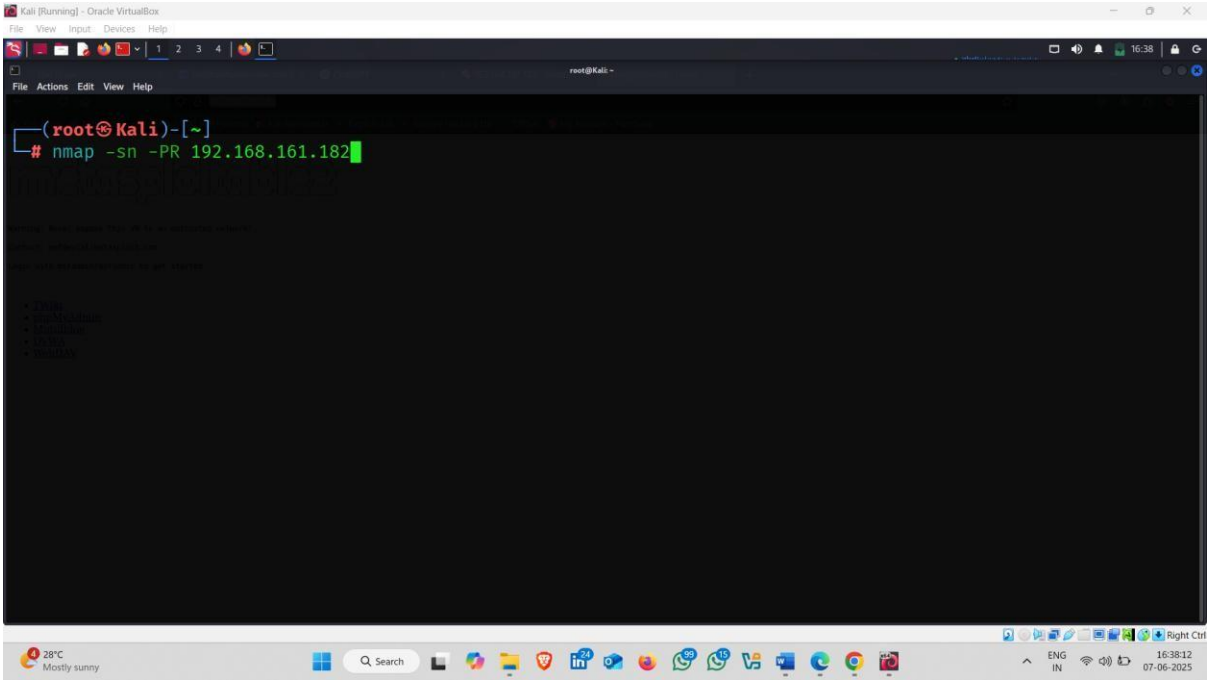
Command-: ping 192.168.161.182

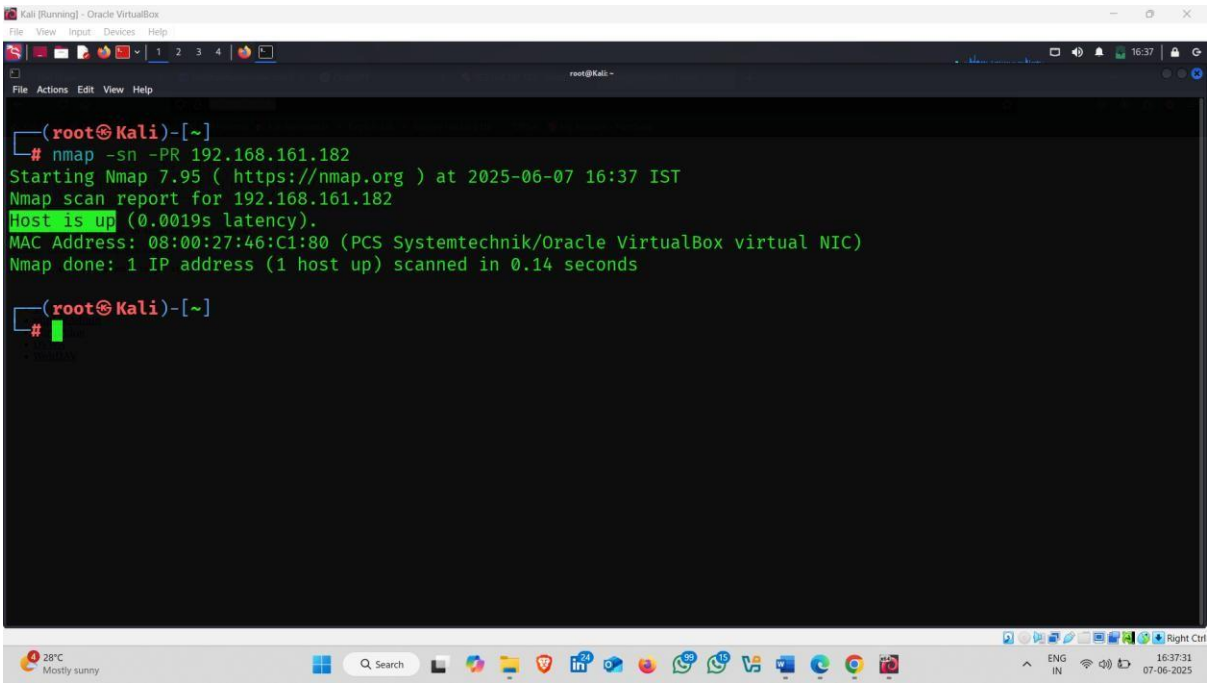




B) Nmap-:

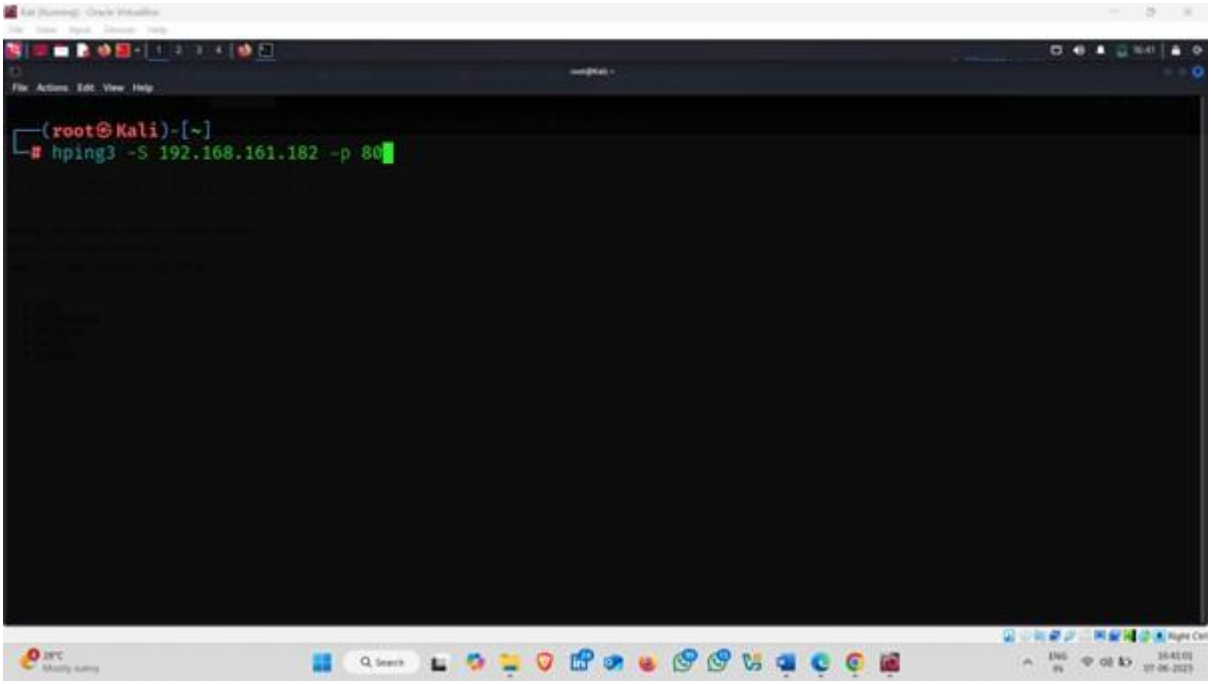
Command-: nmap -sn -PR 192.168.161.182



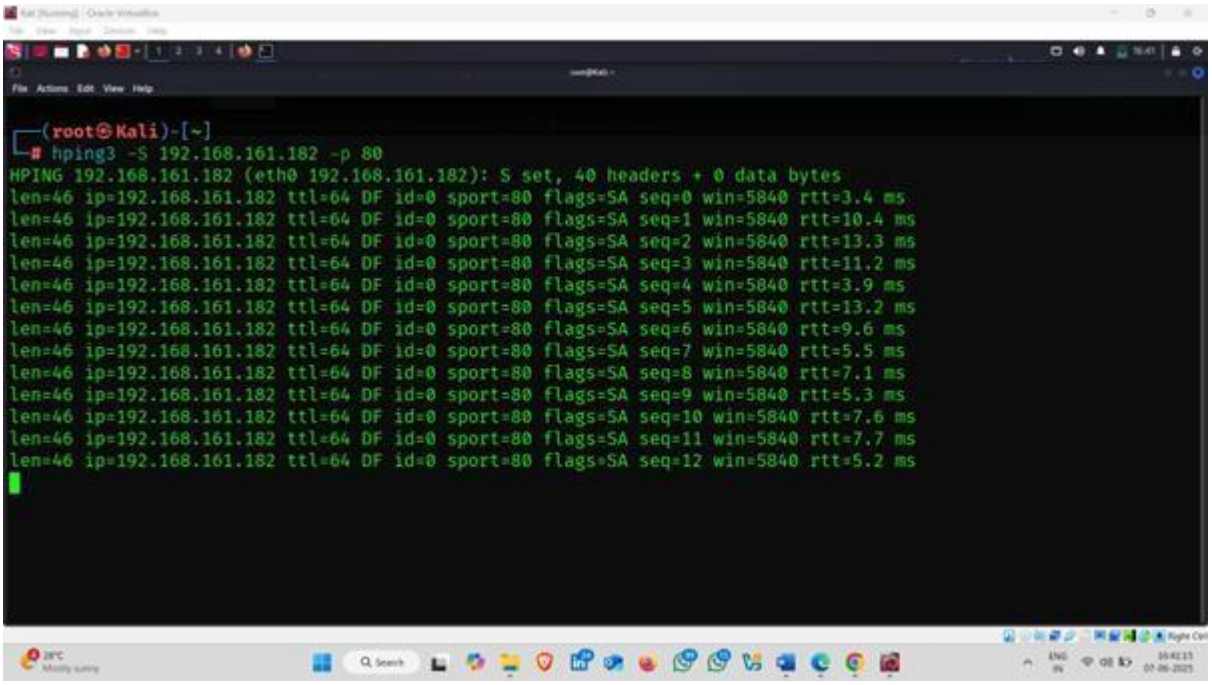


C) Hping3-:

Command – hping3 -S 192.168.161.182 -p 80



- Response is received



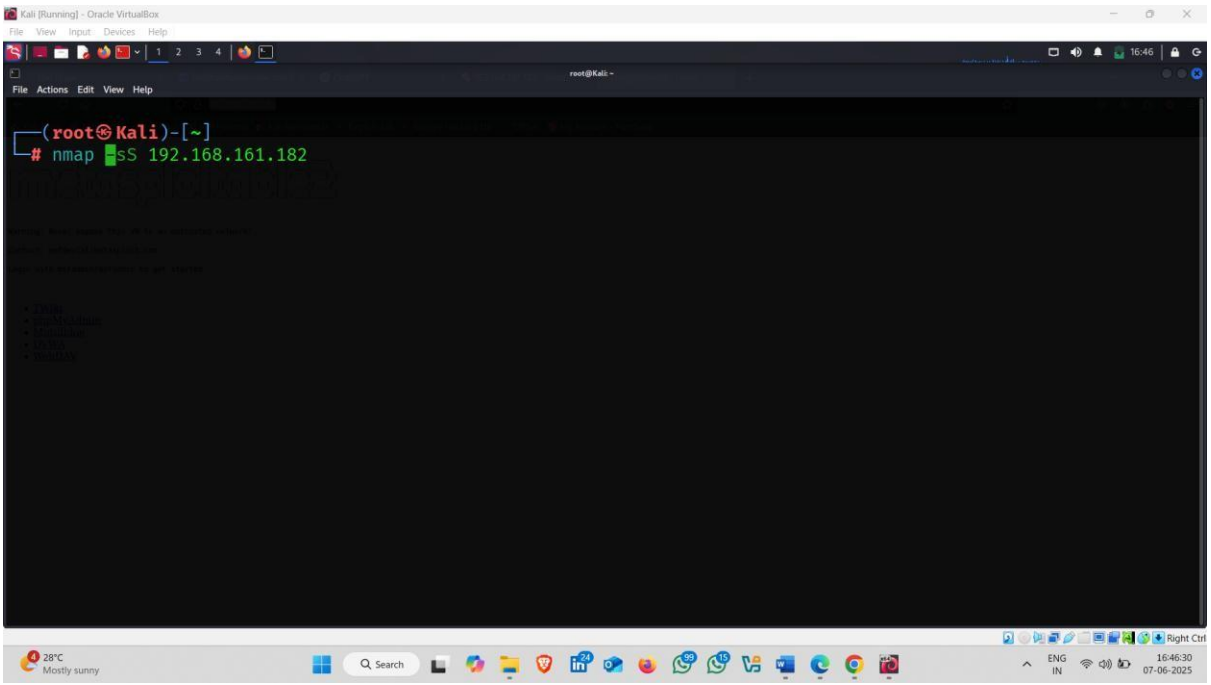
After Finding the host is alive or note using different techniques the next step find the open ports on target

---

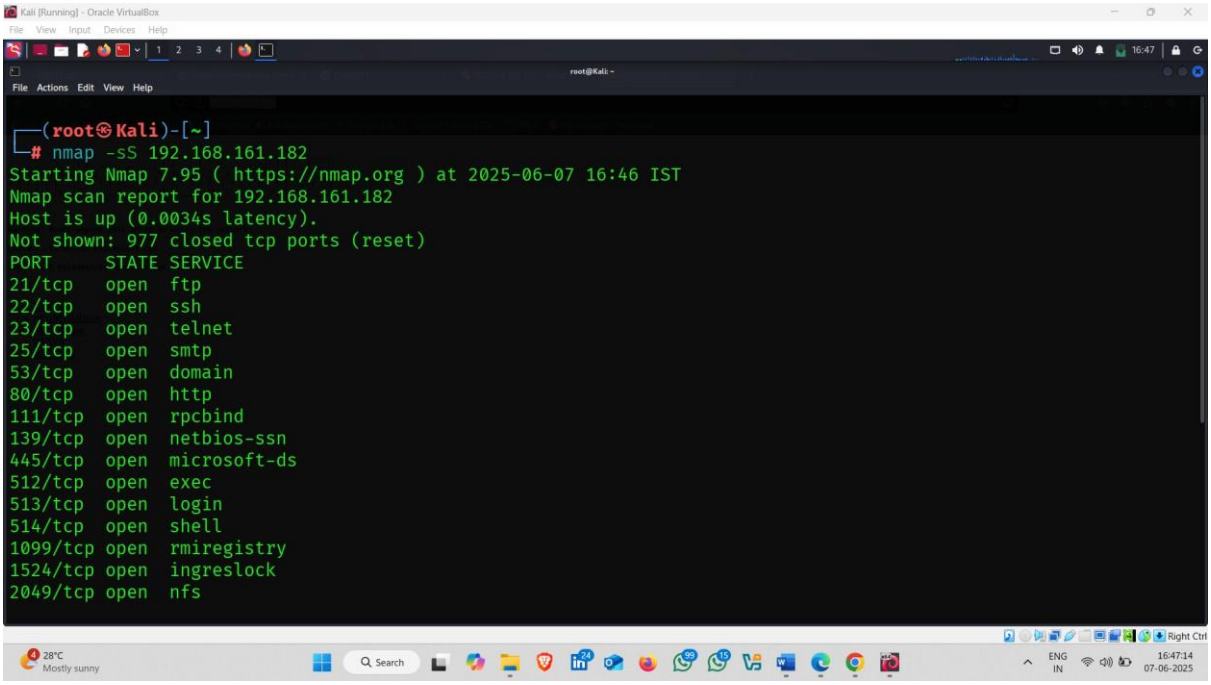
Finding Open Ports Using Different Techniques

A) Nmap -:

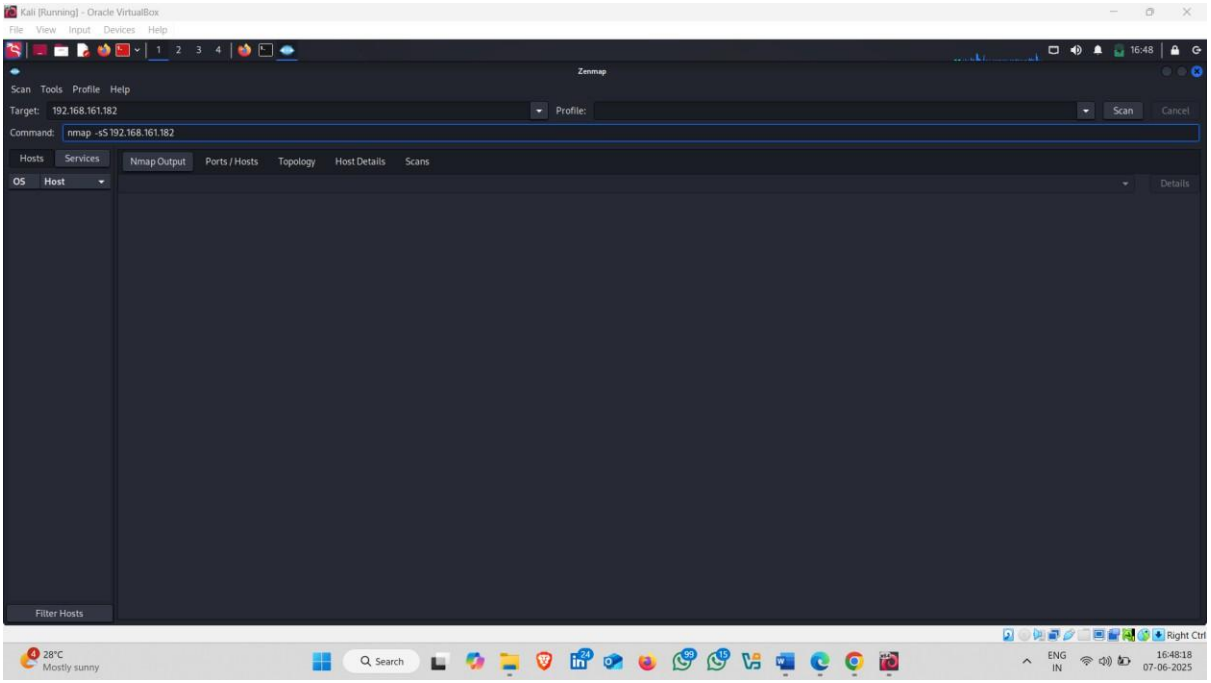
Command -: nmap -sS 192.168.161.182



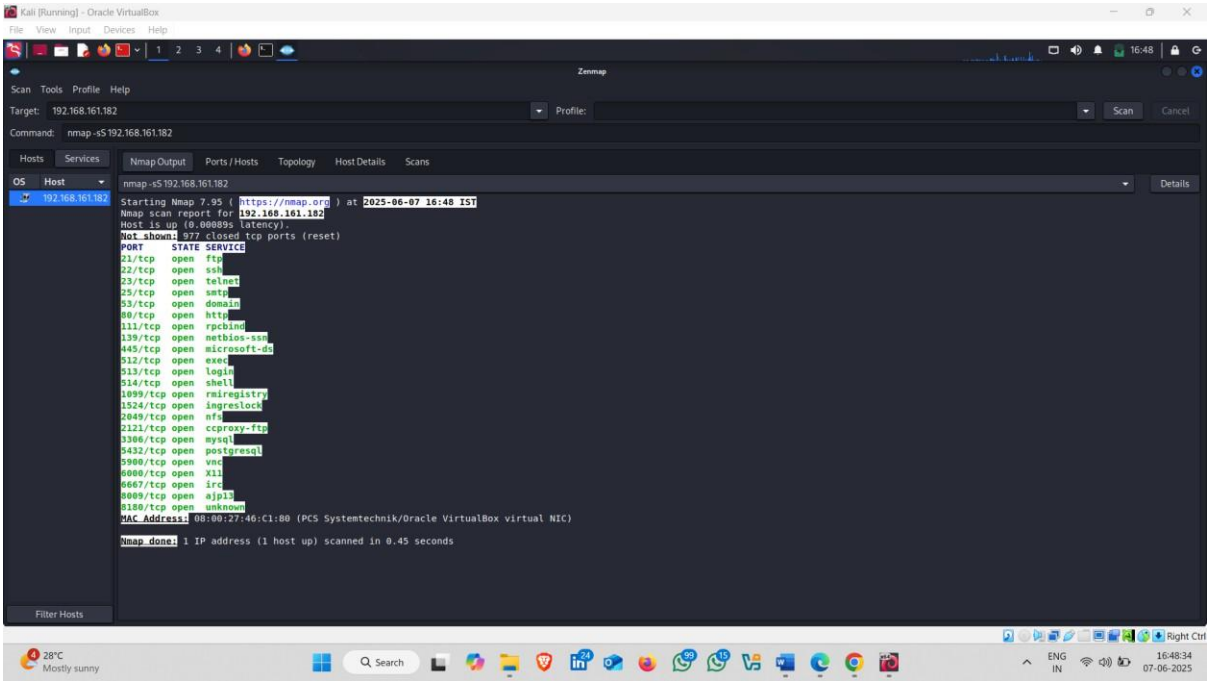
- **Target Open Ports**



B) Zenmap-: Zenmap is the graphical Version of Nmap



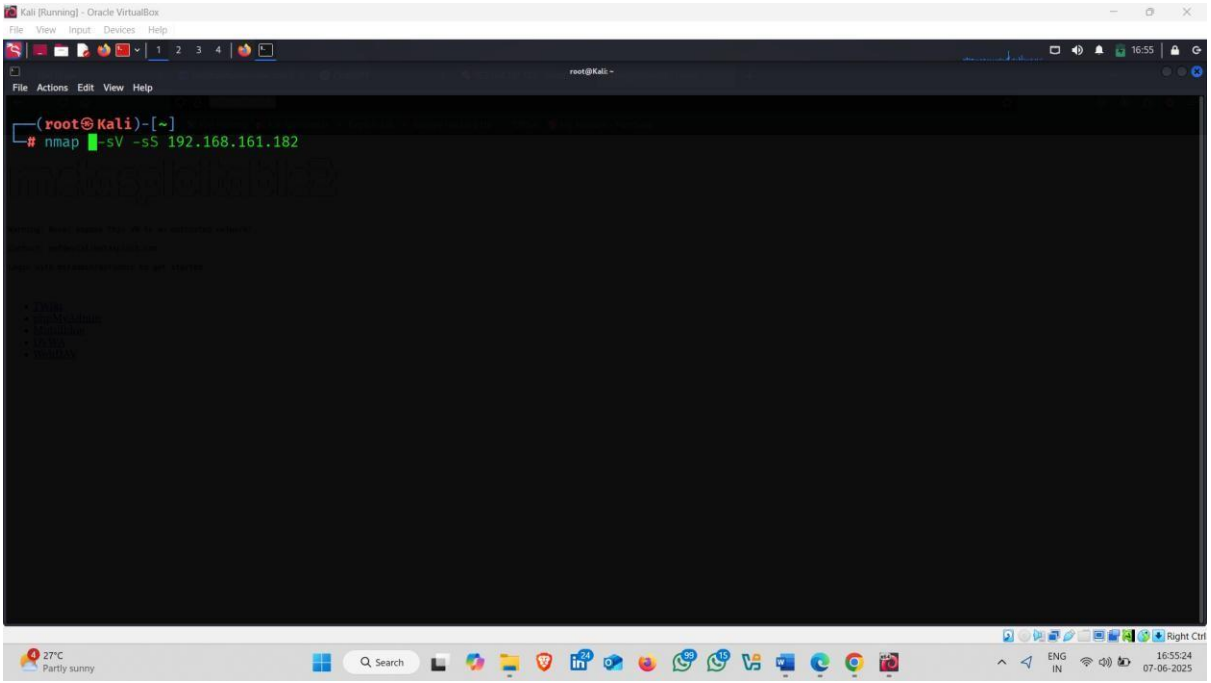
Result-:



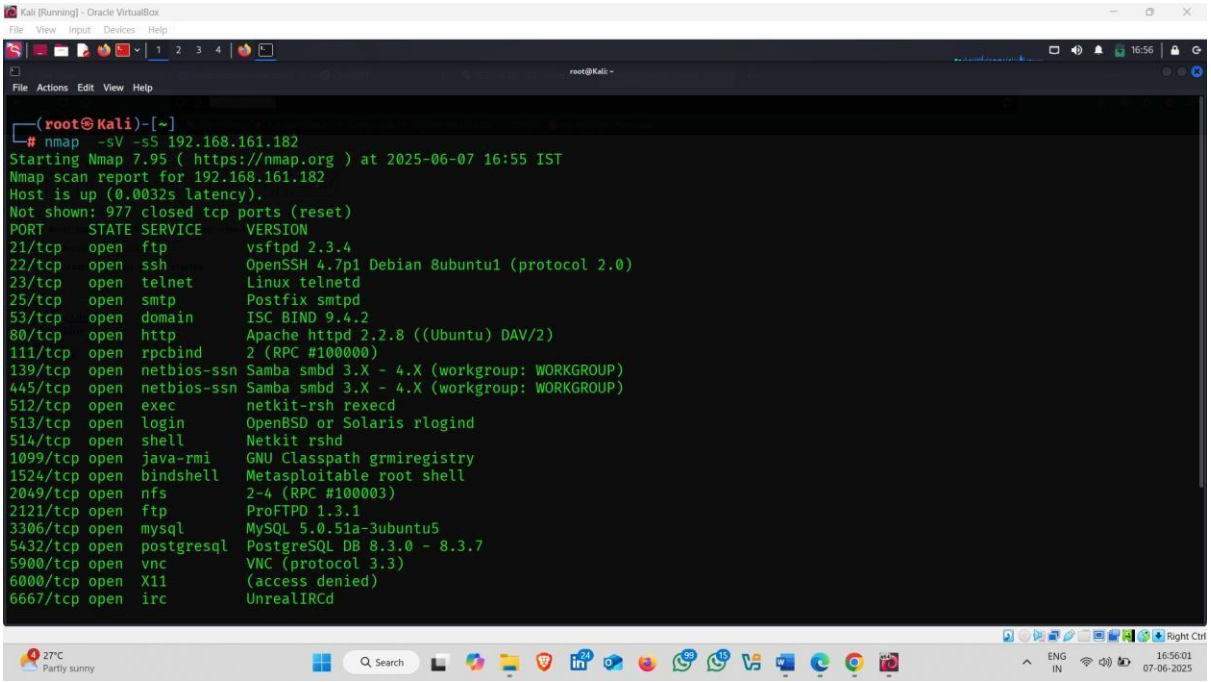
Finding Service Version Using Nmap and zenmap

A) Nmap-:

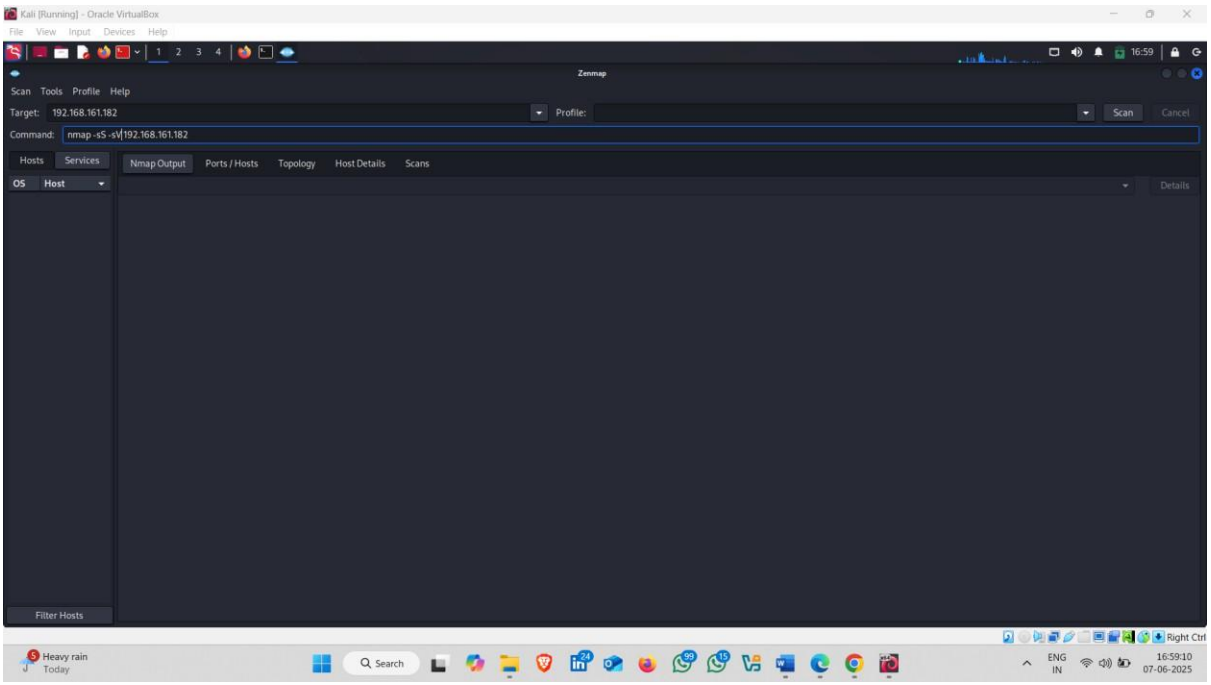
Command -: nmap -sV -sS 192.168.161.182



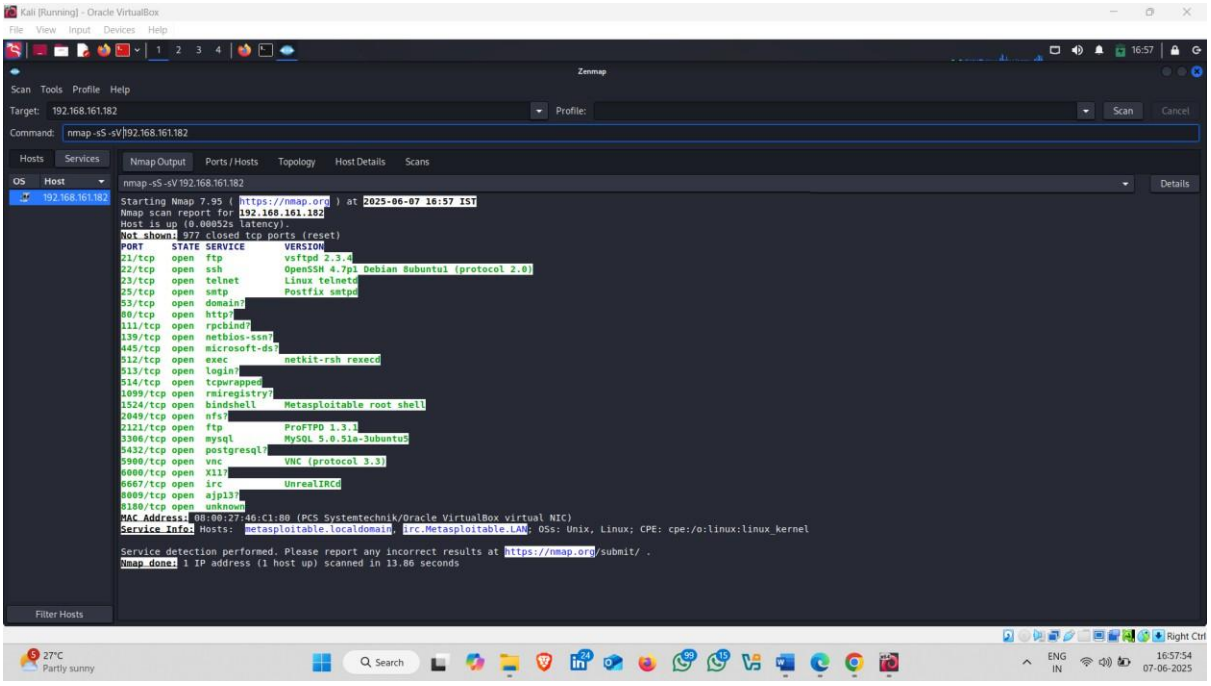
Result-:



B) Zenmap-:



Result-:



After Finding the open ports and service version finding the vulnerability

Vulnerability Analysis

Vulnerability Analysis is the process of **identifying, classifying, and evaluating security weaknesses (vulnerabilities)** in a system, network, application, or infrastructure **before attackers can exploit them**.

Purpose of Vulnerability Analysis:

- To **find weaknesses** in software, hardware, or configurations.
- To **prevent cyber attacks** by fixing known flaws.
- To **assess risk levels** of different vulnerabilities.
- To **help prioritize remediation** efforts based on severity.

Finding Vulnerability Using Various Tools

vulnerability analysis tools

1. **Nessus**
2. **OpenVAS**
3. **Nikto**
4. **Acunetix**
5. **Burp Suite**
6. **Nexpose (Rapid7)**
7. **Qualys**
8. **Netsparker**
9. **Arachni**
10. **OWASP ZAP**

11. **Wapiti**
12. **Vega**
13. **IBM AppScan**
14. **Retina**
15. **GFI LanGuard**
16. **SAINT**
17. **Microsoft Baseline Security Analyzer (MBSA)**
18. **Core Impact**
19. **Invicti (formerly Netsparker)**
20. **Tenable.io**

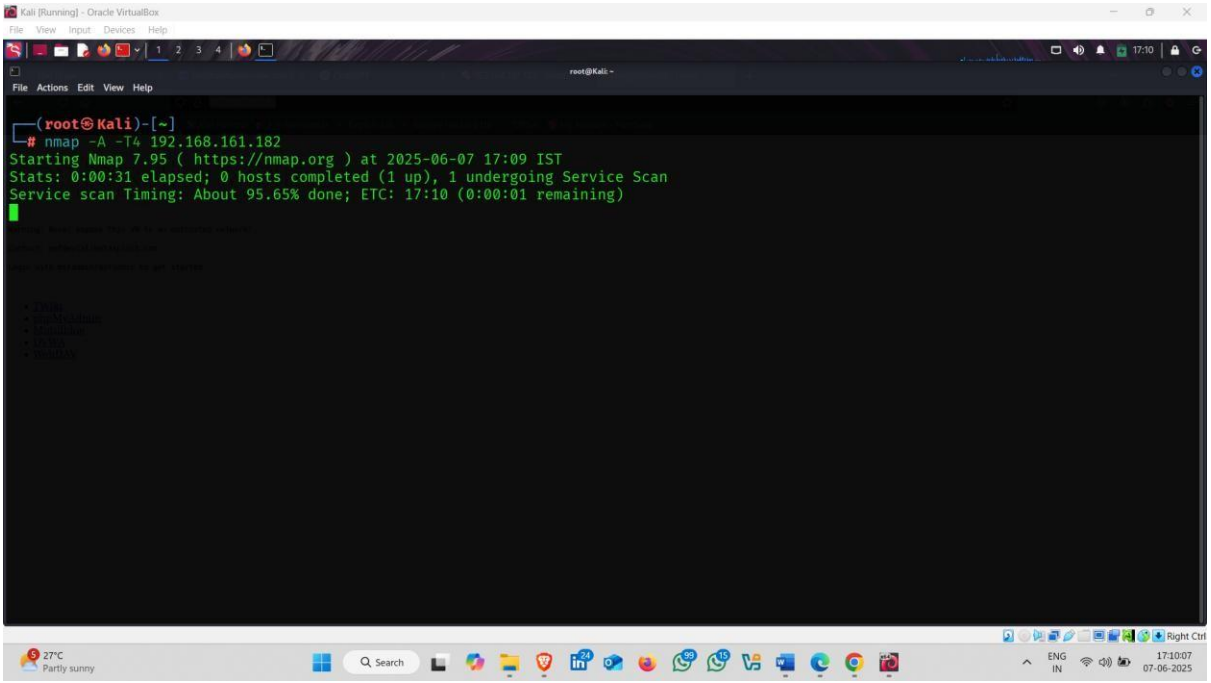
A) nmap Script-:

Command -: nmap -A -T4 192.168.161.182

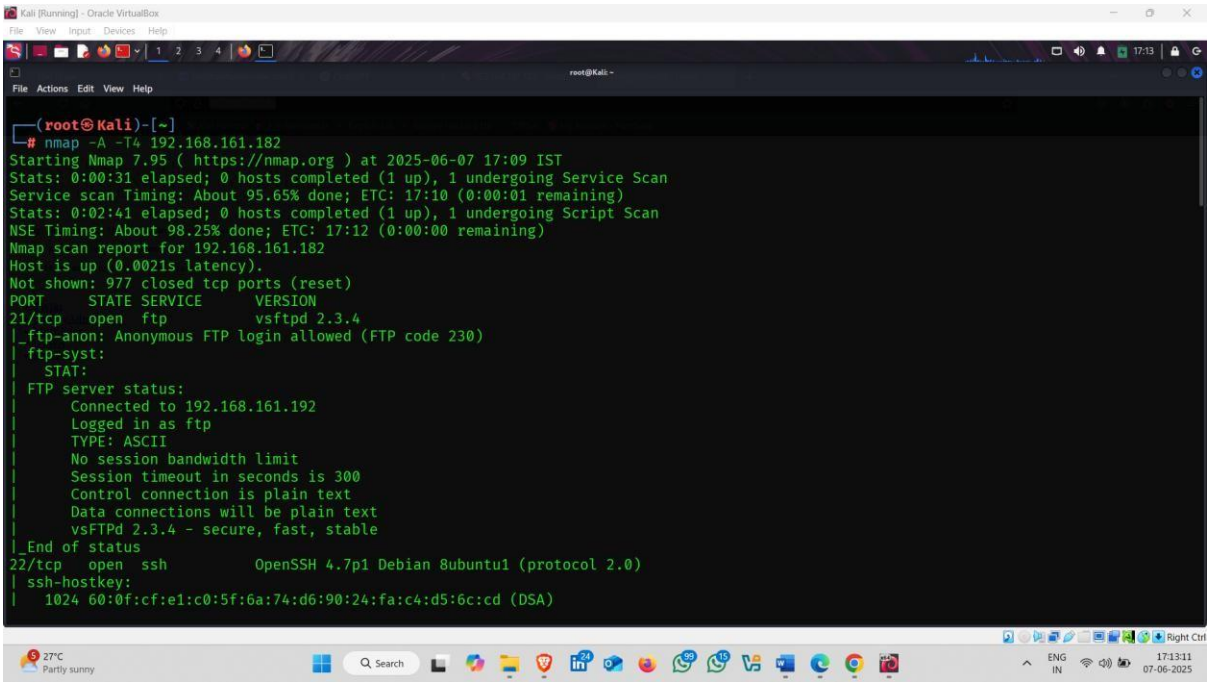
How It Works:

- nmap:
  Launches the Nmap tool (used for network scanning and host discovery).
- -A:
  Stands for **Aggressive Scan**. It enables:
  - **OS detection**
  - **Version detection** of services
  - **Script scanning** (default NSE scripts)
  - **Traceroute**
- -T4:
  Specifies **timing template**. T4 makes the scan faster and is ideal for LAN networks.
  (Range: T0 [slowest] to T5 [fastest]).

- 192.168.161.182:
  This is the **target IP address** that will be scanned.



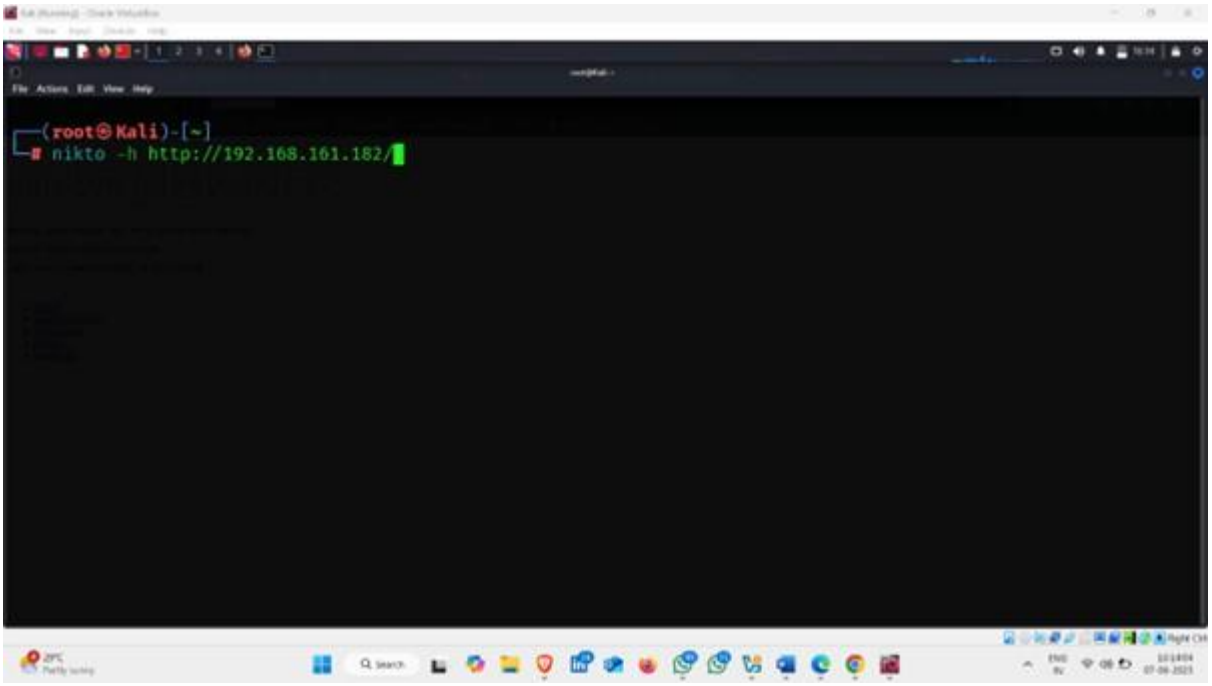
Result-:



B) Nikto-:

Command -: nikto -h http://192.168.161.182

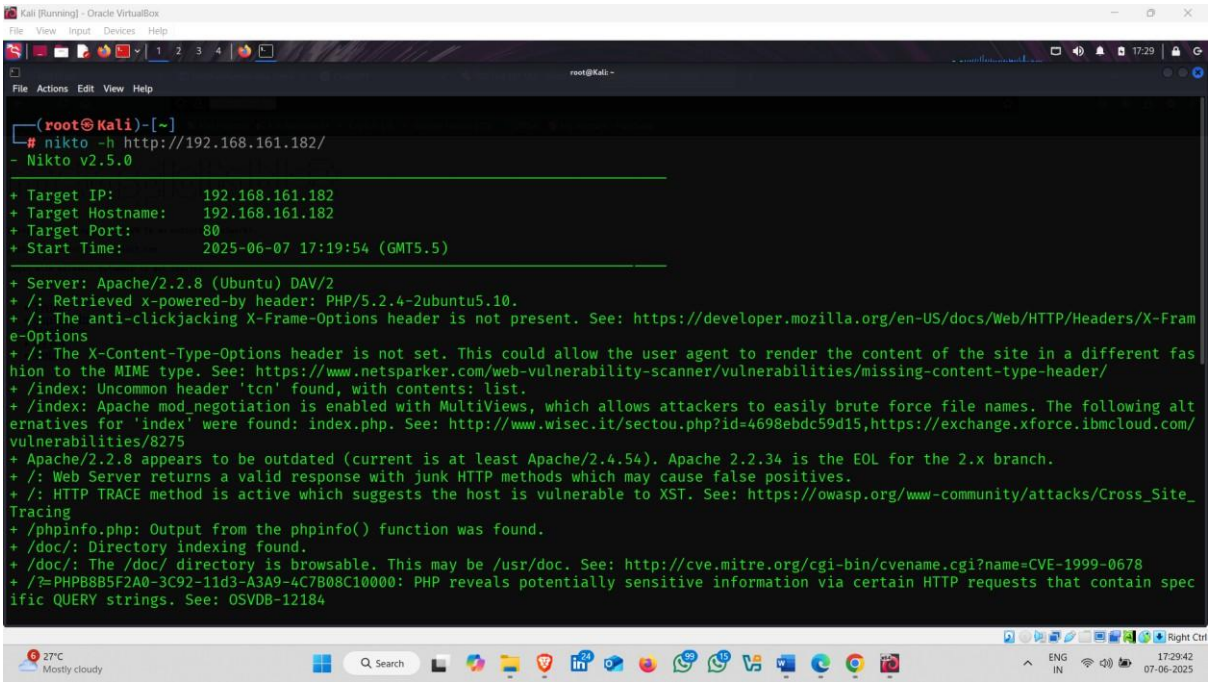
- **nikto – This is a web server vulnerability scanner.**
- **-h – This option is used to specify the host (target web server).**
- **http://192.168.161.182/ – This is the URL or IP address of the web server you want to scan.**



Result -: Vulnerabilities

1. **Apache Server Version:** Using outdated Apache/2.2.8, which has known security risks.
2. **PHP Version:** PHP 5.2.4 detected; this version is outdated and vulnerable.
3. **Missing Security Headers:**
   a. No **X-Frame-Options** header (risk of clickjacking).
   b. No **X-Content-Type-Options** header (risk of MIME type confusion).

4. **Mod_negotiation MultiViews Enabled:** Can allow attackers to brute-force file names.
5. **TRACE HTTP Method Enabled:** Vulnerable to Cross-Site Tracing (XST) attacks.
6. **Directory Indexing Enabled:** Allows browsing of folders like /doc/, /test/, /icons/.
7. **phpinfo.php Script Present:** Reveals sensitive system information.
8. **phpMyAdmin Accessible:** Exposes database management interface, should be restricted.
9. **Sensitive Files Exposed:** Files like wp-config.php found, which may contain credentials.
10. **Junk HTTP Methods Allowed:** Server responds to uncommon or invalid HTTP methods.
11. **Potential Information Disclosure:** Certain URLs reveal PHP version and other data.



C) Acunetix -:

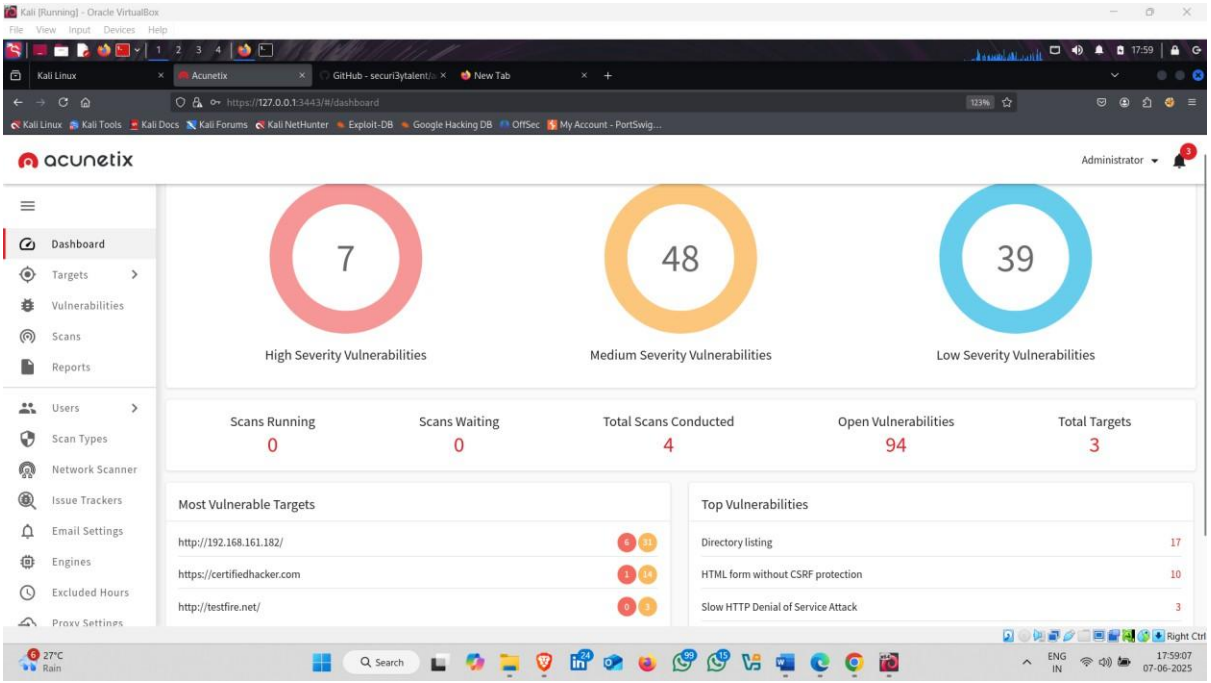
Acunetix is an **automated web application security scanner** designed to identify and help fix vulnerabilities in websites, web applications, and APIs.

Download Link -: https://github.com/securi3ytalent/acunetix-13-kali-linux

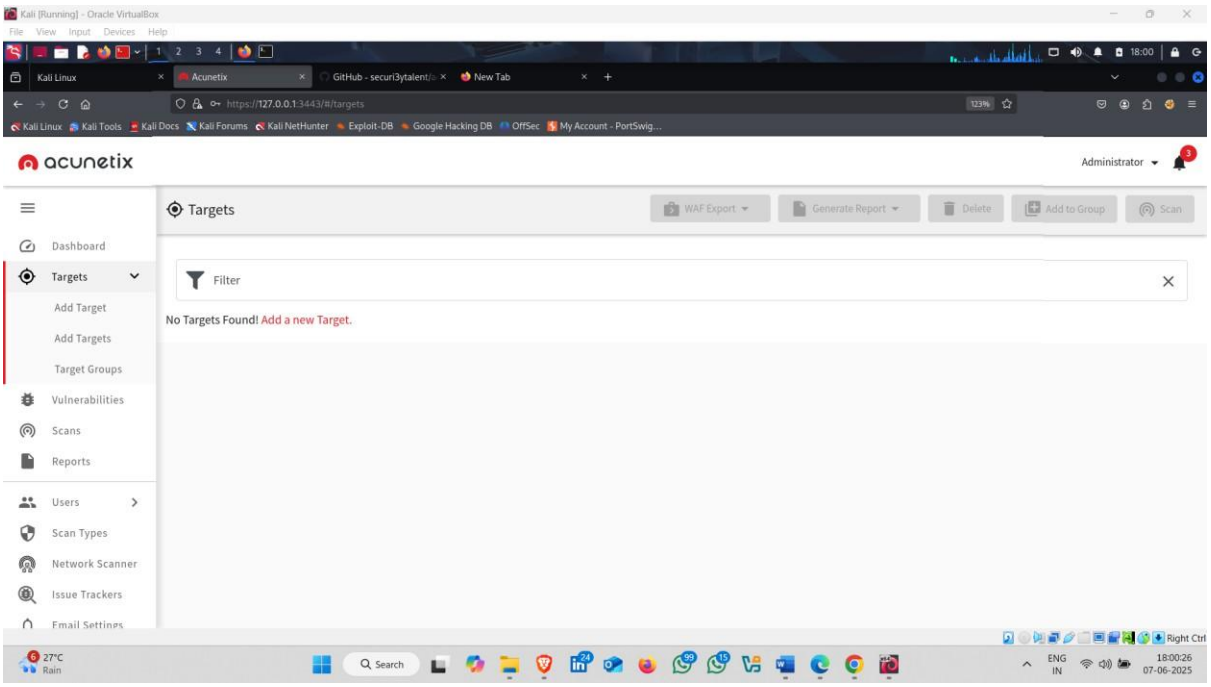
How to use it -:

After Downloading the Acunetix , setup and open it in browser localhost

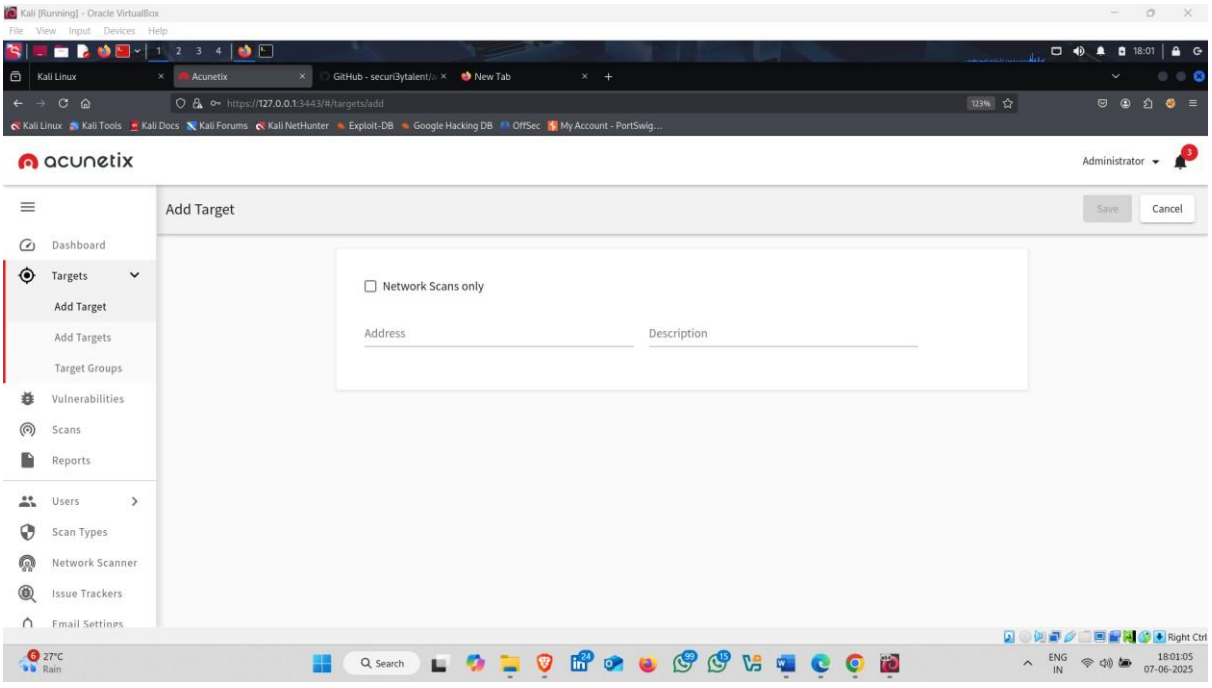
Acunetix



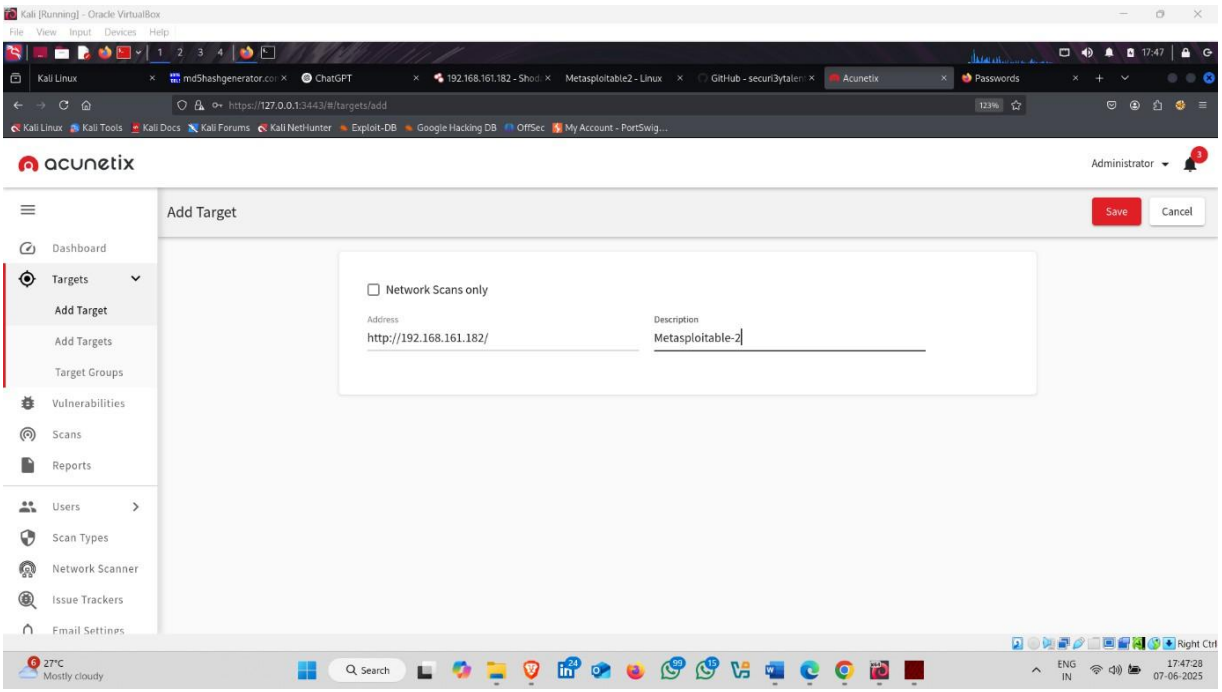
- Click on target



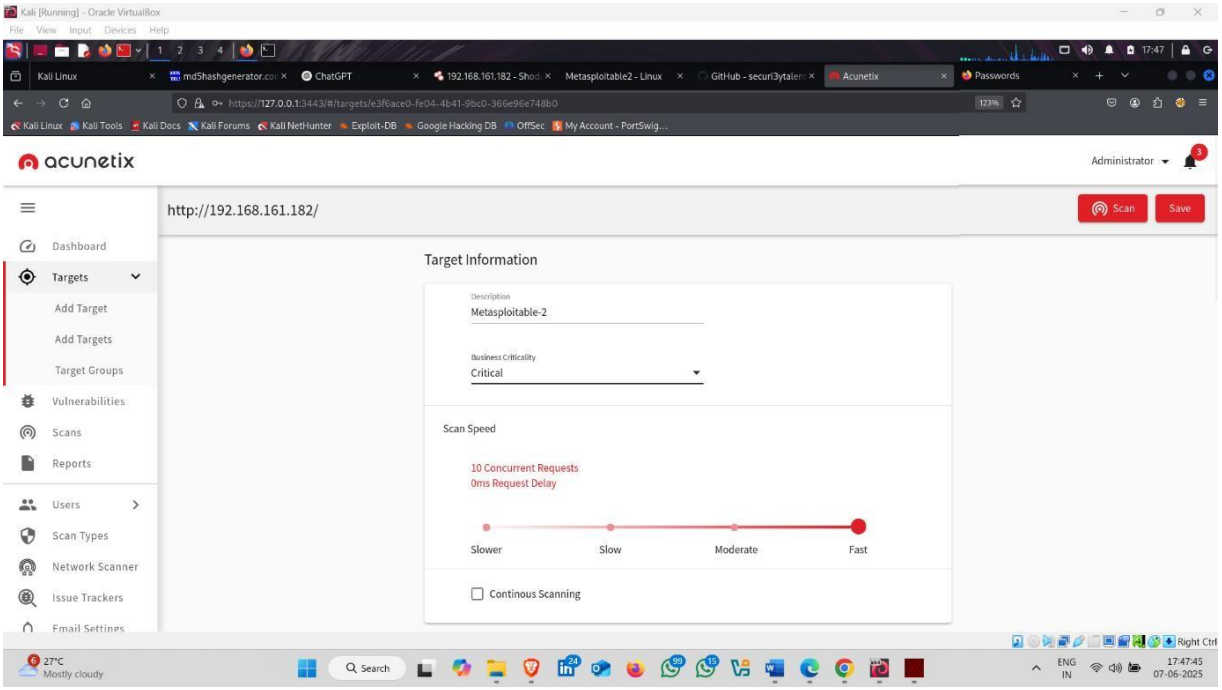
- click on add target



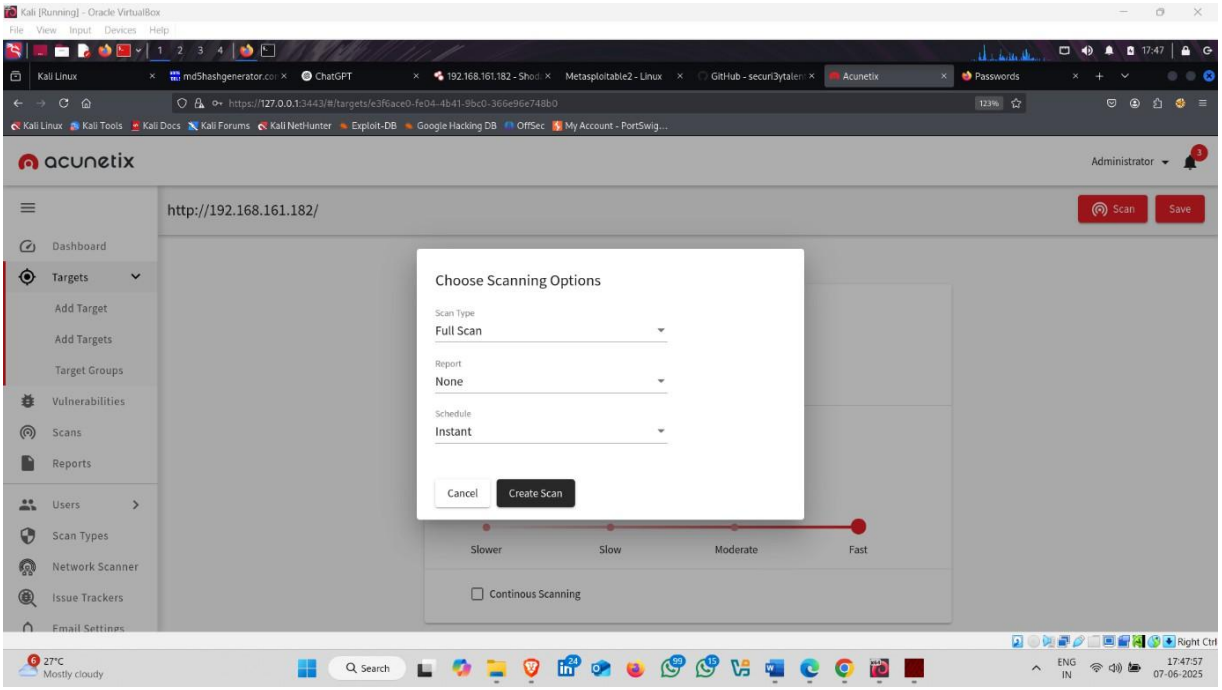
- Set the target address and description name



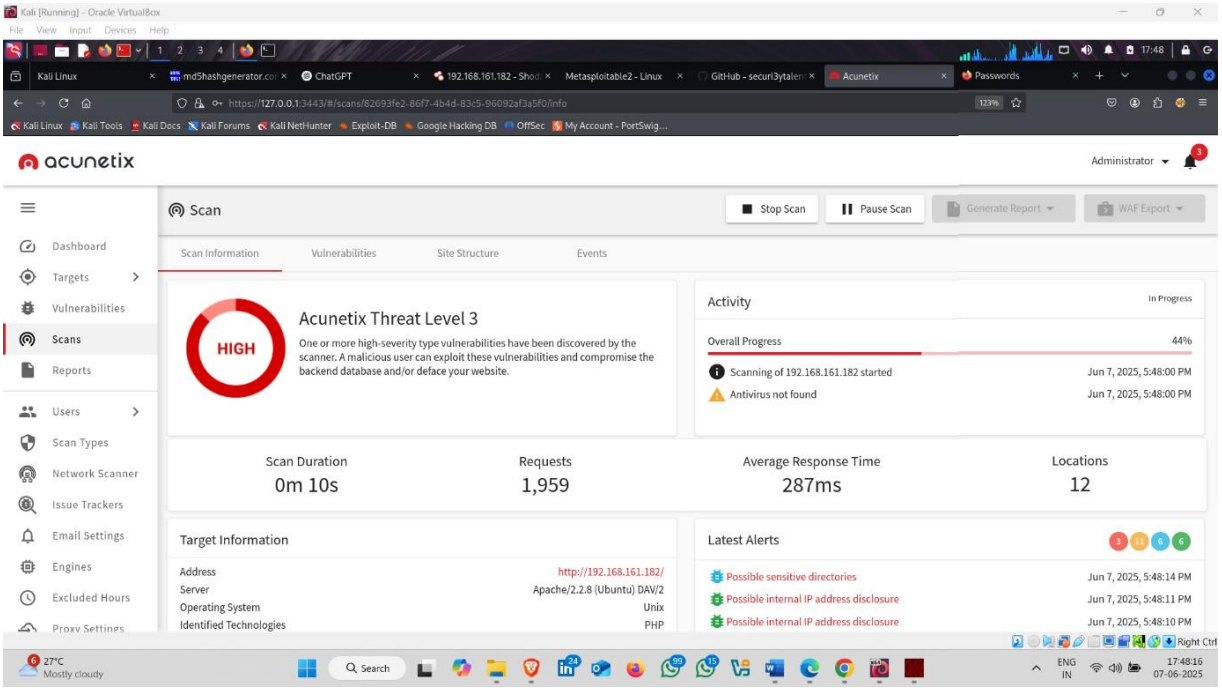
- Click on Scan button



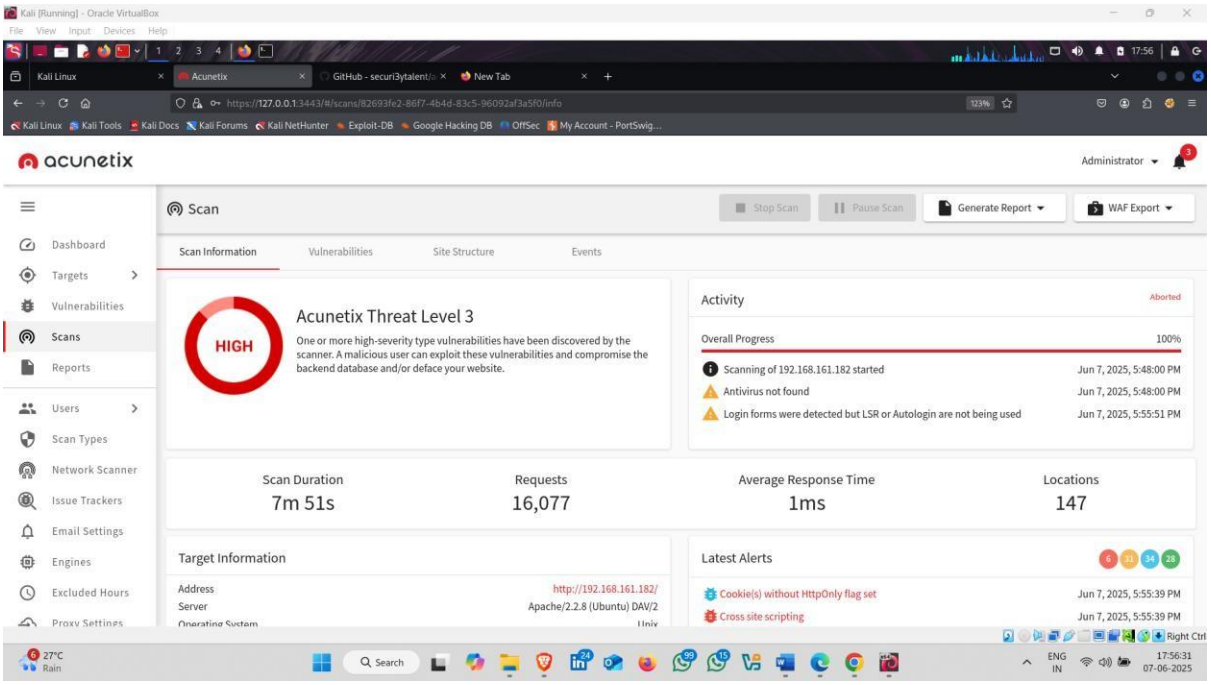
- Click on Create Scan



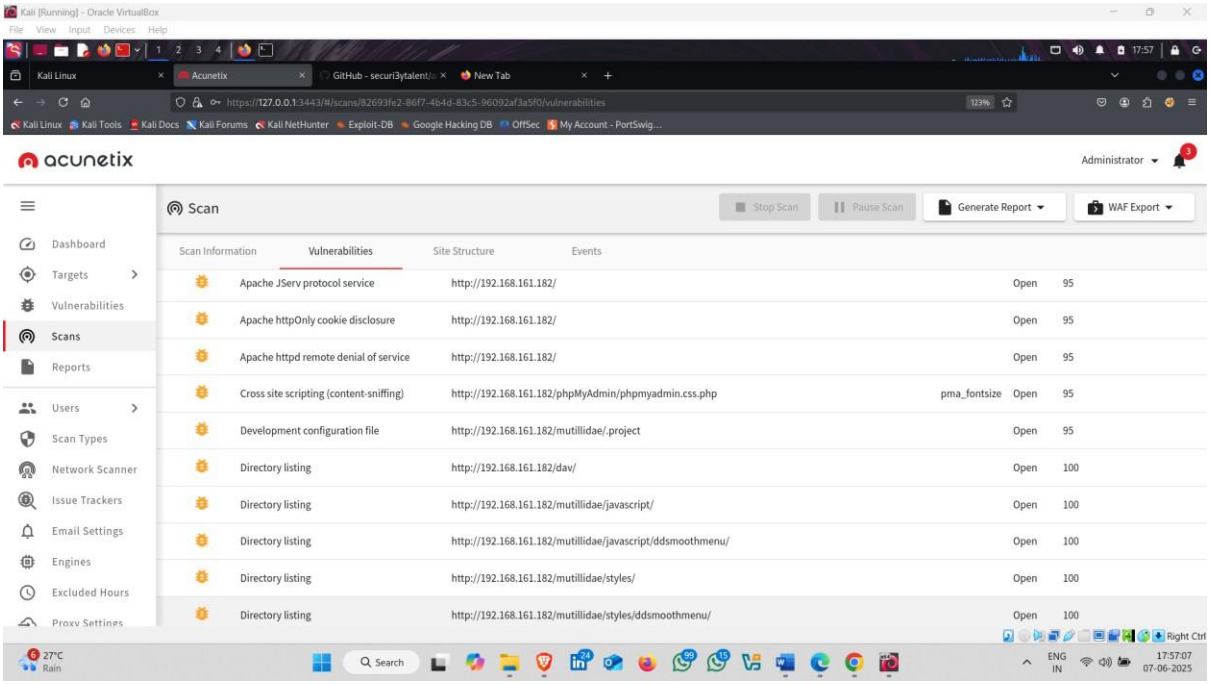
- Here scanning started



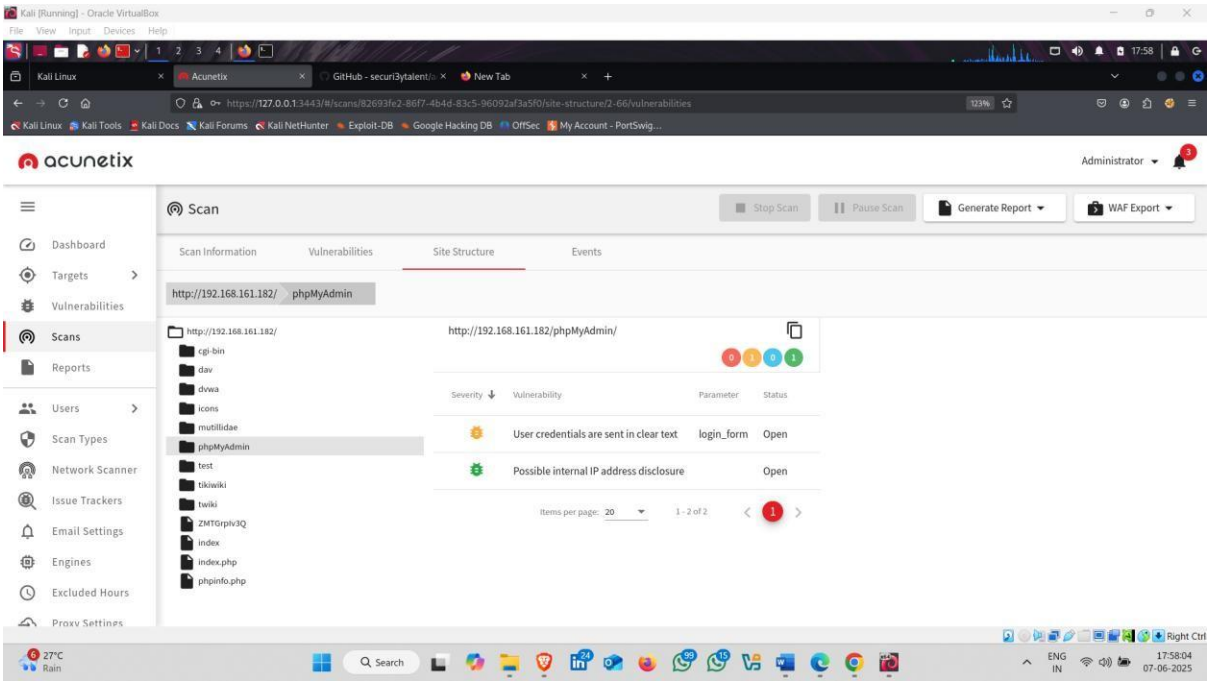
- **Scan completed**



- Click on Vulnerability section to see all vulnerability



- Click on site structure



We are done with web server Footprinting, host scanning, port scanning, Version Scanning, Vulnerability Scanning , now next step gaining access

Gaining Access

Open Ports Summary

Open Ports and Services Detected

- 21/tcp – FTP (vsftpd 2.3.4
- 22/tcp – SSH (OpenSSH 4.7p1 Debian 8ubuntu1)
- 23/tcp – Telnet (Linux telnetd)
- 25/tcp – SMTP (Postfix smtpd
- 53/tcp – DNS (ISC BIND 9.4.2)
- 80/tcp – HTTP (Apache httpd 2.2.8, PHP/5.2.4-2ubuntu5.10)
- 111/tcp – RPCbind (2)
- 139/tcp – NetBIOS-SSN (Samba smbd 3.X - WORKGROUP)
- 445/tcp – Microsoft-DS (Samba smbd 3.X - WORKGROUP)
- 512/tcp – exec
- 513/tcp – login (rlogin)
- 514/tcp – shell (rsh)
- 1099/tcp – Java RMI (rmiregistry)
- 1524/tcp – Metasploitable root shell
- 2049/tcp – NFS (rpc.statd)
- 2121/tcp – FTP (ProFTPD 1.3.1)
- 3306/tcp – MySQL (5.0.51a-3ubuntu5
- 5432/tcp – PostgreSQL (PostgreSQL DB 8.3.0 - 8.3.7)
- 5900/tcp – VNC (protocol 3.3)
- 6000/tcp – X11 (Access denied)

Password Cracking

1.Password Cracking Using Hydra

Hydra (also known as **THC Hydra**) is a popular, fast, and flexible **password cracking tool** used for **brute forcing login credentials** on various network services. It is widely used in penetration testing and ethical hacking to test the strength of passwords by attempting many combinations automatically.

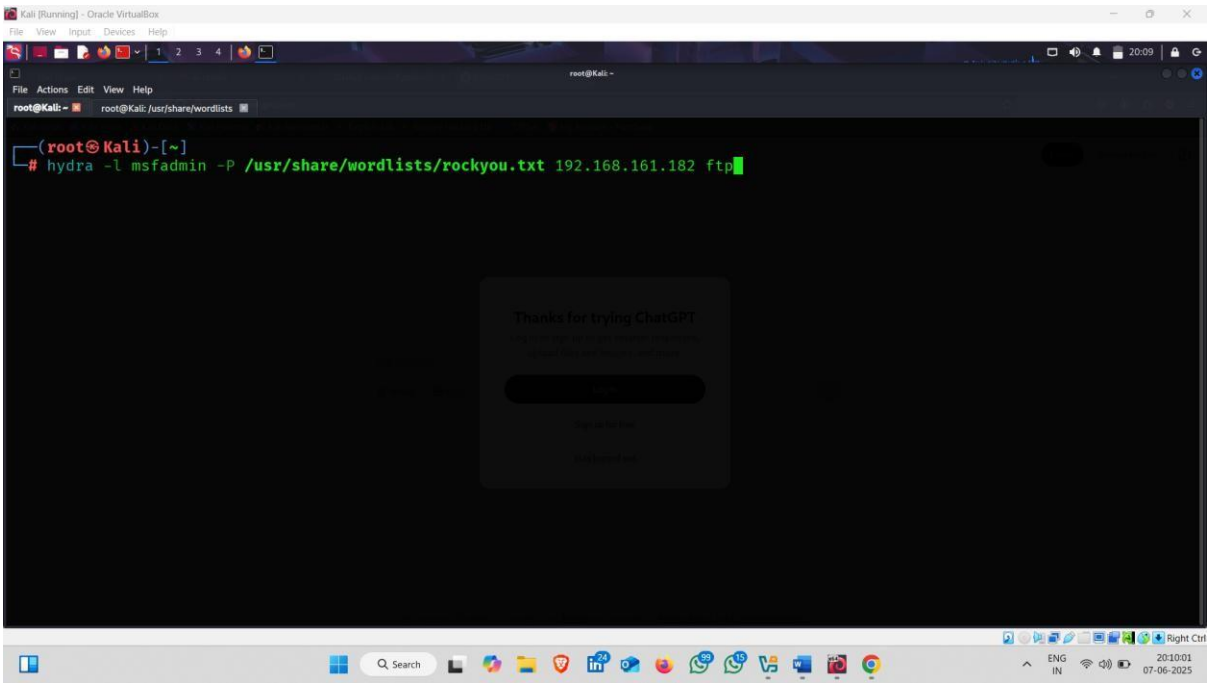
How to use it -:

- Open kali linux terminal and type following command

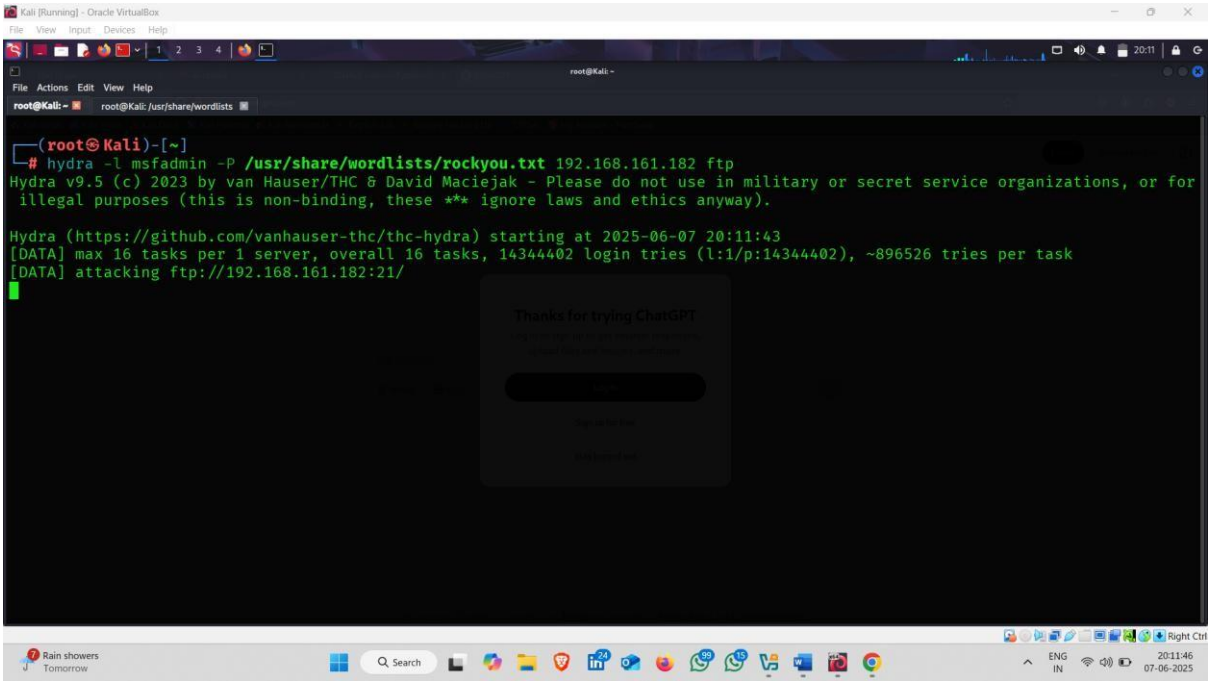
Command -: hydra -l msfadmin -P /usr/share/wordlists//rockyou.txt -F 192.168.161.182 ftp

Explanation-:

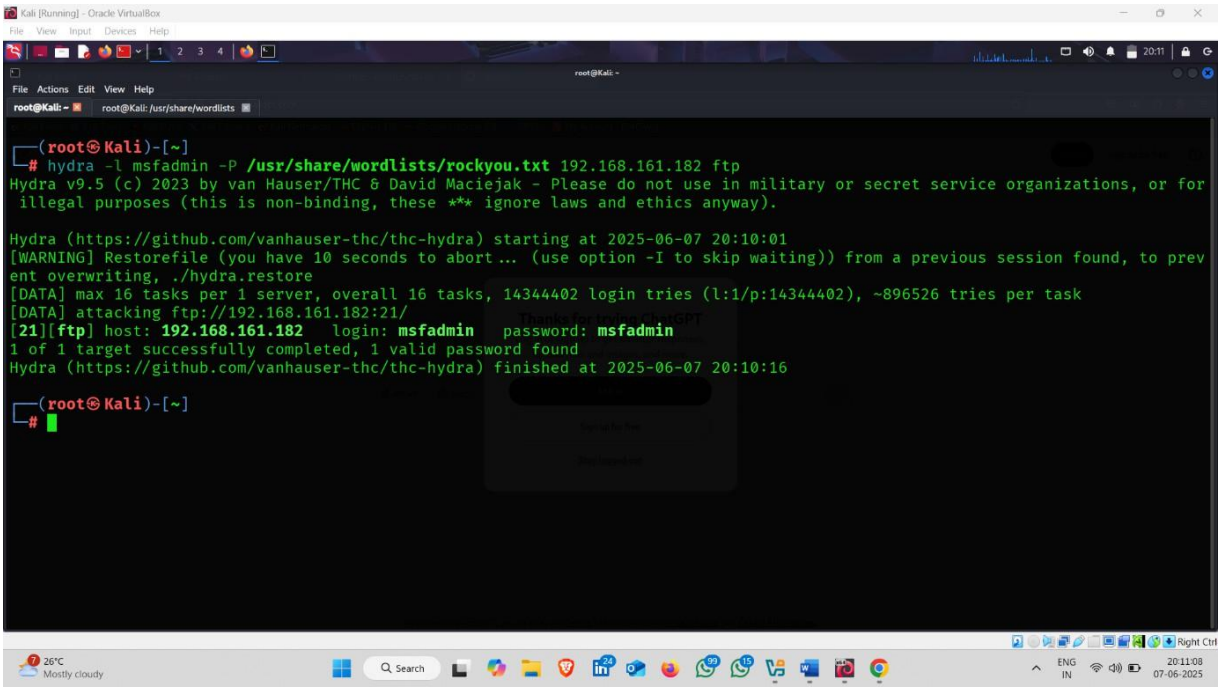
- **hydra**: starts the Hydra brute-force tool
- **-l msfadmin**: sets the username to "msfadmin"
- **-P /usr/share/wordlists/rockyou.txt:** uses the rockyou.txt wordlist for passwords
- **-F:** stops after finding the first valid login
- **192.168.161.182:** target IP address (Metasploitable2)
- **ftp:** the service to attack (FTP login)



- Attack Started



- Crack password



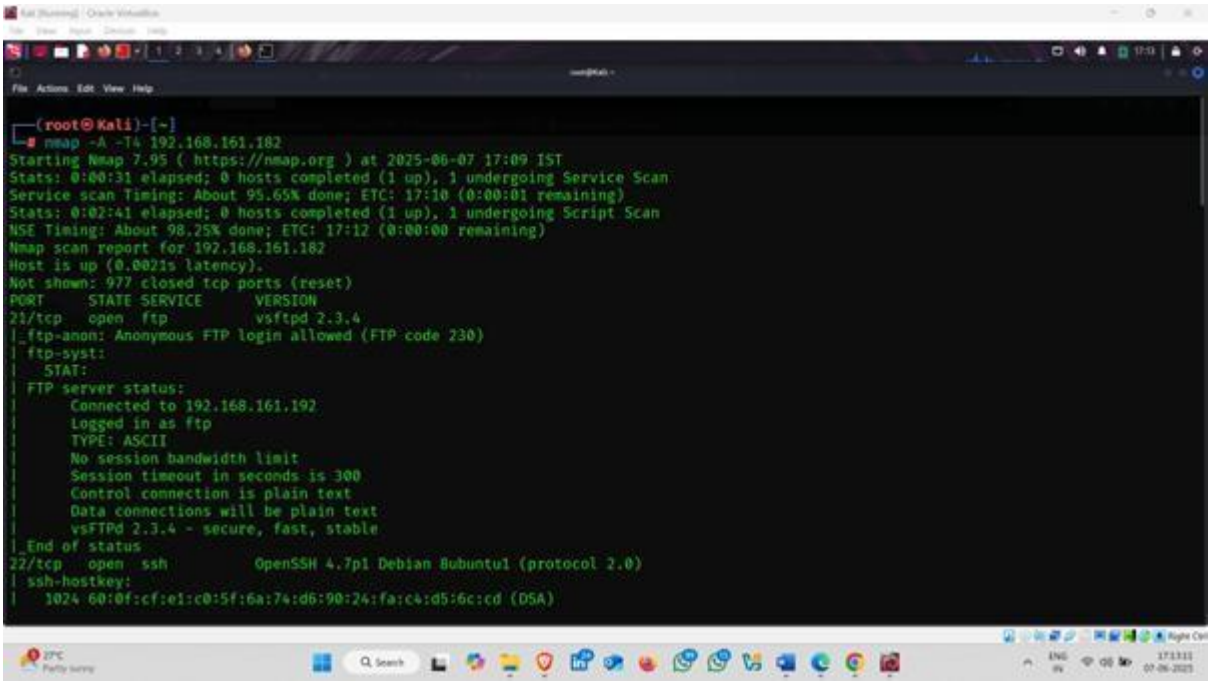
Anonymous login

Anonymous login is a type of access to a system (usually an **FTP server**) where **no username or password is required**, or a **default username like anonymous** is used with **any email or blank password**.

Note-: Our Target Are opened two ports that able to login anonymously i.e. port 21 and port 514

1. Using FTP Port (21) -:

- **As You can see below Scan image show that port number 21 on our target anonymous FTP login allowed . lets do it**

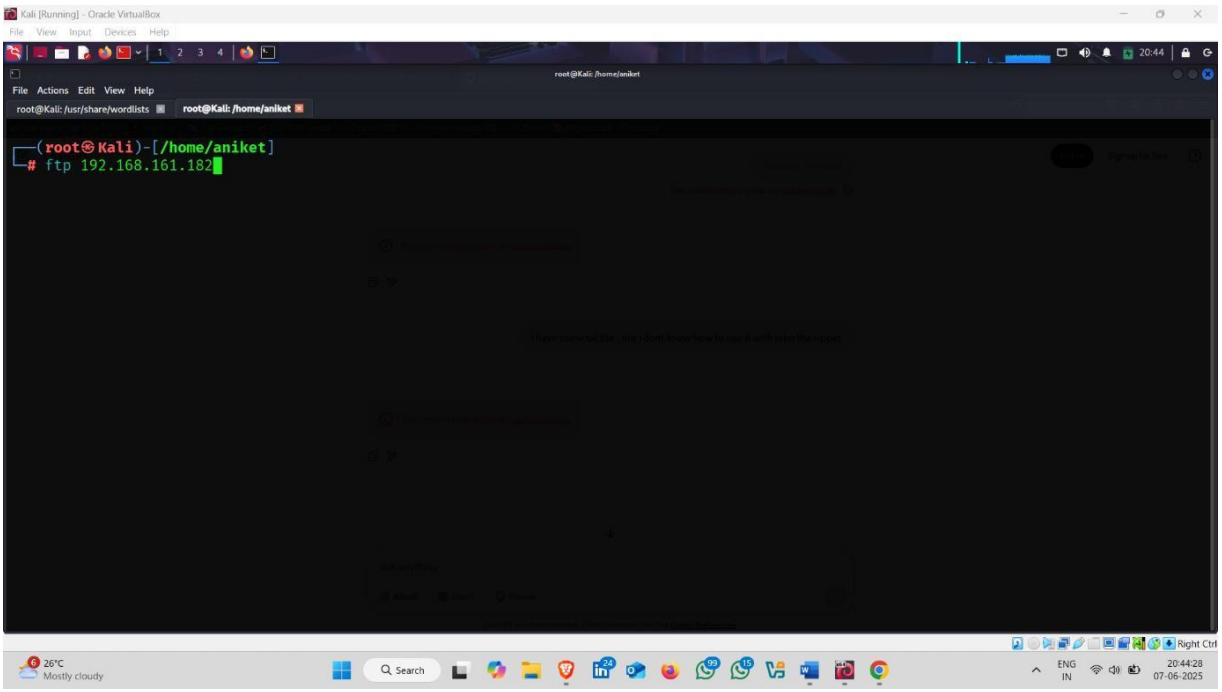


How to do it -:

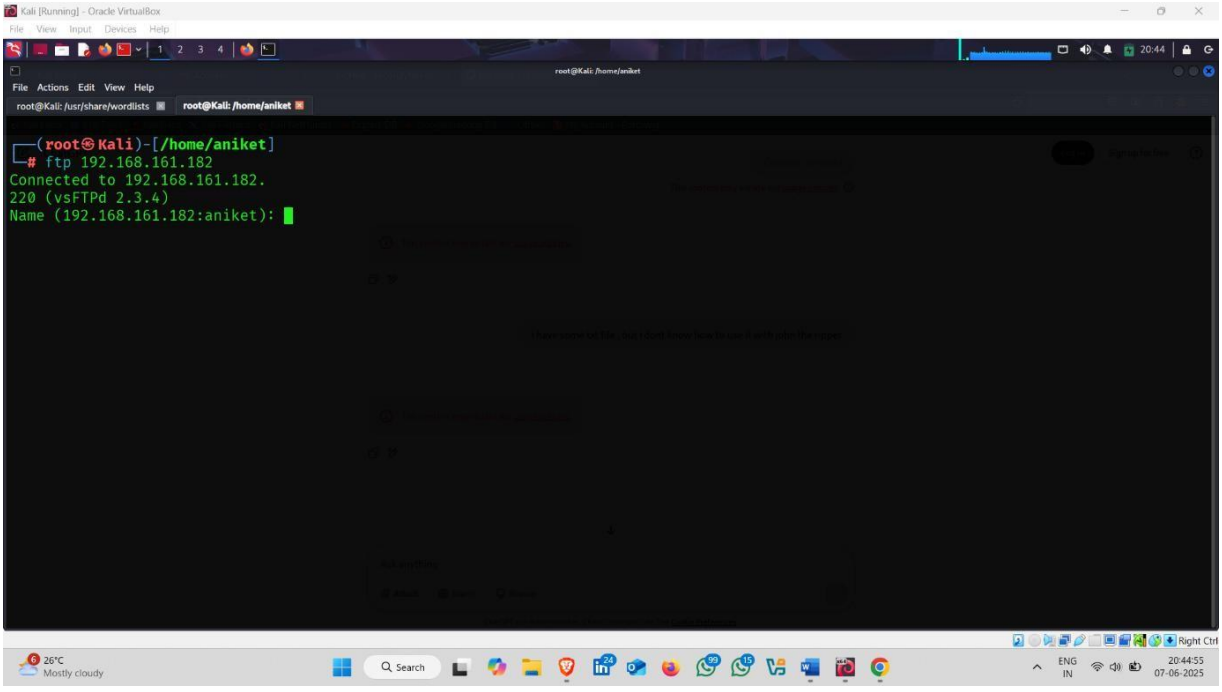
- Open kali linux terminal and type following command

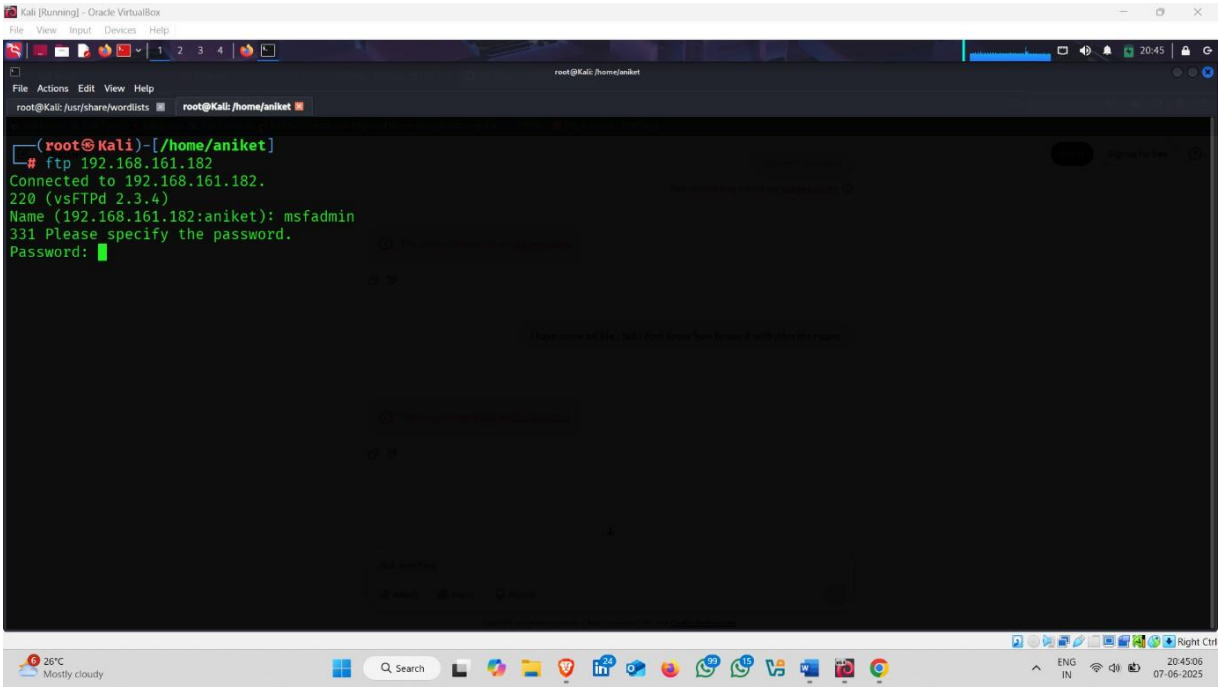
Command -: ftp 192.168.161.182

- Type command and hit enter button

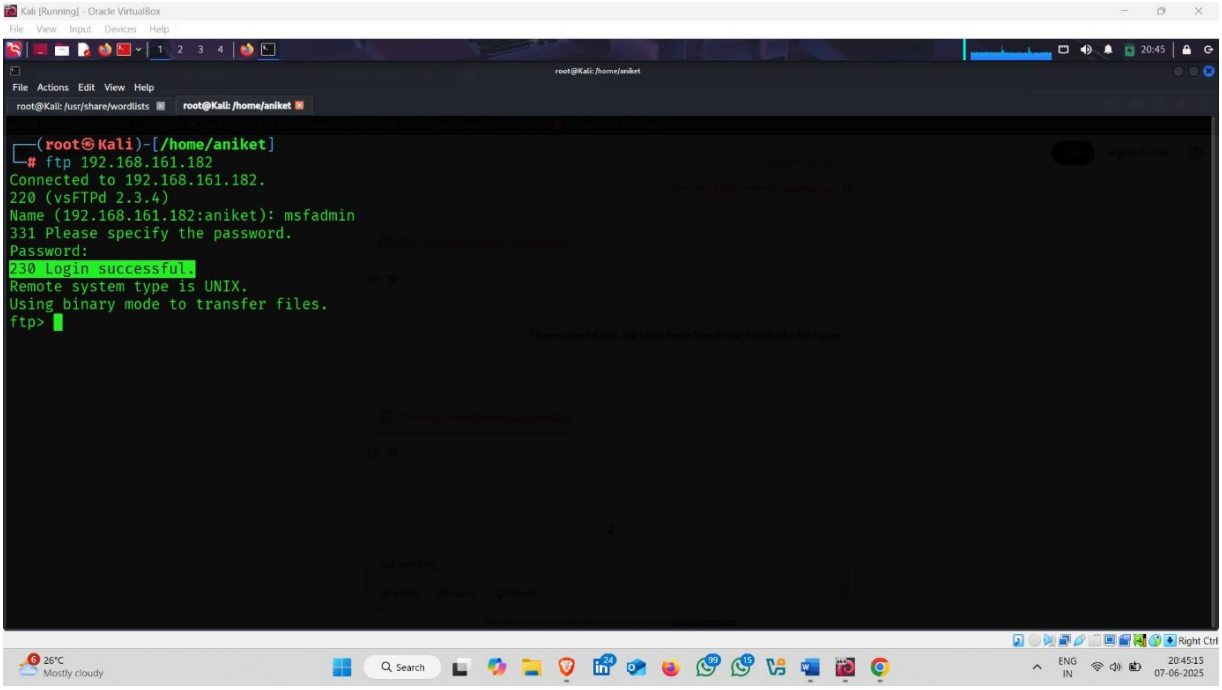


- Provide Username and password that you crack using hydra





- Login Successfully



2. RLogin (514)- remote shell-:

Rlogin (Remote Login) is a protocol used to log into another computer over a network, typically a UNIX system.

🟩 **Key Points:**

- **Port Number: 514** (TCP)
- **Service Name:** rlogin
- **Purpose:** Allows remote users to log in to a system and work as if they were physically present
- **Platform:** Mostly used on UNIX/Linux systems
- **Authentication:** Uses **.rhosts** file for user-based trust (no password sometimes)

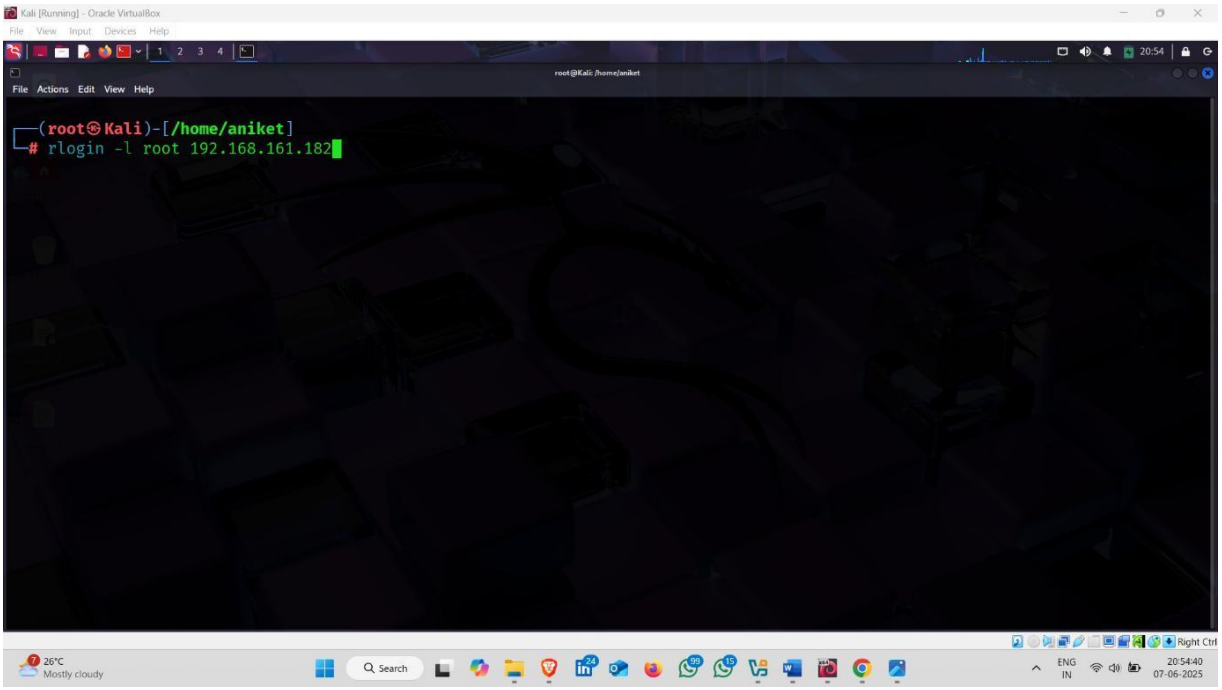
How to use it -:

- Open Kali linux terminal and type following command

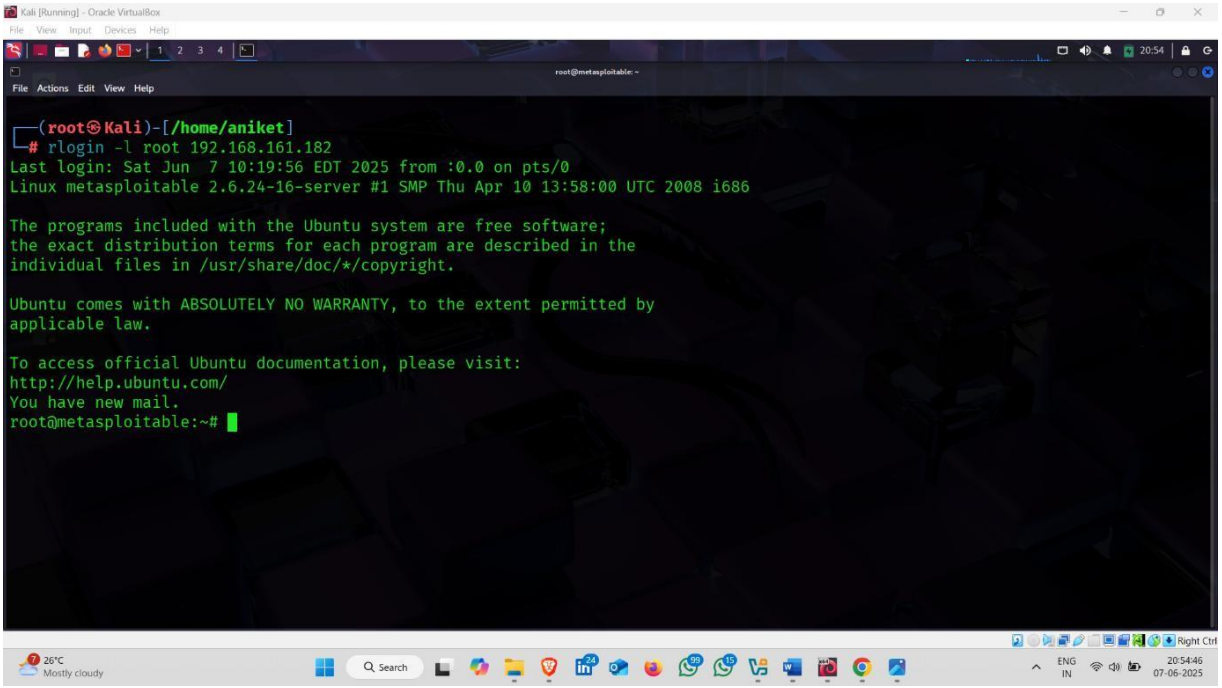
Command-: rlogin -l root 192.168.161.182

- rlogin: Starts the **Remote Login** client
- -l root: Specifies the **username** to log in as (in this case, root)
- 192.168.161.182: The **target IP address** (Metasploitable2 machine).

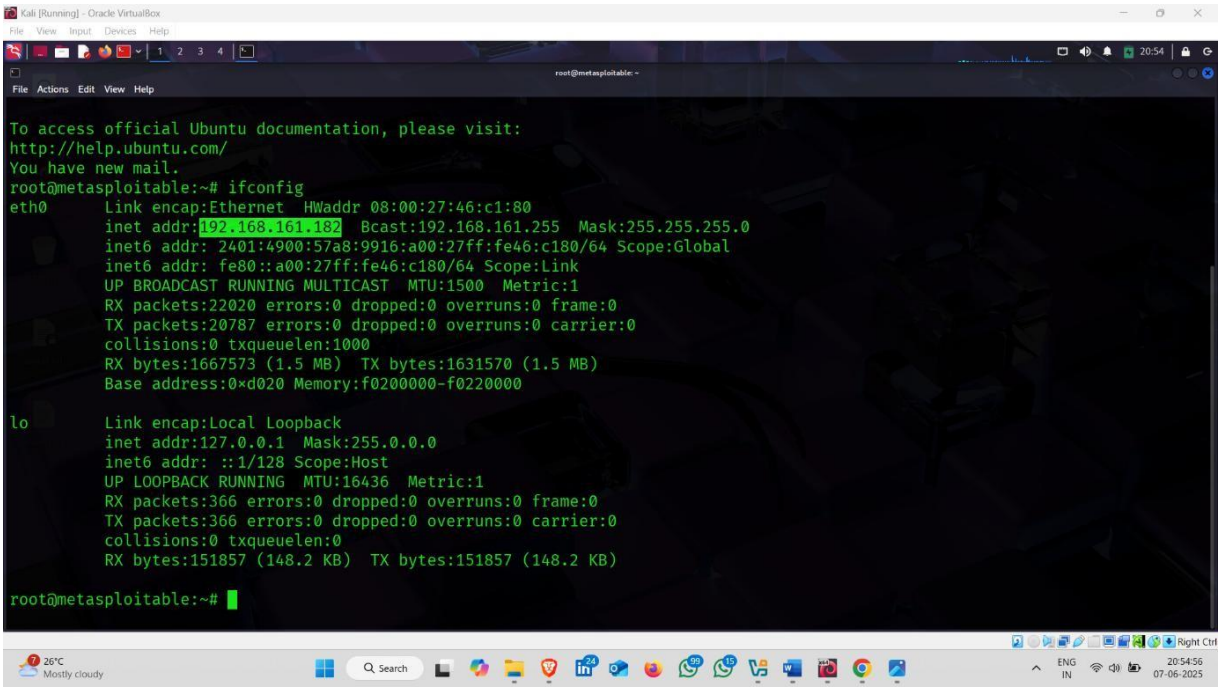
- Type this command and hit enter button



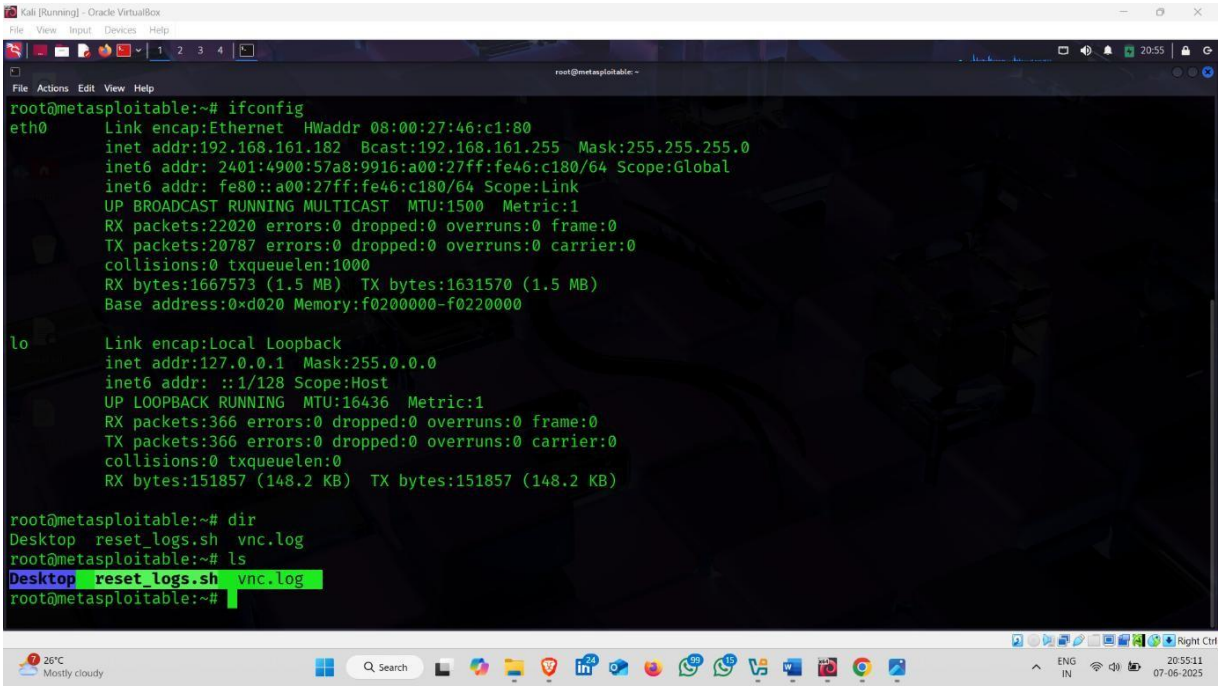
- **Login successfully** without username and password



- Target ip address



- Now you can access all these files and directories



Exploitation Using Metasploit

Metasploit is an open-source tool used for developing, testing, and executing **exploits** against systems to identify vulnerabilities. It helps security professionals simulate real-world attacks.

🟩 Why Metasploit is Used:

1. **Vulnerability Assessment** – Helps find security weaknesses in systems and networks.
2. **Exploit Development** – Used to create and test custom exploits.
3. **Payload Delivery** – Sends malicious code (payloads like Meterpreter) to gain remote access or control.
4. **Post-Exploitation** – Allows further actions after gaining access, such as privilege escalation, keylogging, file download/upload, etc.
5. **Security Testing** – Helps ethical hackers test the effectiveness of defenses.
6. **Learning & Training** – Commonly used in cybersecurity labs and training (e.g., Hack The Box, TryHackMe).

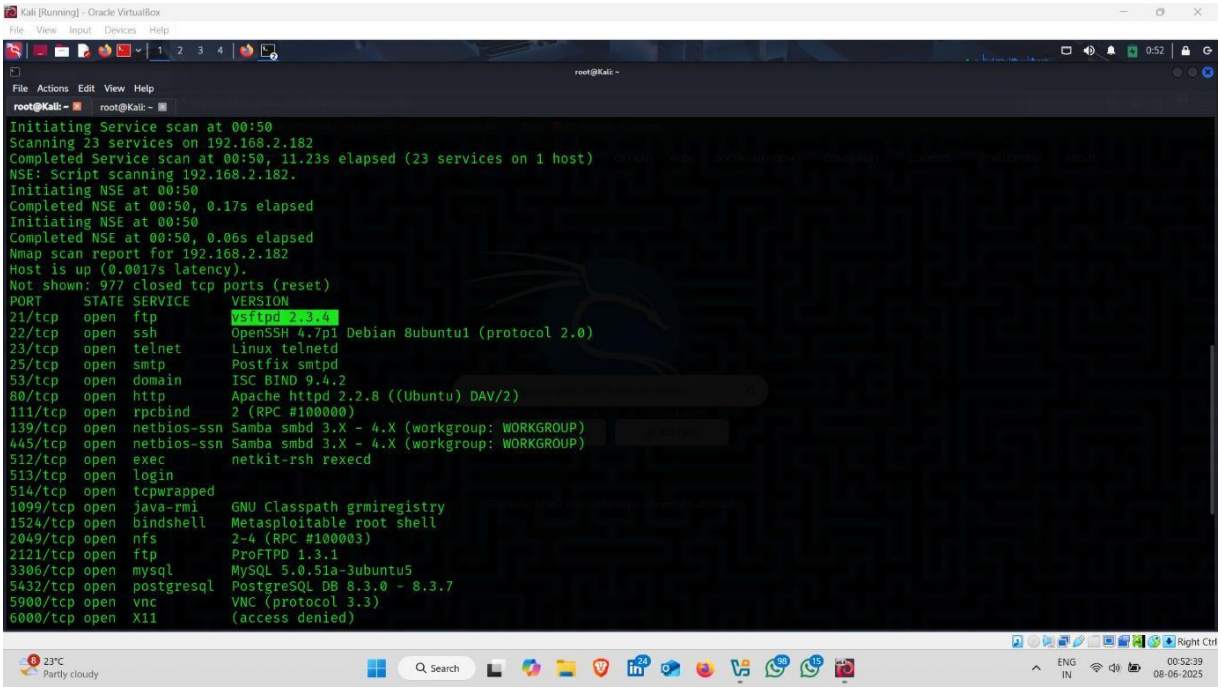
🟩 Key Components:

- **msfconsole** – Main command-line interface for using Metasploit.
- **Exploits** – Code that targets vulnerabilities.
- **Payloads** – Code that runs after the exploit (e.g., reverse shell).
- **Auxiliary Modules** – Scanners, fuzzers, and other tools.
- **Encoders** – Used to hide payloads from antivirus.

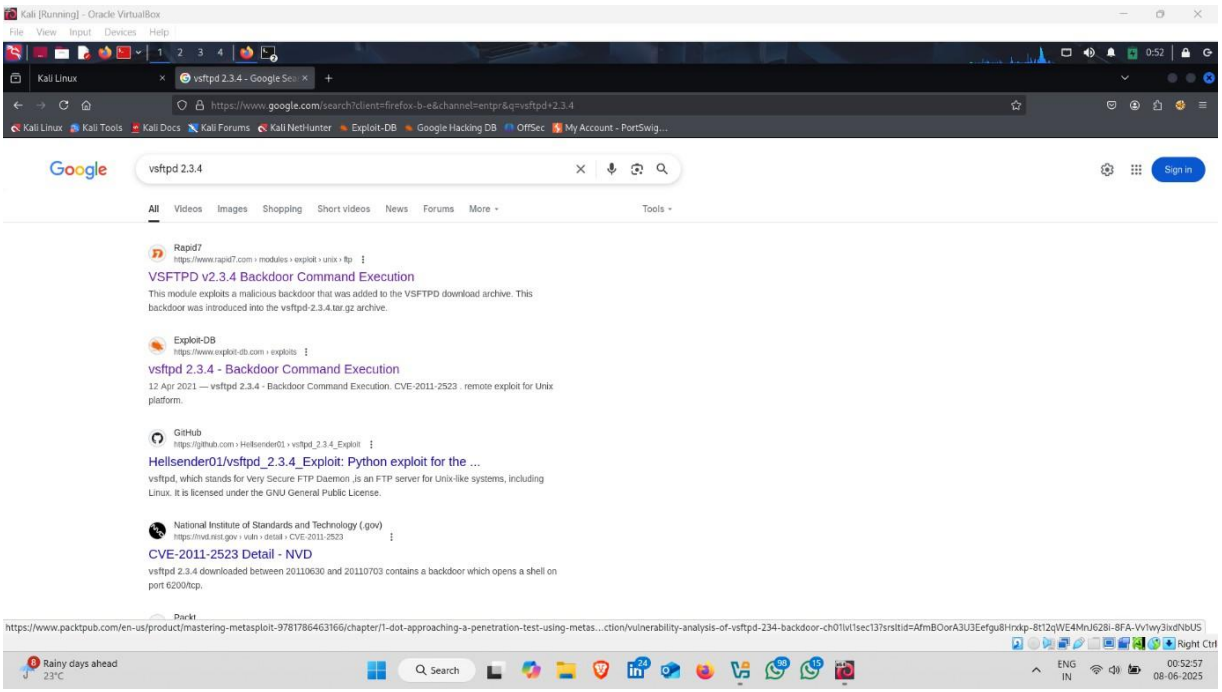
- **Listeners** – Wait for connections from compromised machines.

Before exploiting target , find which service version are vulnerable on our target

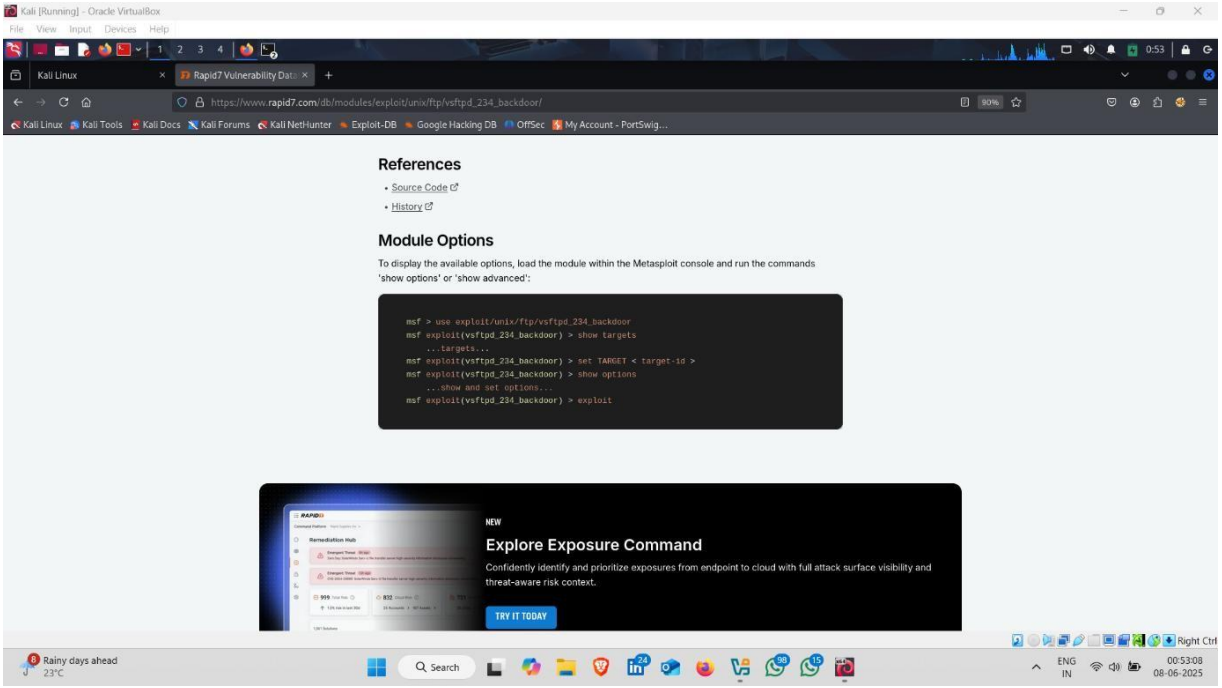
- Just copy the version of service



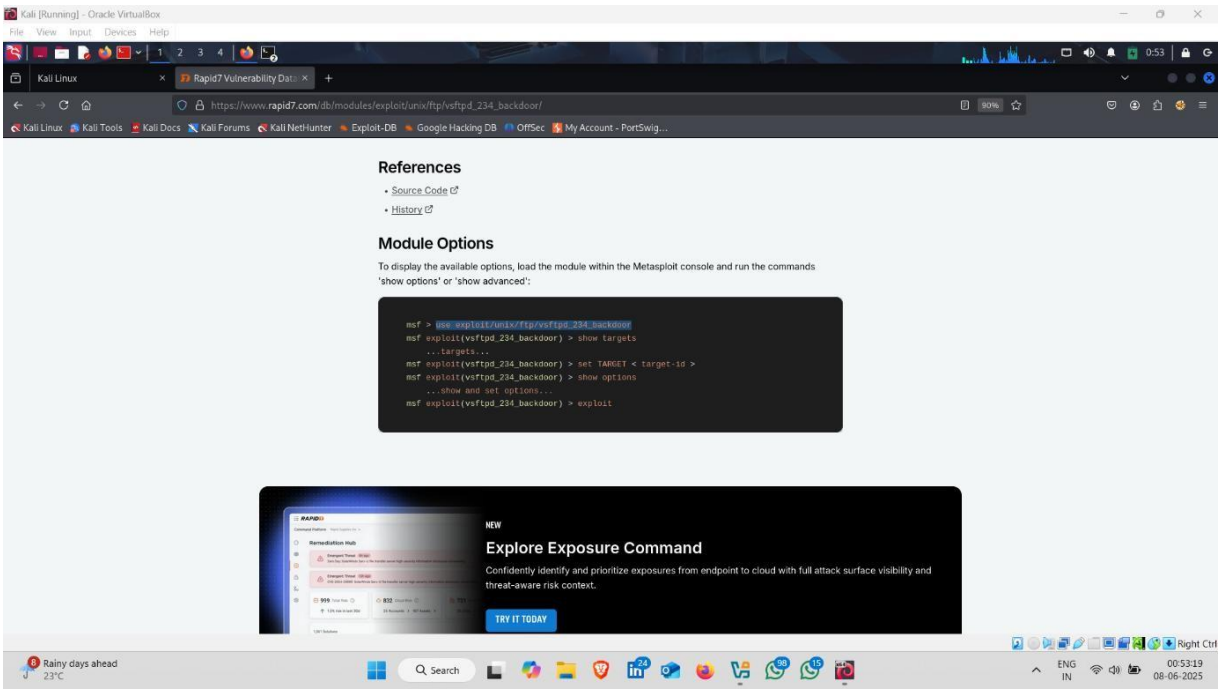
- And paste on google
- Click on first website --- rapid 7



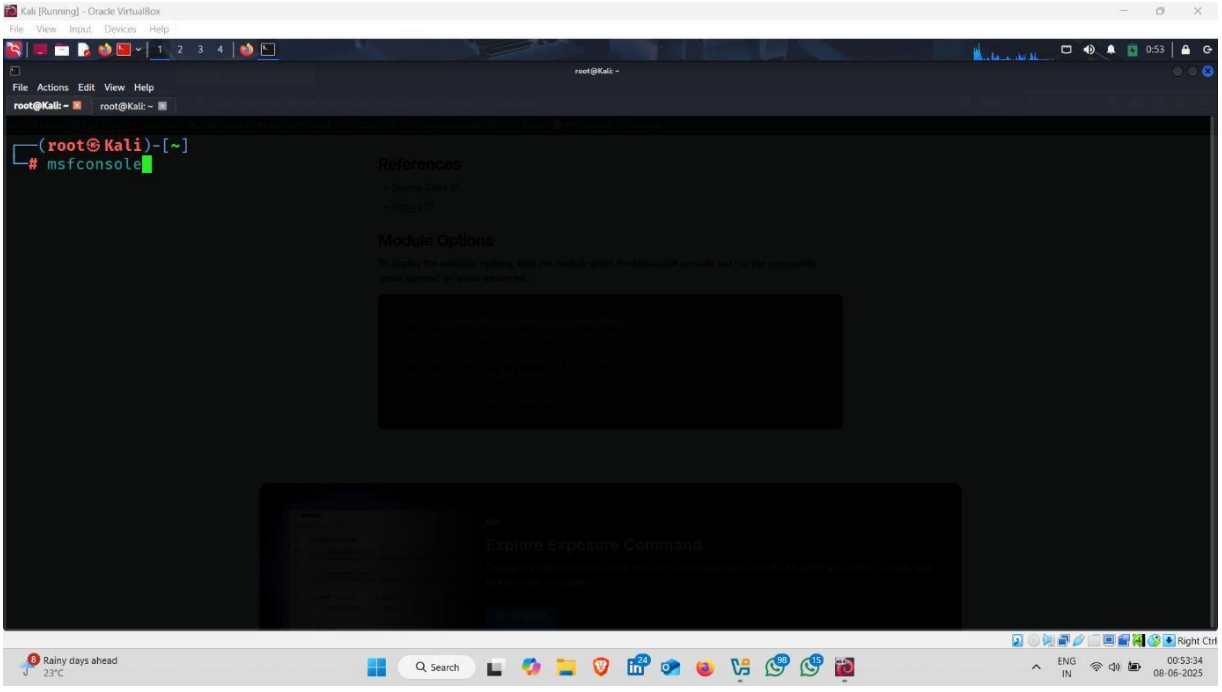
- here, the version is vulnerable



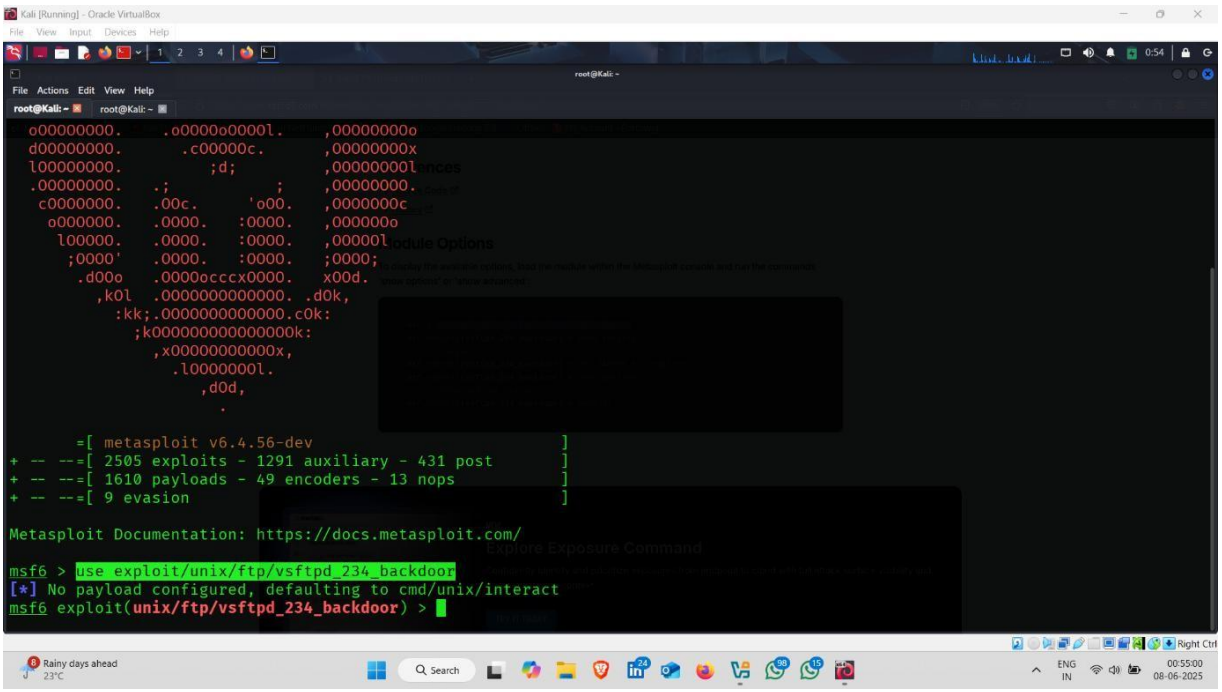
- Copy the exploit and open kali linux



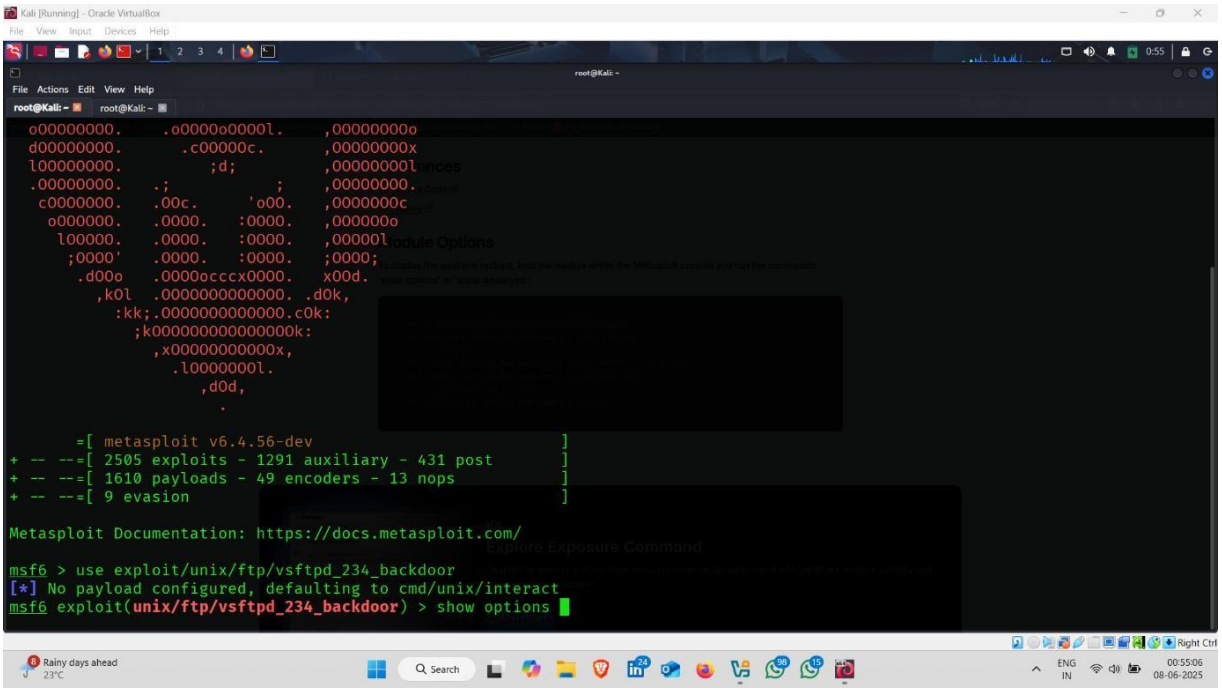
- Type **msfconsole**



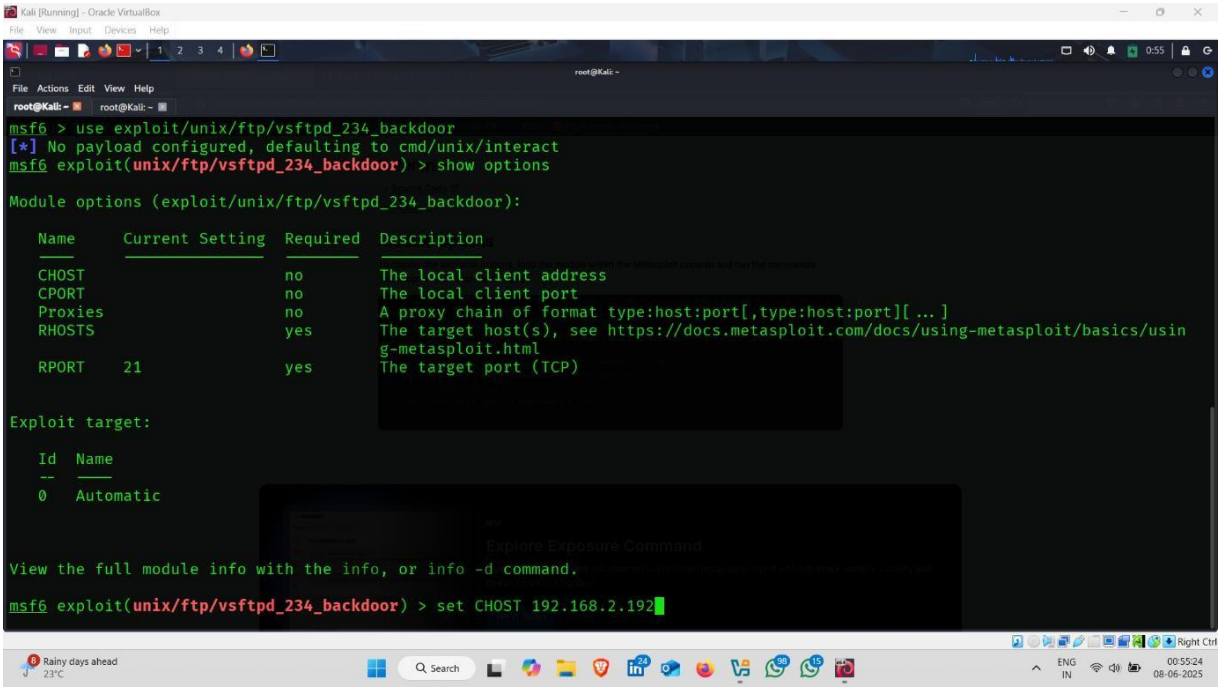
- And paste the exploit



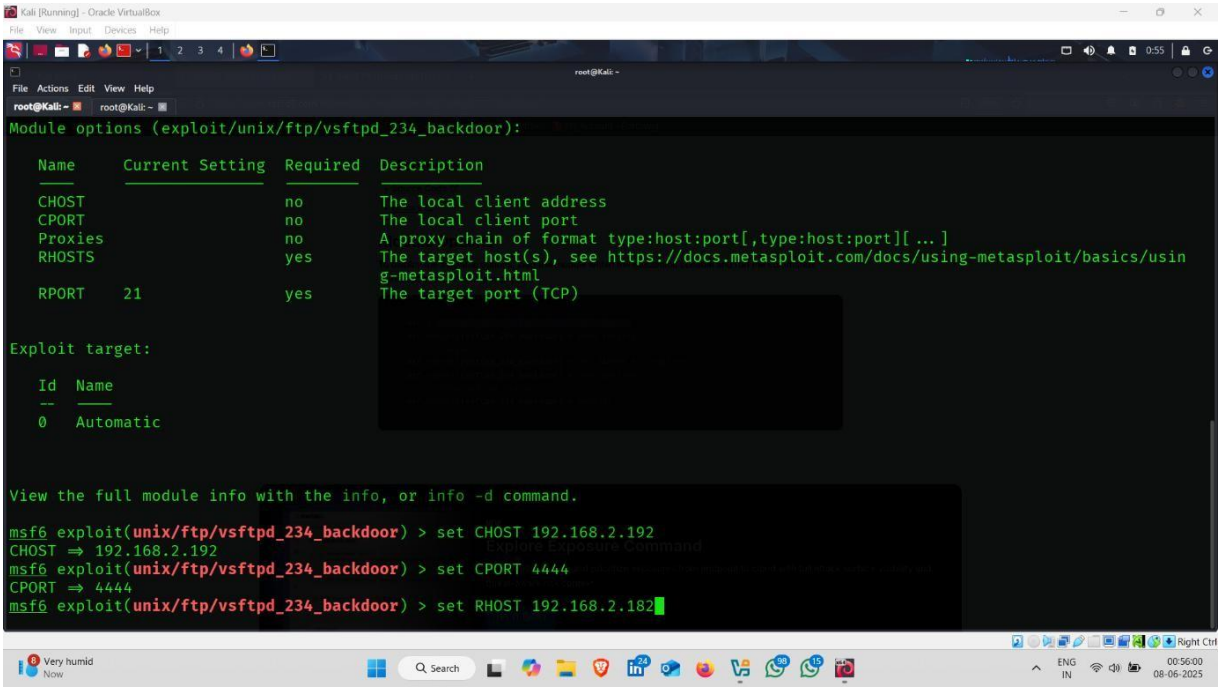
- Type show options



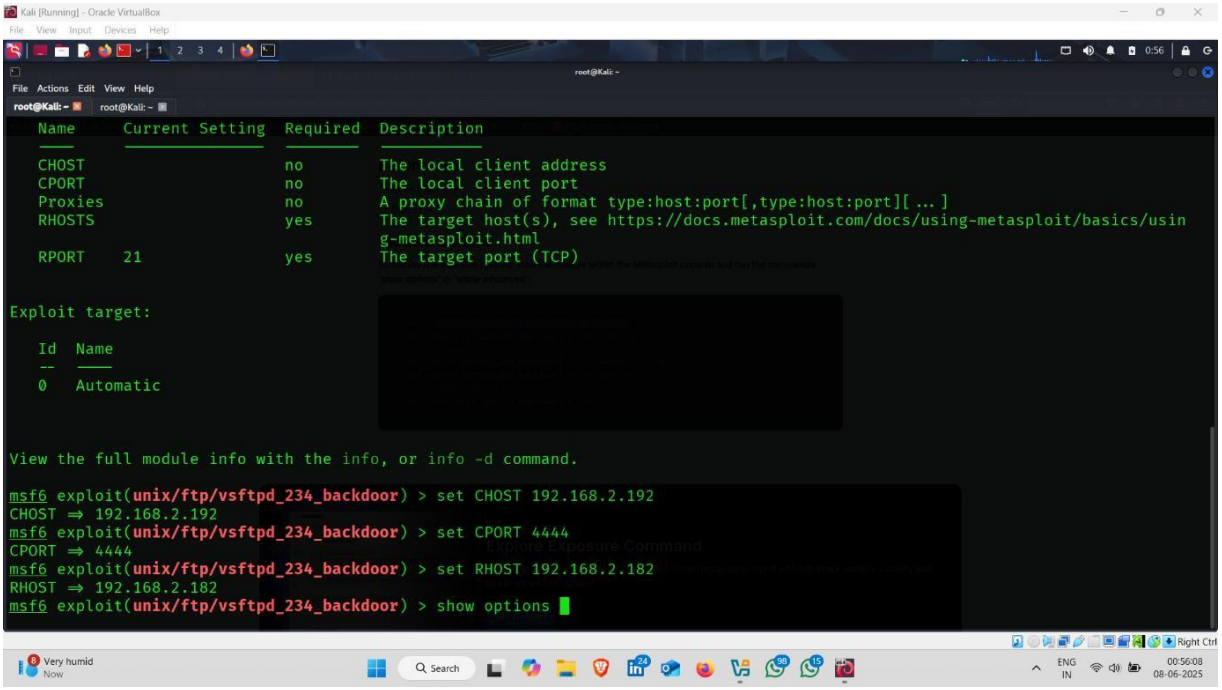
- Set all the requirements like CHOST , CPORT, RHOST
- CHOST -: kali linux Ip address
- RHOST-: target ip address



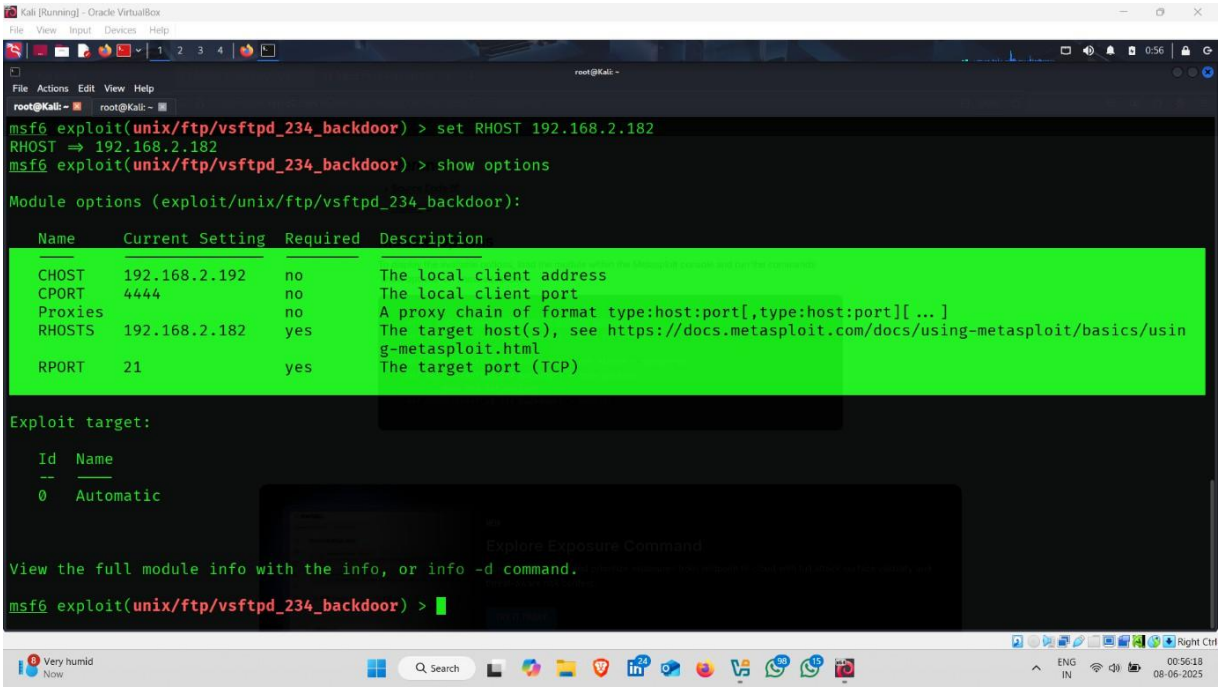
- All set



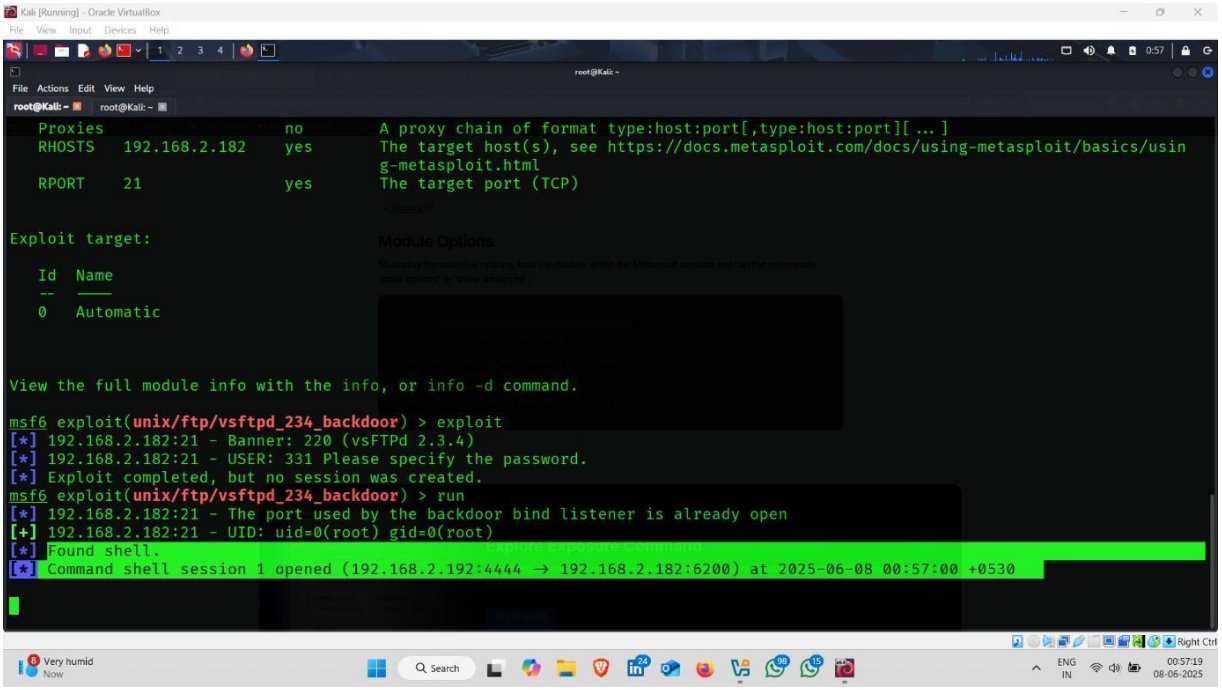
- Type show options to ensure that all ip are set or not



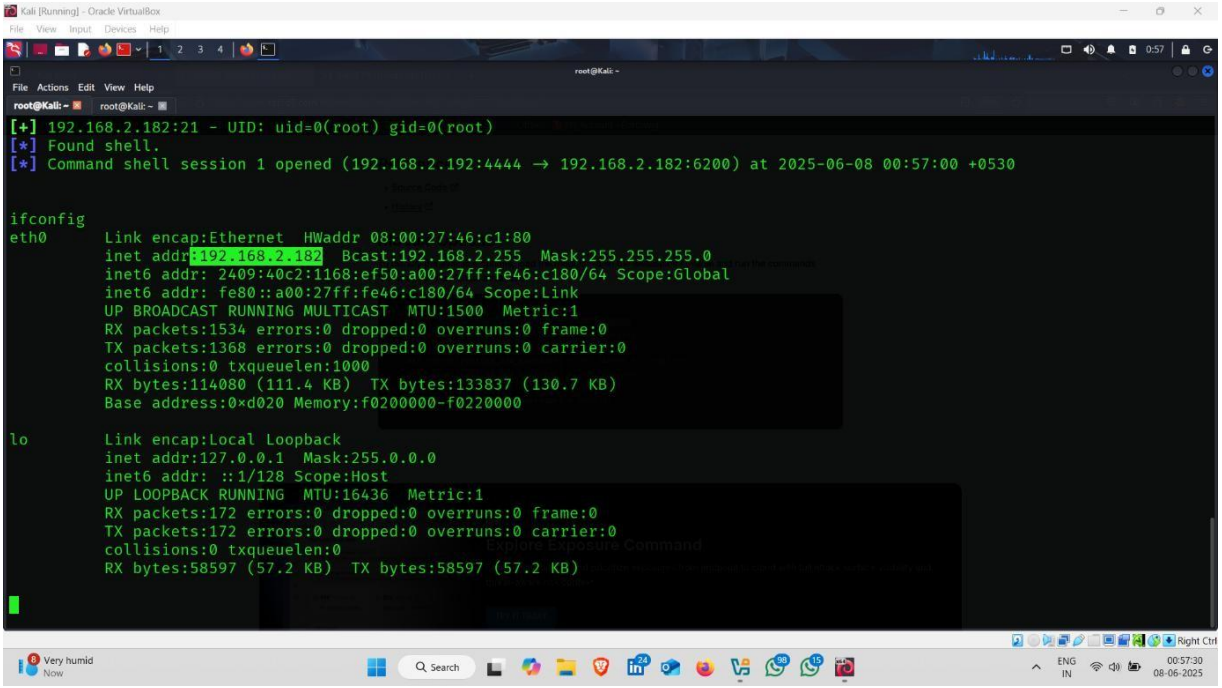
- **All done**



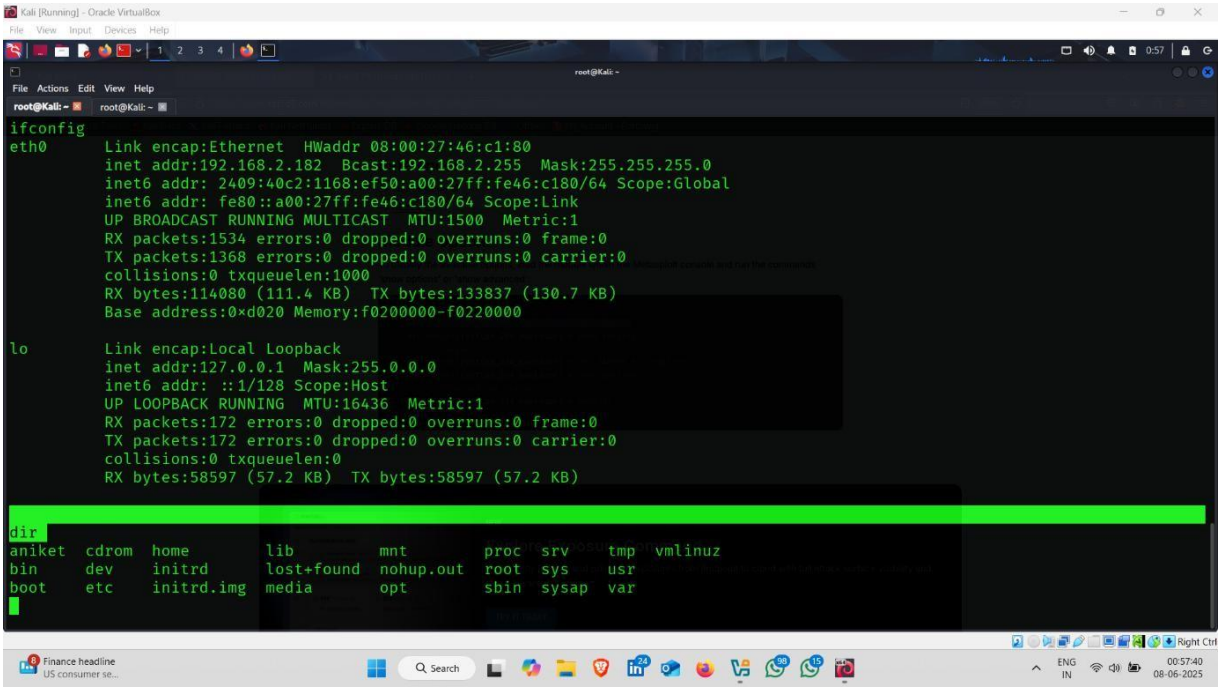
- **Type run**
- **Here , shell found , exploitation done**



- Target ip address



- **Target directories**


Kali [Running] - Oracle VirtualBox
File Actions Edit View Help
root@Kali: ~
ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:27:46:c1:80
inet addr:192.168.2.182 Bcast:192.168.2.255 Mask:255.255.255.0
inet6 addr: 2409:40c2:1168:ef50:a00:27ff:fe46:c180/64 Scope:Global
inet6 addr: fe80::a00:27ff:fe46:c180/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1534 errors:0 dropped:0 overruns:0 frame:0
TX packets:1368 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:114080 (111.4 KB) TX bytes:133837 (130.7 KB)
Base address:0×d020 Memory:f0200000-f0220000
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:172 errors:0 dropped:0 overruns:0 frame:0
TX packets:172 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:58597 (57.2 KB) TX bytes:58597 (57.2 KB)
dir
aniket cdrom home lib mnt proc srv tmp vmlinuz
bin dev initrd lost+found nohup.out root sys usr
boot etc initrd.img media opt sbin sysap var
Search
ENG
IN
08-06-2025

How to Defend Against Web Server Attack

1. **Keep Software Updated**
   - Regularly update the web server, CMS, frameworks, and plugins.
   - Patching fixes known vulnerabilities that attackers can exploit.
2. **Use a Web Application Firewall (WAF)**
   - Filters and blocks malicious HTTP traffic.
   - Helps protect against common attacks like SQL injection, XSS, and DDoS.
3. **Secure Server Configuration**
   - Disable directory listing.
   - Remove default files, pages, and unnecessary services.
   - Restrict access to sensitive files using permissions and authentication.
4. **Input Validation & Sanitization**
   - Validate all user inputs on the server side.
   - Use prepared statements and parameterized queries to prevent SQL injection.
5. **Implement HTTPS (SSL/TLS)**
   - Encrypts data between client and server.
   - Prevents man-in-the-middle attacks and eavesdropping.
6. **Strong Authentication & Session Management**
   - Use strong passwords and multi-factor authentication.
   - Secure session cookies with HttpOnly and Secure flags.

- Implement session timeouts.

7. Regular Security Testing

- Perform vulnerability scans, penetration testing, and code reviews.
- Continuously monitor for security gaps.

8. Logging & Monitoring

- Enable detailed logging of web server access and errors.
- Monitor logs regularly to detect suspicious activity.

9. Backup & Recovery Plan

- Regular backups ensure quick recovery from attacks like ransomware or data breaches.
