PennState
College of Information
Sciences and Technology

Lab Assignment Report

COURSE:	IST 894
ASSIGNMENT:	LAB ASSIGNMENT REPORT 2
SUBMITTED BY:	BRANTLEY PRICE
DUE DATE:	JUNE 15TH, 2025
INSTRUCTOR:	DR. BARTOLACCI

Contents

General Content

Developing a firm grasp of the fundamental principles of network security and network reconnaissance is one of the most important things Information Technology (IT) professionals can do to ensure their networks are postured correctly. Reason Labs (2024) defines network reconnaissance as a process of identifying network hosts and understanding the architecture and systems, particularly to exploit vulnerabilities. This is what the three cyber range exercises covered in this report entail. The student gained hands-on experience in learning how to discover active systems, identify running services, and understand how those services could potentially expose their systems. A necessary first step in cyber operations is network reconnaissance to establish a baseline for the network. The three exercises within this report, Nmap and Hping, Applied Nmap, and Basic Scripting, allow the student to perform some of this reconnaissance.

Viewing the network from a hacker's mindset was a theme. If the student can see things from the attacker's perspective, they may be able to identify vulnerabilities before they are exploited. To achieve this, techniques such as "footprinting" were employed. While similar to reconnaissance, footprinting is really a smaller part of the larger reconnaissance process (EC-Council, 2025), during which the student maps unknown territory to understand the terrain. The student was also able to experience the early stages of vulnerability identification, and in one example, identified outdated software that posed a risk to the system.

In addition to using tools like Nmap and Hping, students are introduced to scripting for automation. This hands-on experience helps them grasp some of the fundamental logic behind network communication. This opens up a world of possibilities that go beyond the provided network tools. By creating custom utilities tailored to their specific needs, students can gain insights and control over their networks that generic tools simply cannot provide. The skills developed in these labs are valuable across the IT field, and the combination of network reconnaissance, vulnerability assessment, and coding fundamentals lays a strong foundation for anyone wanting to enhance their knowledge of network security.

Technical Content

Nmap and Hping

The first exercise, Nmap and Hping, focused on network reconnaissance and advanced scanning methodologies. The student used Nmap to identify live hosts, enumerate open TCP/UDP ports (Figure 1), and review service versions and operating systems running on the target system (Figure 2). For example, an Aggressive Nmap scan was used, with the -A option in place of -sV, to determine the service versions running on the system. Beyond the basic and aggressive scans, stealthier enumeration options like SYN scans were used as well (Figure 3). SYN scans are often considered stealthy because they are quick and never complete TCP connections, making them less likely to be logged, depending on the network configuration (Nmap.org, n.d.). Hping was introduced at the end of the exercise as a method for creating custom packets and combating the SYN scans mentioned above. Using Hping, you can flag specific types of packets, including SYN packets (Kumar, 2025) (Figure 4). Hping can also be used for crafting TCP packets and observing tcpdumps via TCP and UDP protocols (Figure 5, Figure 6).

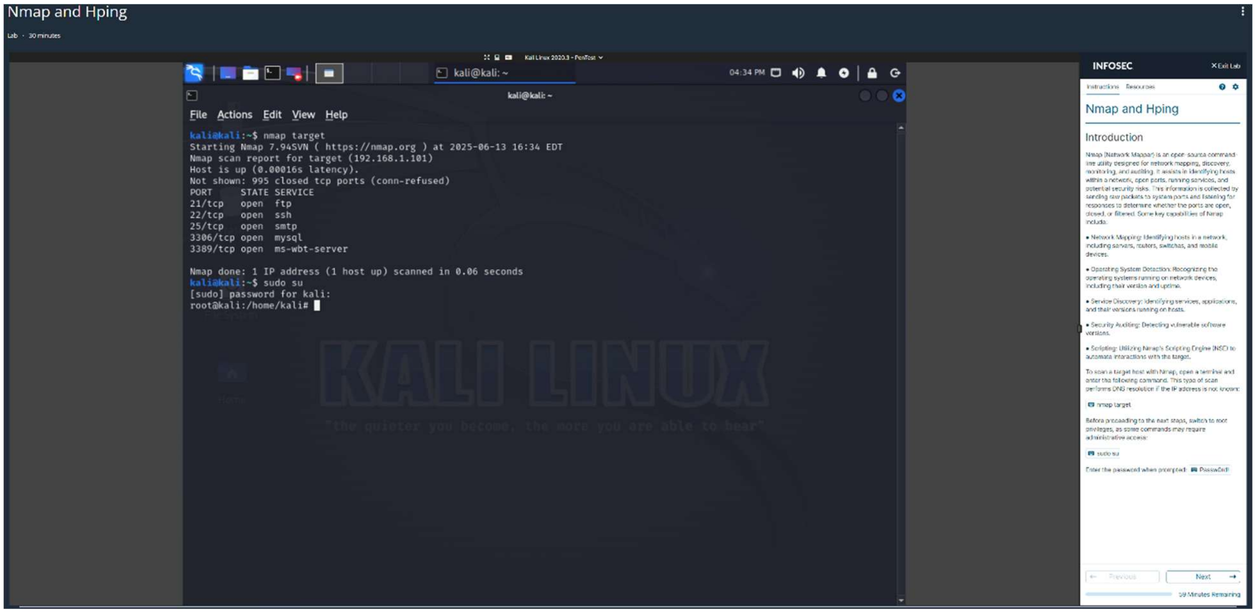
Applied Nmap

Taking some of the Nmap functionalities learned in the first exercise, the student moved to Applied Nmap for vulnerability identification and the identification of exploitation vectors. A bit shorter than the previous exercise, this focused the student on the practical use of information that can be gathered using Nmap. Once scans have been conducted to identify the versions of open services running on the system (Figure 7, Figure 8), the student was instructed as to how to map these versions to known vulnerabilities. For example, attack techniques like brute force credential attacks using Hydra and the exploitation of misconfigurations, such as anonymous FTP access (Figure 9) are executed. This exercise closes out with the connection reconnaissance data and the knowledge of specific exploits found in databases like the MITRE Common Vulnerabilities and Exposures (CVE) Database, such as the MySQL vulnerabilities in Figure 10.

Basic Scripting

Finally, Basic Scripting allowed the student to see how scripting can automate pre-built tools. Scripting is often advanced, but it is heavily used in modern cybersecurity workflows as it provides higher levels of fidelity and customization. The concepts covered in this exercise were presented using Python and guided the student in creating a Python part scanner. Using this scanner, the student was able to interact with network sockets, enumerate ports, and customize executable command-line arguments, all with Python code (Figure 11, Figure 12). Bash scripting techniques were also exercised. The Ping command was automated to demonstrate host reachability (Figure 13). Before automation could occur, chmod 744 is required to give the file owner execute permissions on the script. The combination of Python and Batch techniques in this exercise provides the student with a valuable glimpse into advanced techniques they can use to enhance their skill set.

Screenshots



```
kali@kali:~$ nmap target
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-13 16:34 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.00016s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
3306/tcp open  mysql
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
kali@kali:~$ sudo su
[sudo] password for kali:
root@kali:/home/kali# 
```

Nmap and Hping

Introduction

Nmap (Network Mapper) is an open-source command-line utility designed for network mapping, discovery, monitoring, and auditing. It assists in identifying hosts within a network, open ports, running services, and potential security risks. This information is collected by sending raw packets to system ports and listening for responses to determine whether the ports are open, closed, or filtered. Some key capabilities of Nmap include:

• Network Mapping: Identifying hosts in a network, including servers, routers, switches, and mobile devices.

• Operating System Detection: Recognizing the operating systems running on network devices, including their version and uptime.

• Service Discovery: Identifying services, applications, and their versions running on hosts.

• Security Auditing: Detecting vulnerable software versions.

• Scripting: Utilizing Nmap's Scripting Engine (NSE) to automate interactions with the target.

To scan a target host with Nmap, open a terminal and enter the following command. This type of scan performs DNS resolution if the IP address is not known:

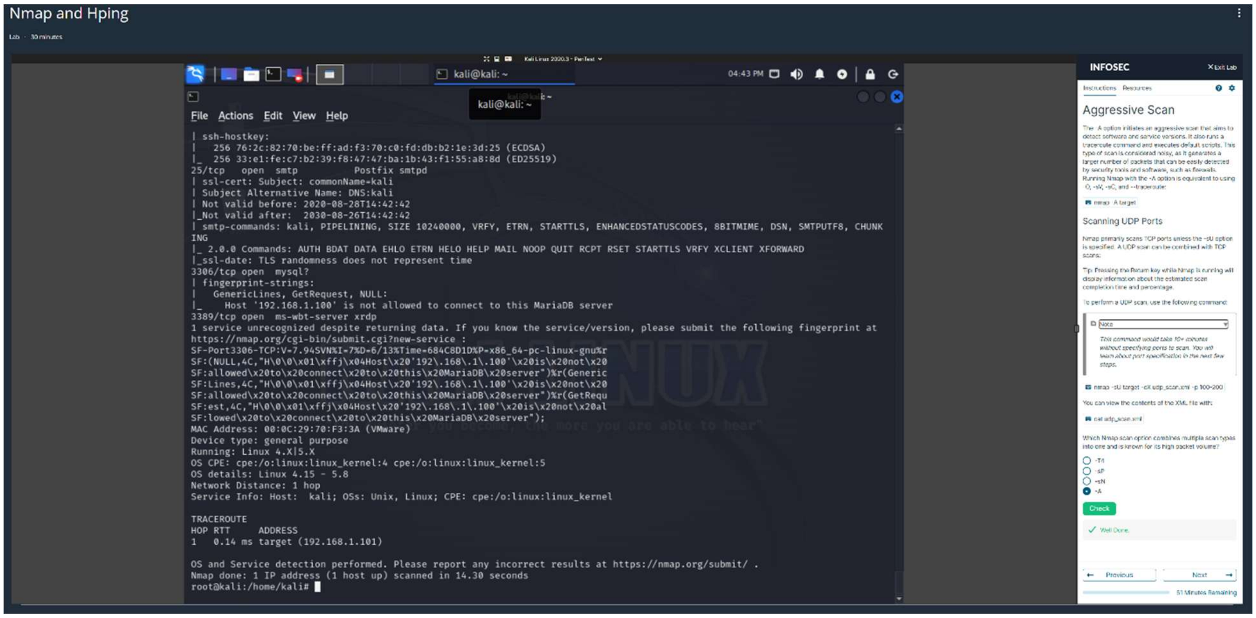
```
nmap target
```

Before proceeding to the next steps, switch to root privileges, as some commands may require administrative access:

```
sudo su
```

Enter the password when prompted: `Passw0rd!`

Figure 1: Nmap Target Scan



```
| ssh-hostkey:
|   256 76:2c:82:70:be:ff:ad:f3:70:c0:fd:db:b2:1e:3d:25 (ECDSA)
|_  256 33:e1:fe:c7:b2:39:f8:47:47:ba:1b:43:f1:55:a8:8d (ED25519)
25/tcp   open  smtp          Postfix smtpd
| ssl-cert: Subject: commonName=kali
| Subject Alternative Name: DNS:kali
| Not valid before: 2020-08-28T14:42:42
|_Not valid after:  2030-08-26T14:42:42
| smtp-commands: kali, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNK
ING
|_ 2.0.0 Commands: AUTH BDAT DATA EHLO ETRN HELO HELP MAIL NOOP QUIT RCPT RSET STARTTLS VRFY XCLIENT XFORWARD
|_ssl-date: TLS randomness does not represent time
3306/tcp open  mysql?
| fingerprint-strings:
|   GenericLines, GetRequest, NULL:
|_    Host '192.168.1.100' is not allowed to connect to this MariaDB server
3389/tcp open  ms-wbt-server xrdp
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3306-TCP:V=7.94SVN%I=7%D=6/13%Time=684C8D1D%P=x86_64-pc-linux-gnu%r
SF:(NULL,4C,"H\0\0\x01\xffj\x04Host\x20'192\.168\.1\.100'\x20is\x20not\x20
SF:allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server")%r(Generic
SF:Lines,4C,"H\0\0\x01\xffj\x04Host\x20'192\.168\.1\.100'\x20is\x20not\x20
SF:allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server")%r(GetRequ
SF:est,4C,"H\0\0\x01\xffj\x04Host\x20'192\.168\.1\.100'\x20is\x20not\x20al
SF:lowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server");
MAC Address: 00:0C:29:70:F3:3A (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: Host:  kali; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.14 ms target (192.168.1.101)

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 14.30 seconds
root@kali:/home/kali# 
```

Aggressive Scan

The -A option initiates an aggressive scan that aims to detect software and service versions. It also runs a traceroute command and executes default scripts. This type of scan is considered noisy, as it generates a larger number of packets that can be easily detected by security tools and software, such as firewalls. Running Nmap with the -A option is equivalent to using -O, -sV, -sC, and --traceroute:

```
nmap -A target
```

Scanning UDP Ports

Nmap primarily scans TCP ports unless the -sU option is specified. A UDP scan can be combined with TCP scans:

Tip: Pressing the Return key while Nmap is running will display information about the estimated scan completion time and percentage.

To perform a UDP scan, use the following command:

Note

This command would take 10+ minutes without specifying ports to scan. You will learn about port specification in the next few steps.

```
nmap -sU target -oX udp_scan.xml -p 100-200
```

You can view the contents of the XML file with:

```
cat udp_scan.xml
```

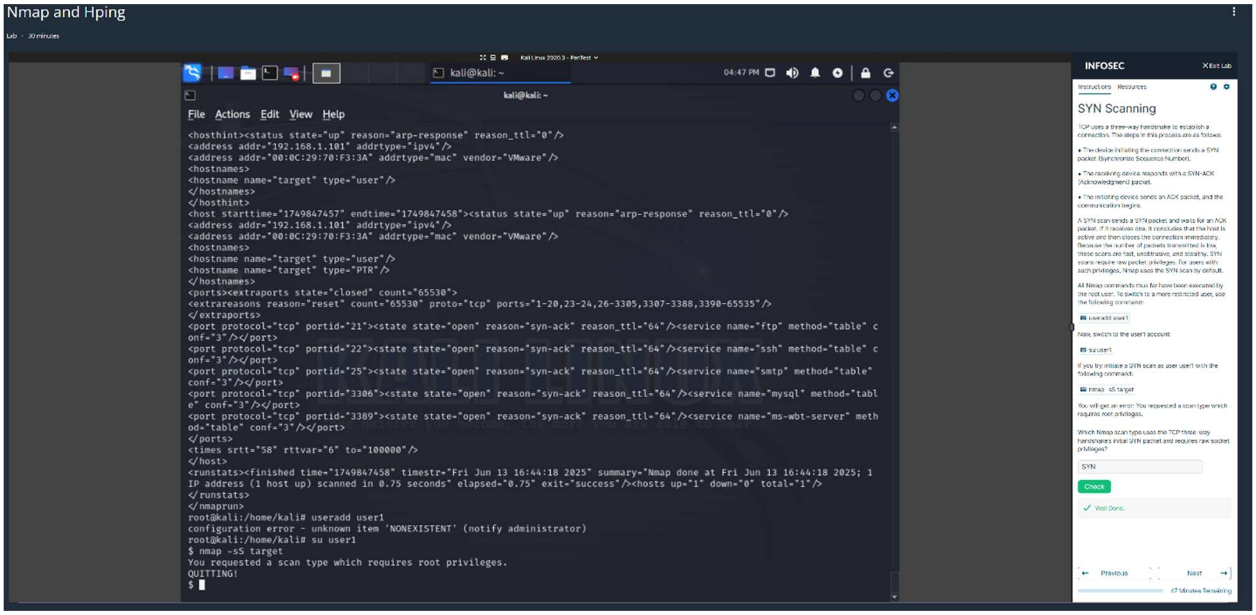
Which Nmap scan option combines multiple scan types into one and is known for its high packet volume?

○ -T4
○ -sP
○ -sN
● -A

Check

✓ Well Done.

Figure 2: Aggressive Scan for Service Versions



```
<hosthint><status state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.1.101" addrtype="ipv4"/>
<address addr="00:0C:29:70:F3:3A" addrtype="mac" vendor="VMware"/>
<hostnames>
<hostname name="target" type="user"/>
</hostnames>
</hosthint>
<host starttime="1749847457" endtime="1749847458"><status state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.1.101" addrtype="ipv4"/>
<address addr="00:0C:29:70:F3:3A" addrtype="mac" vendor="VMware"/>
<hostnames>
<hostname name="target" type="user"/>
<hostname name="target" type="PTR"/>
</hostnames>
<ports><extraports state="closed" count="65530">
<extrareasons reason="reset" count="65530" proto="tcp" ports="1-20,23-24,26-3305,3307-3388,3390-65535"/>
</extraports>
<port protocol="tcp" portid="21"><state state="open" reason="syn-ack" reason_ttl="64"/><service name="ftp" method="table" c
onf="3"/></port>
<port protocol="tcp" portid="22"><state state="open" reason="syn-ack" reason_ttl="64"/><service name="ssh" method="table" c
onf="3"/></port>
<port protocol="tcp" portid="25"><state state="open" reason="syn-ack" reason_ttl="64"/><service name="smtp" method="table"
conf="3"/></port>
<port protocol="tcp" portid="3306"><state state="open" reason="syn-ack" reason_ttl="64"/><service name="mysql" method="tabl
e" conf="3"/></port>
<port protocol="tcp" portid="3389"><state state="open" reason="syn-ack" reason_ttl="64"/><service name="ms-wbt-server" meth
od="table" conf="3"/></port>
</ports>
<times srtt="58" rttvar="6" to="100000"/>
</host>
<runstats><finished time="1749847458" timestr="Fri Jun 13 16:44:18 2025" summary="Nmap done at Fri Jun 13 16:44:18 2025; 1
IP address (1 host up) scanned in 0.75 seconds" elapsed="0.75" exit="success"/><hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>
root@kali:/home/kali# useradd user1
configuration error - unknown item 'NONEXISTENT' (notify administrator)
root@kali:/home/kali# su user1
$ nmap -sS target
You requested a scan type which requires root privileges.
QUITTING!
$ █
```

SYN Scanning

TCP uses a three-way handshake to establish a connection. The steps in this process are as follows:

- The device initiating the connection sends a SYN packet (Synchronize Sequence Number).
- The receiving device responds with a SYN-ACK (Acknowledgment) packet.
- The initiating device sends an ACK packet, and the communication begins.

A SYN scan sends a SYN packet and waits for an ACK packet. If it receives one, it concludes that the host is active and then closes the connection immediately. Because the number of packets transmitted is low, these scans are fast, unobtrusive, and stealthy. SYN scans require raw packet privileges. For users with such privileges, Nmap uses the SYN scan by default.

All Nmap commands thus far have been executed by the root user. To switch to a more restricted user, use the following command:

```
useradd user1
```

Now, switch to the user1 account:

```
su user1
```

If you try initiate a SYN scan as user user1 with the following command:

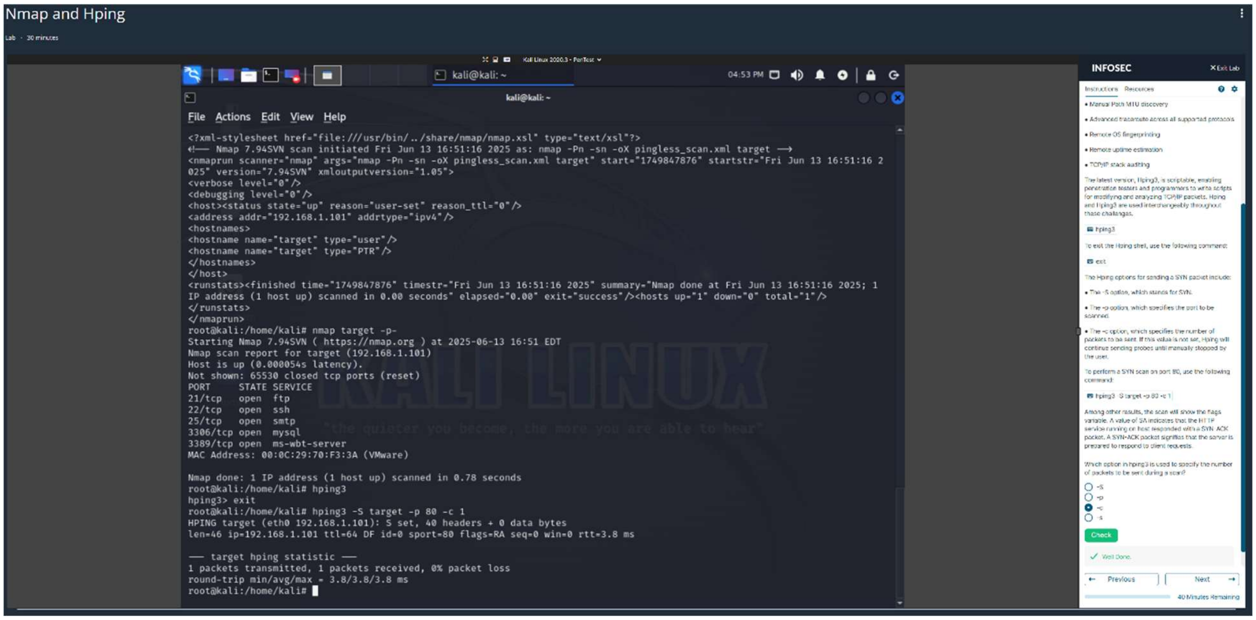
```
nmap -sS target
```

You will get an error: You requested a scan type which requires root privileges.

Which Nmap scan type uses the TCP three-way handshake's initial SYN packet and requires raw socket privileges?

SYN

Figure 3: SYN Scans



```
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.94SVN scan initiated Fri Jun 13 16:51:16 2025 as: nmap -Pn -sn -oX pingless_scan.xml target -->
<nmaprun scanner="nmap" args="nmap -Pn -sn -oX pingless_scan.xml target" start="1749847876" startstr="Fri Jun 13 16:51:16 2
025" version="7.94SVN" xmloutputversion="1.05">
<verbose level="0"/>
<debugging level="0"/>
<host><status state="up" reason="user-set" reason_ttl="0"/>
<address addr="192.168.1.101" addrtype="ipv4"/>
<hostnames>
<hostname name="target" type="user"/>
<hostname name="target" type="PTR"/>
</hostnames>
</host>
<runstats><finished time="1749847876" timestr="Fri Jun 13 16:51:16 2025" summary="Nmap done at Fri Jun 13 16:51:16 2025; 1
IP address (1 host up) scanned in 0.00 seconds" elapsed="0.00" exit="success"/><hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>
root@kali:/home/kali# nmap target -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-13 16:51 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.000054s latency).
Not shown: 65530 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
3306/tcp open  mysql
3389/tcp open  ms-wbt-server
MAC Address: 00:0C:29:70:F3:3A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.78 seconds
root@kali:/home/kali# hping3
hping3> exit
root@kali:/home/kali# hping3 -S target -p 80 -c 1
HPING target (eth0 192.168.1.101): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.101 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=3.8 ms

--- target hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.8/3.8/3.8 ms
root@kali:/home/kali# █
```

- Manual Path MTU discovery
- Advanced traceroute across all supported protocols
- Remote OS fingerprinting
- Remote uptime estimation
- TCP/IP stack auditing

The latest version, Hping3, is scriptable, enabling penetration testers and programmers to write scripts for modifying and analyzing TCP/IP packets. Hping and Hping3 are used interchangeably throughout these challenges.

```
hping3
```

To exit the Hping shell, use the following command:

```
exit
```

The Hping options for sending a SYN packet include:

- The -S option, which stands for SYN.
- The -p option, which specifies the port to be scanned.
- The -c option, which specifies the number of packets to be sent. If this value is not set, Hping will continue sending probes until manually stopped by the user.

To perform a SYN scan on port 80, use the following command:

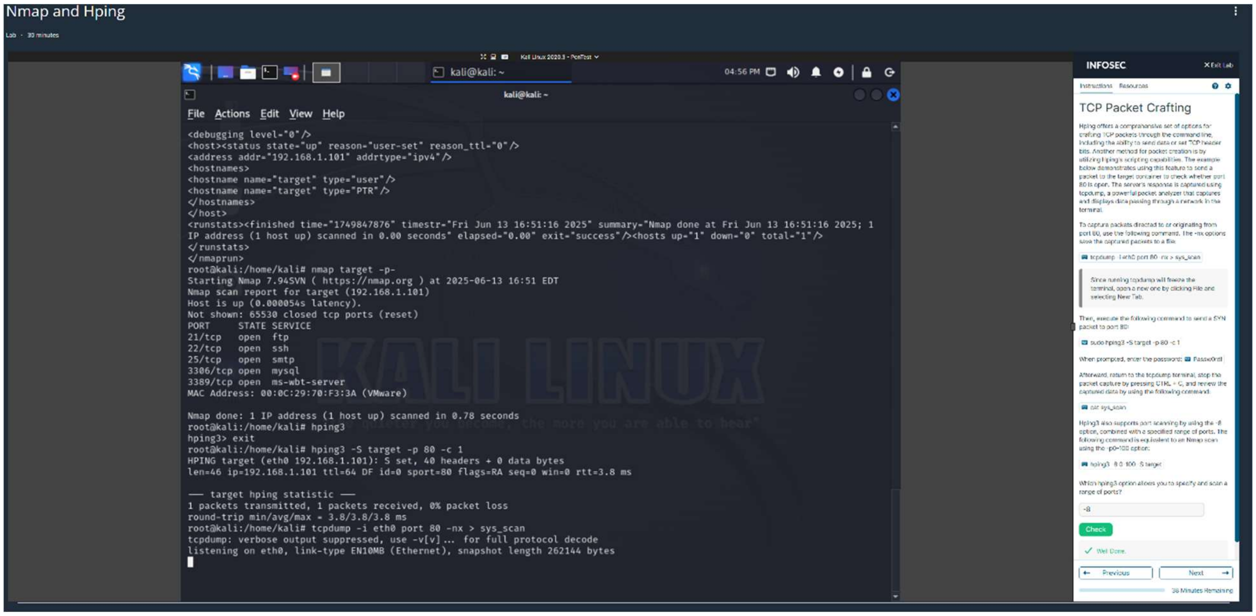
```
hping3 -S target -p 80 -c 1
```

Among other results, the scan will show the flags variable. A value of SA indicates that the HTTP service running on host responded with a SYN-ACK packet. A SYN-ACK packet signifies that the server is prepared to respond to client requests.

Which option in hping3 is used to specify the number of packets to be sent during a scan?

○ -S
○ -p
● -c
○ -s

Figure 4: Hping SYN Packet Scan



```
<debugging level="0"/>
<host><status state="up" reason="user-set" reason_ttl="0"/>
<address addr="192.168.1.101" addrtype="ipv4"/>
<hostnames>
<hostname name="target" type="user"/>
<hostname name="target" type="PTR"/>
</hostnames>
</host>
<runstats><finished time="1749847876" timestr="Fri Jun 13 16:51:16 2025" summary="Nmap done at Fri Jun 13 16:51:16 2025; 1
IP address (1 host up) scanned in 0.00 seconds" elapsed="0.00" exit="success"/><hosts up="1" down="0" total="1"/>
</runstats>
</nmaprun>
root@kali:/home/kali# nmap target -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-13 16:51 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.000054s latency).
Not shown: 65530 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
3306/tcp open  mysql
3389/tcp open  ms-wbt-server
MAC Address: 00:0C:29:70:F3:3A (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.78 seconds
root@kali:/home/kali# hping3
hping3> exit
root@kali:/home/kali# hping3 -S target -p 80 -c 1
HPING target (eth0 192.168.1.101): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.101 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=3.8 ms

--- target hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 3.8/3.8/3.8 ms
root@kali:/home/kali# tcpdump -i eth0 port 80 -nx > sys_scan
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
█
```

TCP Packet Crafting

Hping offers a comprehensive set of options for crafting TCP packets through the command line, including the ability to send data or set TCP header bits. Another method for packet creation is by utilizing Hping's scripting capabilities. The example below demonstrates using this feature to send a packet to the target container to check whether port 80 is open. The server's response is captured using tcpdump, a powerful packet analyzer that captures and displays data passing through a network in the terminal.

To capture packets directed to or originating from port 80, use the following command. The -nx options save the captured packets to a file:

`tcpdump -i eth0 port 80 -nx > sys_scan`

> Since running tcpdump will freeze the terminal, open a new one by clicking File and selecting New Tab.

Then, execute the following command to send a SYN packet to port 80:

`sudo hping3 -S target -p 80 -c 1`

When prompted, enter the password: `Passw0rd!`

Afterward, return to the tcpdump terminal, stop the packet capture by pressing CTRL + C, and review the captured data by using the following command:

`cat sys_scan`

Hping3 also supports port scanning by using the -8 option, combined with a specified range of ports. The following command is equivalent to an Nmap scan using the -p0-100 option:

`hping3 -8 0-100 -S target`

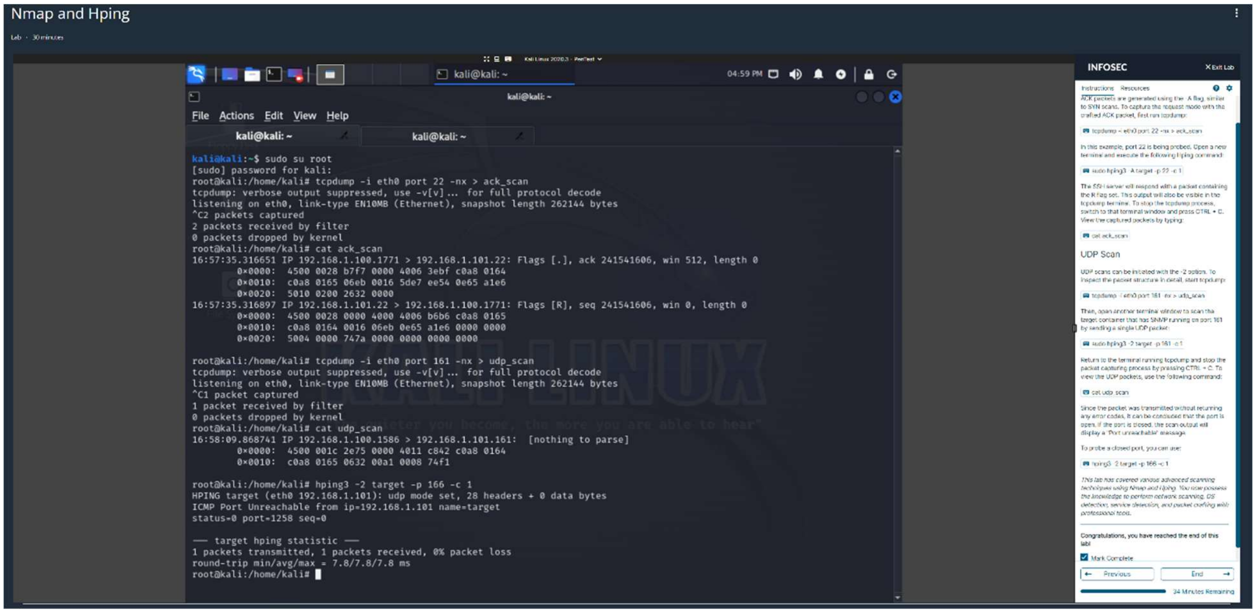
Which hping3 option allows you to specify and scan a range of ports?

-8

Check

✓ Well Done.

Figure 5: TCP Packet Crafting and tcpdump



```
kali@kali:~$ sudo su root
[sudo] password for kali:
root@kali:/home/kali# tcpdump -i eth0 port 22 -nx > ack_scan
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C2 packets captured
2 packets received by filter
0 packets dropped by kernel
root@kali:/home/kali# cat ack_scan
16:57:35.316651 IP 192.168.1.100.1771 > 192.168.1.101.22: Flags [.], ack 241541606, win 512, length 0
        0x0000:  4500 0028 b7f7 0000 4006 3ebf c0a8 0164
        0x0010:  c0a8 0165 06eb 0016 5de7 ee54 0e65 a1e6
        0x0020:  5010 0200 2632 0000
16:57:35.316897 IP 192.168.1.101.22 > 192.168.1.100.1771: Flags [R], seq 241541606, win 0, length 0
        0x0000:  4500 0028 0000 4000 4006 b6b6 c0a8 0165
        0x0010:  c0a8 0164 0016 06eb 0e65 a1e6 0000 0000
        0x0020:  5004 0000 747a 0000 0000 0000 0000

root@kali:/home/kali# tcpdump -i eth0 port 161 -nx > udp_scan
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C1 packet captured
1 packet received by filter
0 packets dropped by kernel
root@kali:/home/kali# cat udp_scan
16:58:09.868741 IP 192.168.1.100.1586 > 192.168.1.101.161:  [nothing to parse]
        0x0000:  4500 001c 2e75 0000 4011 c842 c0a8 0164
        0x0010:  c0a8 0165 0632 00a1 0008 74f1

root@kali:/home/kali# hping3 -2 target -p 166 -c 1
HPING target (eth0 192.168.1.101): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.1.101 name=target
status=0 port=1258 seq=0

--- target hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 7.8/7.8/7.8 ms
root@kali:/home/kali# █
```

ACK packets are generated using the -A flag, similar to SYN scans. To capture the request made with the crafted ACK packet, first run tcpdump:

`tcpdump -i eth0 port 22 -nx > ack_scan`

In this example, port 22 is being probed. Open a new terminal and execute the following Hping command:

`sudo hping3 -A target -p 22 -c 1`

The SSH server will respond with a packet containing the R flag set. This output will also be visible in the tcpdump terminal. To stop the tcpdump process, switch to that terminal window and press CTRL + C. View the captured packets by typing:

`cat ack_scan`

UDP Scan

UDP scans can be initiated with the -2 option. To inspect the packet structure in detail, start tcpdump:

`tcpdump -i eth0 port 161 -nx > udp_scan`

Then, open another terminal window to scan the target container that has SNMP running on port 161 by sending a single UDP packet:

`sudo hping3 -2 target -p 161 -c 1`

Return to the terminal running tcpdump and stop the packet capturing process by pressing CTRL + C. To view the UDP packets, use the following command:

`cat udp_scan`

Since the packet was transmitted without returning any error codes, it can be concluded that the port is open. If the port is closed, the scan output will display a 'Port unreachable' message.

To probe a closed port, you can use:

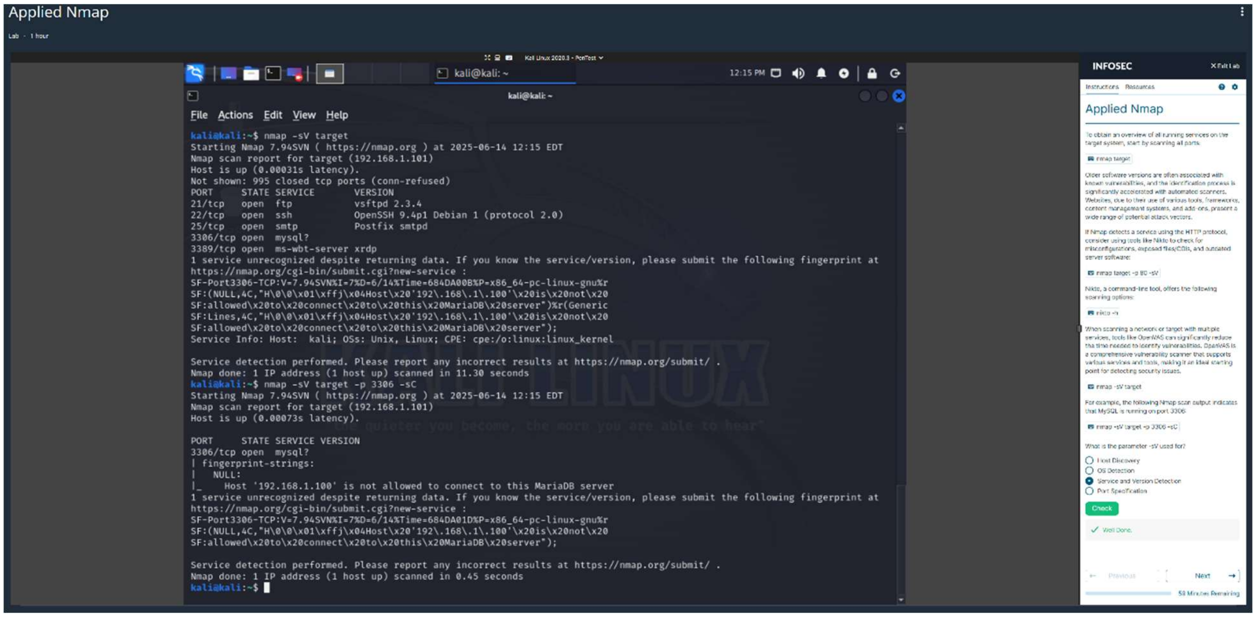
`hping3 -2 target -p 166 -c 1`

This lab has covered various advanced scanning techniques using Nmap and Hping. You now possess the knowledge to perform network scanning, OS detection, service detection, and packet crafting with professional tools.

Congratulations, you have reached the end of this lab!

☑ Mark Complete

Figure 6: UDP Scan and tcpdump



```
kali@kali:~$ nmap -sV target
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-14 12:15 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.00031s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           vsftpd 2.3.4
22/tcp   open  ssh           OpenSSH 9.4p1 Debian 1 (protocol 2.0)
25/tcp   open  smtp          Postfix smtpd
3306/tcp open  mysql?
3389/tcp open  ms-wbt-server xrdp
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3306-TCP:V=7.94SVN%I=7%D=6/14%Time=684DA00B%P=x86_64-pc-linux-gnu%r
SF:(NULL,4C,"H\0\0\x01\xffj\x04Host\x20'192\.168\.1\.100'\x20is\x20not\x20
SF:allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server")%r(Generic
SF:Lines,4C,"H\0\0\x01\xffj\x04Host\x20'192\.168\.1\.100'\x20is\x20not\x20
SF:allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server");
Service Info: Host:  kali; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.30 seconds
kali@kali:~$ nmap -sV target -p 3306 -sC
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-14 12:15 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.00073s latency).

PORT     STATE SERVICE VERSION
3306/tcp open  mysql?
| fingerprint-strings:
|   NULL:
|_    Host '192.168.1.100' is not allowed to connect to this MariaDB server
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3306-TCP:V=7.94SVN%I=7%D=6/14%Time=684DA01D%P=x86_64-pc-linux-gnu%r
SF:(NULL,4C,"H\0\0\x01\xffj\x04Host\x20'192\.168\.1\.100'\x20is\x20not\x20
SF:allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server");

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
kali@kali:~$
```

Applied Nmap

To obtain an overview of all running services on the target system, start by scanning all ports.

`nmap target`

Older software versions are often associated with known vulnerabilities, and the identification process is significantly accelerated with automated scanners. Websites, due to their use of various tools, frameworks, content management systems, and add-ons, present a wide range of potential attack vectors.

If Nmap detects a service using the HTTP protocol, consider using tools like Nikto to check for misconfigurations, exposed files/CGIs, and outdated server software:

`nmap target -p 80 -sV`

Nikto, a command-line tool, offers the following scanning options:

`nikto -h`

When scanning a network or target with multiple services, tools like OpenVAS can significantly reduce the time needed to identify vulnerabilities. OpenVAS is a comprehensive vulnerability scanner that supports various services and tools, making it an ideal starting point for detecting security issues.

`nmap -sV target`

For example, the following Nmap scan output indicates that MySQL is running on port 3306

`nmap -sV target -p 3306 -sC`

What is the parameter -sV used for?

○ Host Discovery
○ OS Detection
● Service and Version Detection
○ Port Specification

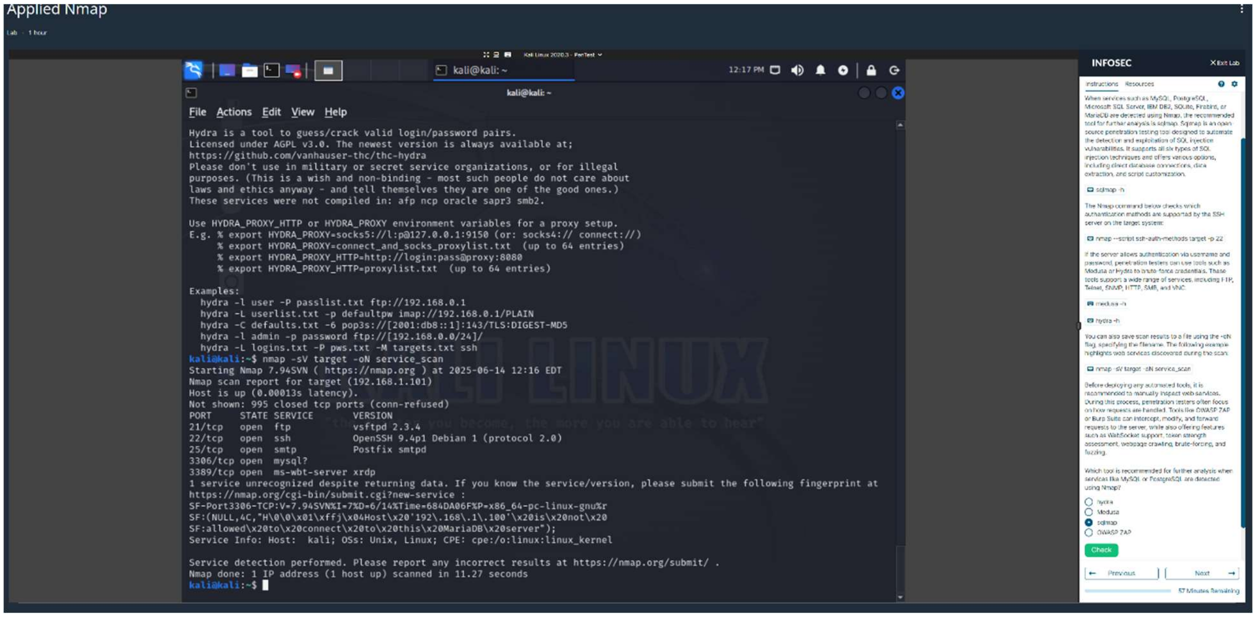
Check

✓ Well Done.

← Previous | Next →

58 Minutes Remaining

Figure 7: Nmap -sV Service Version



```
Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)
These services were not compiled in: afp ncp oracle sapr3 smb2.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY environment variables for a proxy setup.
E.g. % export HYDRA_PROXY=socks5://l:p@127.0.0.1:9150 (or: socks4:// connect://)
     % export HYDRA_PROXY=connect_and_socks_proxylist.txt  (up to 64 entries)
     % export HYDRA_PROXY_HTTP=http://login:pass@proxy:8080
     % export HYDRA_PROXY_HTTP=proxylist.txt  (up to 64 entries)

Examples:
  hydra -l user -P passlist.txt ftp://192.168.0.1
  hydra -L userlist.txt -p defaultpw imap://192.168.0.1/PLAIN
  hydra -C defaults.txt -6 pop3s://[2001:db8::1]:143/TLS:DIGEST-MD5
  hydra -l admin -p password ftp://[192.168.0.0/24]/
  hydra -L logins.txt -P pws.txt -M targets.txt ssh
kali@kali:~$ nmap -sV target -oN service_scan
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-14 12:16 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.00013s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           vsftpd 2.3.4
22/tcp   open  ssh           OpenSSH 9.4p1 Debian 1 (protocol 2.0)
25/tcp   open  smtp          Postfix smtpd
3306/tcp open  mysql?
3389/tcp open  ms-wbt-server xrdp
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port3306-TCP:V=7.94SVN%I=7%D=6/14%Time=684DA06F%P=x86_64-pc-linux-gnu%r
SF:(NULL,4C,"H\0\0\x01\xffj\x04Host\x20'192\.168\.1\.100'\x20is\x20not\x20
SF:allowed\x20to\x20connect\x20to\x20this\x20MariaDB\x20server");
Service Info: Host:  kali; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.27 seconds
kali@kali:~$
```

When services such as MySQL, PostgreSQL, Microsoft SQL Server, IBM DB2, SQLite, Firebird, or MariaDB are detected using Nmap, the recommended tool for further analysis is sqlmap. Sqlmap is an open source penetration testing tool designed to automate the detection and exploitation of SQL injection vulnerabilities. It supports all six types of SQL injection techniques and offers various options, including direct database connections, data extraction, and script customization.

`sqlmap -h`

The Nmap command below checks which authentication methods are supported by the SSH server on the target system:

`nmap --script ssh-auth-methods target -p 22`

If the server allows authentication via username and password, penetration testers can use tools such as Medusa or Hydra to brute-force credentials. These tools support a wide range of services, including FTP, Telnet, SNMP, HTTP, SMB, and VNC.

`medusa -h`

`hydra -h`

You can also save scan results to a file using the -oN flag, specifying the filename. The following example highlights web services discovered during the scan.

`nmap -sV target -oN service_scan`

Before deploying any automated tools, it is recommended to manually inspect web services. During this process, penetration testers often focus on how requests are handled. Tools like OWASP ZAP or Burp Suite can intercept, modify, and forward requests to the server, while also offering features such as WebSocket support, token strength assessment, webpage crawling, brute-forcing, and fuzzing.

Which tool is recommended for further analysis when services like MySQL or PostgreSQL are detected using Nmap?

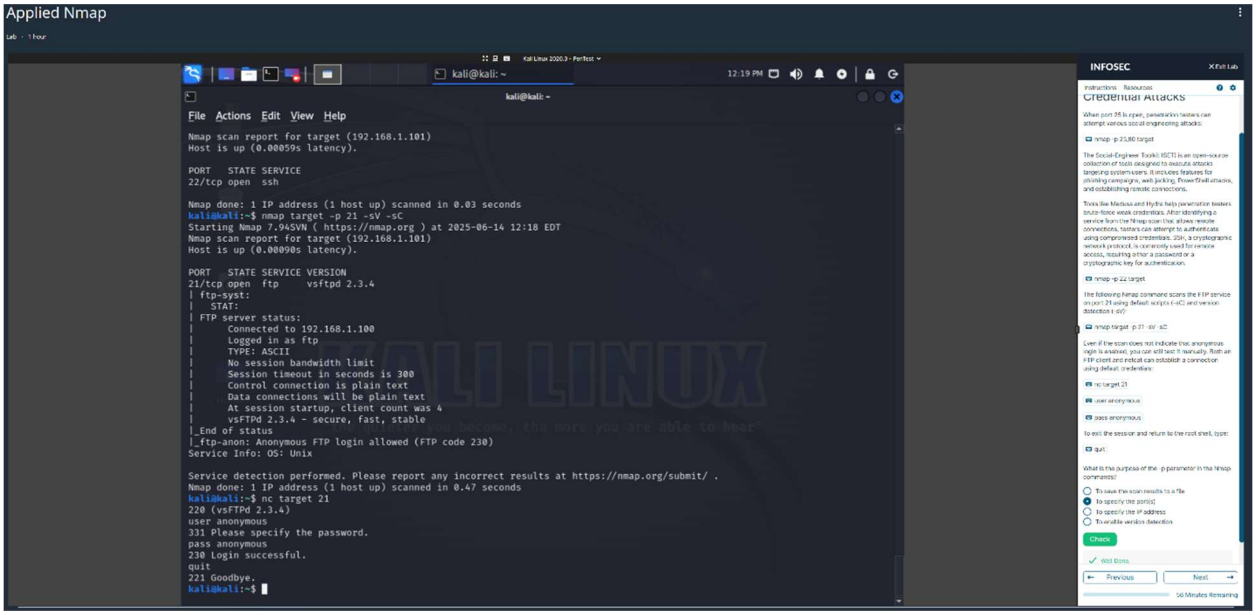
○ hydra
○ Medusa
● sqlmap
○ OWASP ZAP

Check

← Previous | Next →

57 Minutes Remaining

Figure 8: Further Service Version Identification



Applied Nmap

Lab · 1 hour

```
Nmap scan report for target (192.168.1.101)
Host is up (0.00059s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds
kali@kali:~$ nmap target -p 21 -sV -sC
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-14 12:18 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.00090s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.1.100
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.47 seconds
kali@kali:~$ nc target 21
220 (vsFTPd 2.3.4)
user anonymous
331 Please specify the password.
pass anonymous
230 Login successful.
quit
221 Goodbye.
kali@kali:~$ 
```

INFOSEC ✕ Exit Lab

Instructions Resources

Credential Attacks

When port 25 is open, penetration testers can attempt various social engineering attacks:

`nmap -p 25,80 target`

The Social-Engineer Toolkit (SET) is an open-source collection of tools designed to execute attacks targeting system users. It includes features for phishing campaigns, web jacking, PowerShell attacks, and establishing remote connections.

Tools like Medusa and Hydra help penetration testers brute-force weak credentials. After identifying a service from the Nmap scan that allows remote connections, testers can attempt to authenticate using compromised credentials. SSH, a cryptographic network protocol, is commonly used for remote access, requiring either a password or a cryptographic key for authentication.

`nmap -p 22 target`

The following Nmap command scans the FTP service on port 21 using default scripts (-sC) and version detection (-sV):

`nmap target -p 21 -sV -sC`

Even if the scan does not indicate that anonymous login is enabled, you can still test it manually. Both an FTP client and netcat can establish a connection using default credentials:

`nc target 21`

`user anonymous`

`pass anonymous`

To exit the session and return to the root shell, type:

`quit`

What is the purpose of the -p parameter in the Nmap commands?

○ To save the scan results to a file
● To specify the port(s)
○ To specify the IP address
○ To enable version detection

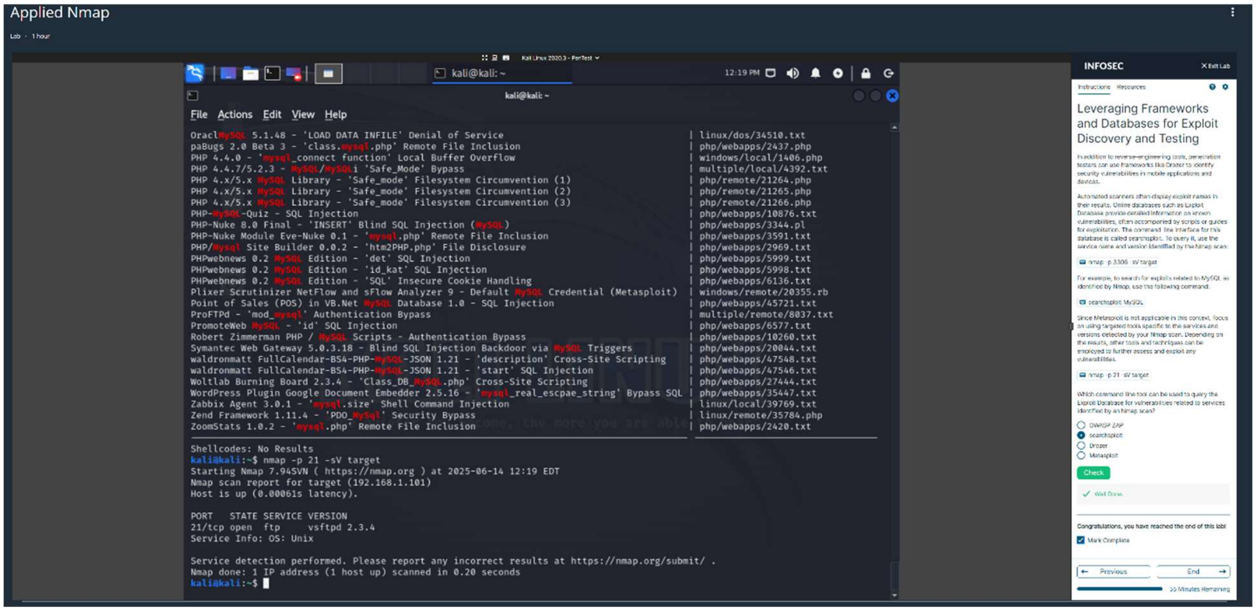
Check

✓ Well Done.

← Previous   Next →

56 Minutes Remaining

Figure 9: Anonymous FTP Login



Applied Nmap

Lab · 1 hour

```
OraclMySQL 5.1.48 - 'LOAD DATA INFILE' Denial of Service                      | linux/dos/34510.txt
paBugs 2.0 Beta 3 - 'class.mysql.php' Remote File Inclusion                   | php/webapps/2437.php
PHP 4.4.0 - 'mysql_connect function' Local Buffer Overflow                    | windows/local/1406.php
PHP 4.4.7/5.2.3 - MySQL/MySQLi 'Safe_Mode' Bypass                             | multiple/local/4392.txt
PHP 4.x/5.x MySQL Library - 'Safe_mode' Filesystem Circumvention (1)          | php/remote/21264.php
PHP 4.x/5.x MySQL Library - 'Safe_mode' Filesystem Circumvention (2)          | php/remote/21265.php
PHP 4.x/5.x MySQL Library - 'Safe_mode' Filesystem Circumvention (3)          | php/remote/21266.php
PHP-MySQL-Quiz - SQL Injection                                                | php/webapps/10876.txt
PHP-Nuke 8.0 Final - 'INSERT' Blind SQL Injection (MySQL)                     | php/webapps/3344.pl
PHP-Nuke Module Eve-Nuke 0.1 - 'mysql.php' Remote File Inclusion              | php/webapps/3591.txt
PHP/Mysql Site Builder 0.0.2 - 'htm2PHP.php' File Disclosure                  | php/webapps/2969.txt
PHPwebnews 0.2 MySQL Edition - 'det' SQL Injection                            | php/webapps/5999.txt
PHPwebnews 0.2 MySQL Edition - 'id_kat' SQL Injection                         | php/webapps/5998.txt
PHPwebnews 0.2 MySQL Edition - 'SQL' Insecure Cookie Handling                 | php/webapps/6136.txt
Plixer Scrutinizer NetFlow and sFlow Analyzer 9 - Default MySQL Credential (Metasploit) | windows/remote/20355.rb
Point of Sales (POS) in VB.Net MySQL Database 1.0 - SQL Injection             | php/webapps/45721.txt
ProFTPd - 'mod_mysql' Authentication Bypass                                   | multiple/remote/8037.txt
PromoteWeb MySQL - 'id' SQL Injection                                         | php/webapps/6577.txt
Robert Zimmerman PHP / MySQL Scripts - Authentication Bypass                  | php/webapps/10260.txt
Symantec Web Gateway 5.0.3.18 - Blind SQL Injection Backdoor via MySQL Triggers | php/webapps/20044.txt
waldronmatt FullCalendar-BS4-PHP-MySQL-JSON 1.21 - 'description' Cross-Site Scripting | php/webapps/47548.txt
waldronmatt FullCalendar-BS4-PHP-MySQL-JSON 1.21 - 'start' SQL Injection      | php/webapps/47546.txt
Woltlab Burning Board 2.3.4 - 'Class_DB_MySQL.php' Cross-Site Scripting       | php/webapps/27444.txt
WordPress Plugin Google Document Embedder 2.5.16 - 'mysql_real_escpae_string' Bypass SQL | php/webapps/35447.txt
Zabbix Agent 3.0.1 - 'mysql.size' Shell Command Injection                     | linux/local/39769.txt
Zend Framework 1.11.4 - 'PDO_MySql' Security Bypass                           | linux/remote/35784.php
ZoomStats 1.0.2 - 'mysql.php' Remote File Inclusion                           | php/webapps/2420.txt
------------------------------------------------------------------------------ ---------------------------------

Shellcodes: No Results
kali@kali:~$ nmap -p 21 -sV target
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-14 12:19 EDT
Nmap scan report for target (192.168.1.101)
Host is up (0.00061s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
kali@kali:~$ 
```

INFOSEC ✕ Exit Lab

Instructions Resources

Leveraging Frameworks and Databases for Exploit Discovery and Testing

In addition to reverse-engineering tools, penetration testers can use frameworks like Drozer to identify security vulnerabilities in mobile applications and devices.

Automated scanners often display exploit names in their results. Online databases such as Exploit Database provide detailed information on known vulnerabilities, often accompanied by scripts or guides for exploitation. The command-line interface for this database is called searchsploit. To query it, use the service name and version identified by the Nmap scan:

`nmap -p 3306 -sV target`

For example, to search for exploits related to MySQL as identified by Nmap, use the following command:

`searchsploit MySQL`

Since Metasploit is not applicable in this context, focus on using targeted tools specific to the services and versions detected by your Nmap scan. Depending on the results, other tools and techniques can be employed to further assess and exploit any vulnerabilities.

`nmap -p 21 -sV target`

Which command line tool can be used to query the Exploit Database for vulnerabilities related to services identified by an Nmap scan?

○ OWASP ZAP
● searchsploit
○ Drozer
○ Metasploit

Check

✓ Well Done.

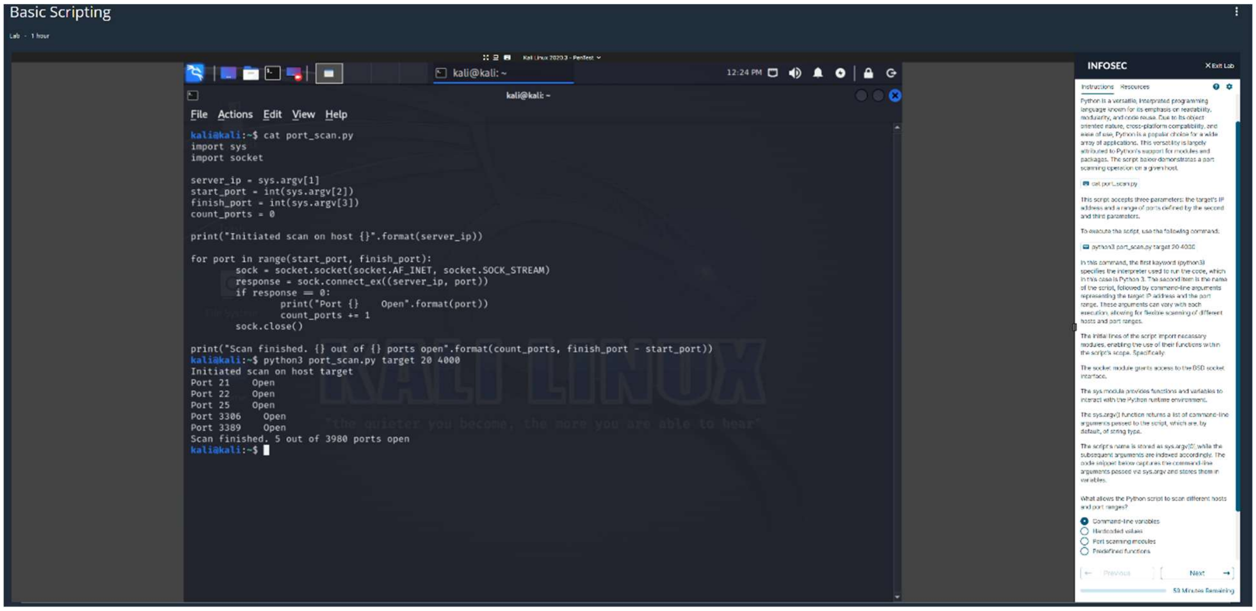
Congratulations, you have reached the end of this lab!

☑ Mark Complete

← Previous   End →

55 Minutes Remaining

Figure 10: MySQL Vulnerabilities



```
kali@kali:~$ cat port_scan.py
import sys
import socket

server_ip = sys.argv[1]
start_port = int(sys.argv[2])
finish_port = int(sys.argv[3])
count_ports = 0

print("Initiated scan on host {}".format(server_ip))

for port in range(start_port, finish_port):
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        response = sock.connect_ex((server_ip, port))
        if response == 0:
                print("Port {}    Open".format(port))
                count_ports += 1
        sock.close()

print("Scan finished. {} out of {} ports open".format(count_ports, finish_port - start_port))
kali@kali:~$ python3 port_scan.py target 20 4000
Initiated scan on host target
Port 21    Open
Port 22    Open
Port 25    Open
Port 3306    Open
Port 3389    Open
Scan finished. 5 out of 3980 ports open
kali@kali:~$
```

Python is a versatile, interpreted programming language known for its emphasis on readability, modularity, and code reuse. Due to its object oriented nature, cross-platform compatibility, and ease of use, Python is a popular choice for a wide array of applications. This versatility is largely attributed to Python's support for modules and packages. The script below demonstrates a port scanning operation on a given host.

```
cat port_scan.py
```

This script accepts three parameters: the target's IP address and a range of ports defined by the second and third parameters.

To execute the script, use the following command:

```
python3 port_scan.py target 20 4000
```

In this command, the first keyword (python3) specifies the interpreter used to run the code, which in this case is Python 3. The second item is the name of the script, followed by command-line arguments representing the target IP address and the port range. These arguments can vary with each execution, allowing for flexible scanning of different hosts and port ranges.

The initial lines of the script import necessary modules, enabling the use of their functions within the script's scope. Specifically:

The socket module grants access to the BSD socket interface.

The sys module provides functions and variables to interact with the Python runtime environment.

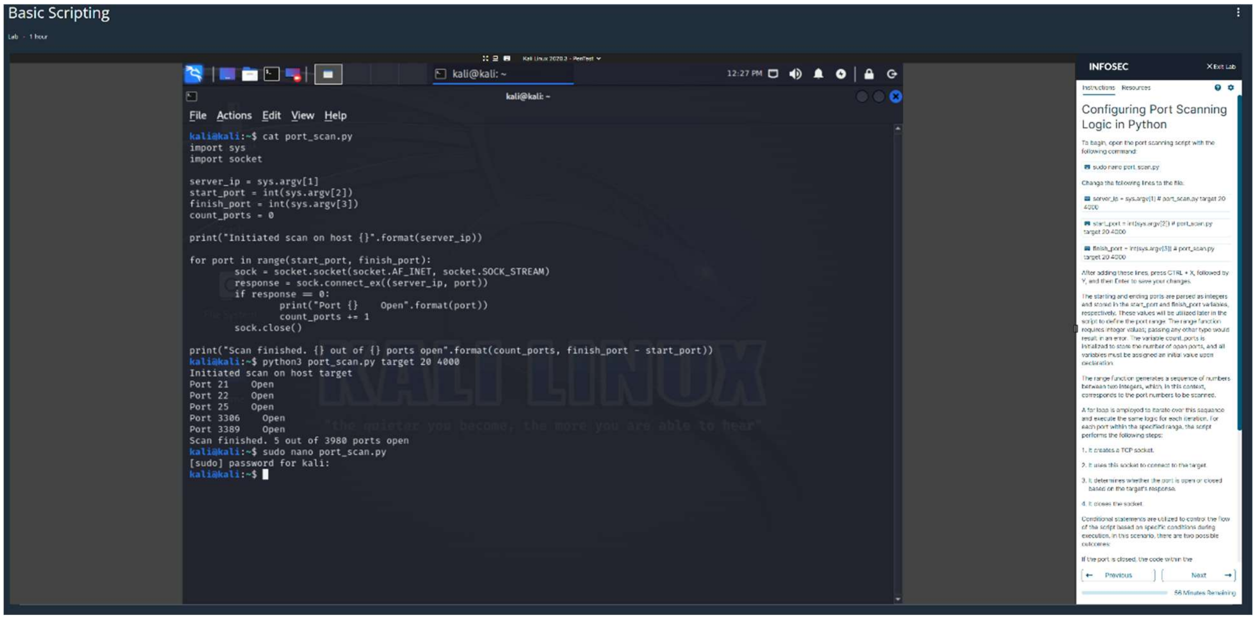
The sys.argv() function returns a list of command-line arguments passed to the script, which are, by default, of string type.

The script's name is stored as sys.argv[0], while the subsequent arguments are indexed accordingly. The code snippet below captures the command-line arguments passed via sys.argv and stores them in variables.

What allows the Python script to scan different hosts and port ranges?

● Command-line variables
○ Hardcoded values
○ Port scanning modules
○ Predefined functions

Figure 11: Python Port Scan



```
kali@kali:~$ cat port_scan.py
import sys
import socket

server_ip = sys.argv[1]
start_port = int(sys.argv[2])
finish_port = int(sys.argv[3])
count_ports = 0

print("Initiated scan on host {}".format(server_ip))

for port in range(start_port, finish_port):
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        response = sock.connect_ex((server_ip, port))
        if response == 0:
                print("Port {}    Open".format(port))
                count_ports += 1
        sock.close()

print("Scan finished. {} out of {} ports open".format(count_ports, finish_port - start_port))
kali@kali:~$ python3 port_scan.py target 20 4000
Initiated scan on host target
Port 21    Open
Port 22    Open
Port 25    Open
Port 3306    Open
Port 3389    Open
Scan finished. 5 out of 3980 ports open
kali@kali:~$ sudo nano port_scan.py
[sudo] password for kali:
kali@kali:~$
```

Configuring Port Scanning Logic in Python

To begin, open the port scanning script with the following command:

```
sudo nano port_scan.py
```

Change the following lines to the file:

```
server_ip = sys.argv[1] # port_scan.py target 20 4000
```

```
start_port = int(sys.argv[2]) # port_scan.py target 20 4000
```

```
finish_port = int(sys.argv[3]) # port_scan.py target 20 4000
```

After adding these lines, press CTRL + X, followed by Y, and then Enter to save your changes.

The starting and ending ports are parsed as integers and stored in the start_port and finish_port variables, respectively. These values will be utilized later in the script to define the port range. The range function requires integer values; passing any other type would result in an error. The variable count_ports is initialized to store the number of open ports, and all variables must be assigned an initial value upon declaration.

The range function generates a sequence of numbers between two integers, which, in this context, corresponds to the port numbers to be scanned.

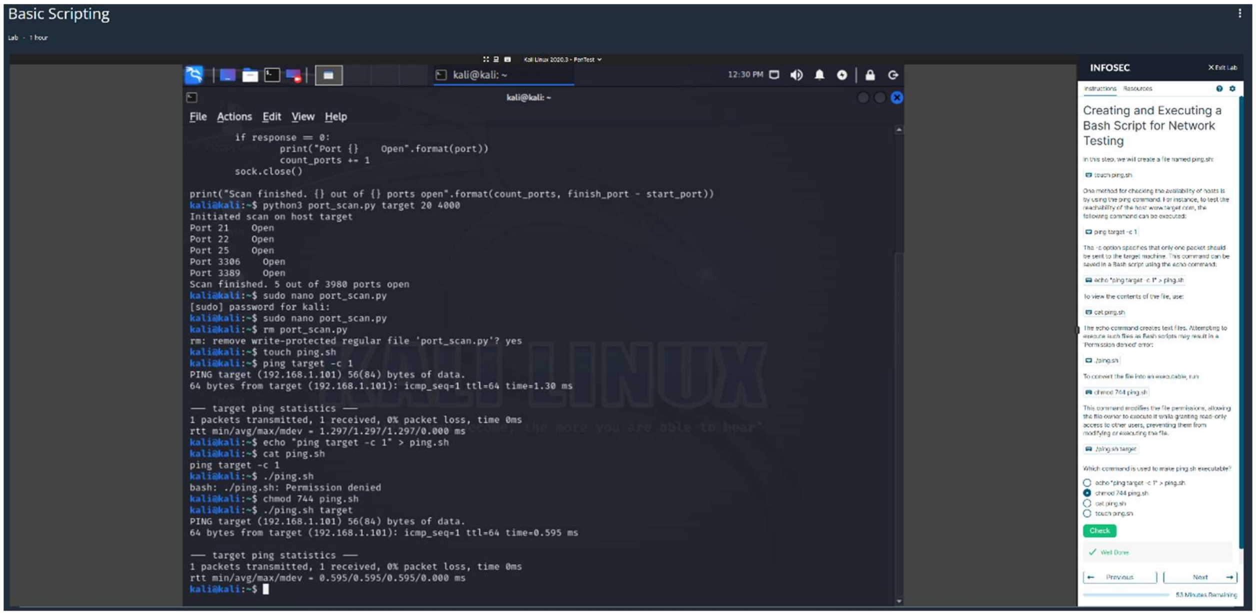
A for loop is employed to iterate over this sequence and execute the same logic for each iteration. For each port within the specified range, the script performs the following steps:

1. It creates a TCP socket.
2. It uses this socket to connect to the target.
3. It determines whether the port is open or closed based on the target's response.
4. It closes the socket.

Conditional statements are utilized to control the flow of the script based on specific conditions during execution. In this scenario, there are two possible outcomes:

If the port is closed, the code within the

Figure 12: Configuring Port Scan Logic



```
        if response == 0:
                print("Port {}    Open".format(port))
                count_ports += 1
        sock.close()

print("Scan finished. {} out of {} ports open".format(count_ports, finish_port - start_port))
kali@kali:~$ python3 port_scan.py target 20 4000
Initiated scan on host target
Port 21    Open
Port 22    Open
Port 25    Open
Port 3306    Open
Port 3389    Open
Scan finished. 5 out of 3980 ports open
kali@kali:~$ sudo nano port_scan.py
[sudo] password for kali:
kali@kali:~$ sudo nano port_scan.py
kali@kali:~$ rm port_scan.py
rm: remove write-protected regular file 'port_scan.py'? yes
kali@kali:~$ touch ping.sh
kali@kali:~$ ping target -c 1
PING target (192.168.1.101) 56(84) bytes of data.
64 bytes from target (192.168.1.101): icmp_seq=1 ttl=64 time=1.30 ms

—— target ping statistics ——
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.297/1.297/1.297/0.000 ms
kali@kali:~$ echo "ping target -c 1" > ping.sh
kali@kali:~$ cat ping.sh
ping target -c 1
kali@kali:~$ ./ping.sh
bash: ./ping.sh: Permission denied
kali@kali:~$ chmod 744 ping.sh
kali@kali:~$ ./ping.sh target
PING target (192.168.1.101) 56(84) bytes of data.
64 bytes from target (192.168.1.101): icmp_seq=1 ttl=64 time=0.595 ms

—— target ping statistics ——
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.595/0.595/0.595/0.000 ms
kali@kali:~$
```

Creating and Executing a Bash Script for Network Testing

In this step, we will create a file named ping.sh:

```
touch ping.sh
```

One method for checking the availability of hosts is by using the ping command. For instance, to test the reachability of the host www.target.com, the following command can be executed:

```
ping target -c 1
```

The -c option specifies that only one packet should be sent to the target machine. This command can be saved in a Bash script using the echo command:

```
echo "ping target -c 1" > ping.sh
```

To view the contents of the file, use:

```
cat ping.sh
```

The echo command creates text files. Attempting to execute such files as Bash scripts may result in a 'Permission denied' error:

```
./ping.sh
```

To convert the file into an executable, run

```
chmod 744 ping.sh
```

This command modifies the file permissions, allowing the file owner to execute it while granting read-only access to other users, preventing them from modifying or executing the file.

```
./ping.sh target
```

Which command is used to make ping.sh executable?

○ echo "ping target -c 1" > ping.sh
● chmod 744 ping.sh
○ cat ping.sh
○ touch ping.sh

Figure 13: Script Automation of Ping

References

EC-Council. (2025, April 22). *What are footprinting and reconnaissance?* Cybersecurity Exchange. https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/basics-footprinting-reconnaissance/

Kumar, M. (2025, May 8). *hping3 - Advanced Guide to Network Testing, Security Audits, and Penetration Testing*. The Geek Institute of Cyber Security - Blog. Retrieved June 14, 2025, from https://blog.geekinstitute.org/2025/05/mastering-hping3-advanced-guide-to-network-testing.html

Nmap.org. (n.d.). *TCP SYN (Stealth) Scan (-SS) | NMAP Network Scanning*. Retrieved June 14, 2025, from https://nmap.org/book/synscan.html

Reason Labs. (2024). *What is Network reconnaissance?* Retrieved June 14, 2025, from https://cyberpedia.reasonlabs.com/EN/network%20reconnaissance.html