ETHICAL HACKING V2
LAB SERIES

Lab 19: Scanning Methodology

Document Version: 2021-10-05

Material in this Lab Aligns to the Following	
Books/Certifications	**Chapters/Modules/Objectives**
All-In-One CEH Chapters ISBN-13: 978-1260454550	2: Reconnaissance: Information Gathering for the Ethical Hacker 3: Scanning and Enumeration 4: Sniffing and Evasion
EC-Council CEH v10 Domain Modules	2: Footprinting and Reconnaissance 3: Scanning Networks 5: Vulnerability Analysis 8: Sniffing

Copyright © 2021 Network Development Group, Inc.
www.netdevgroup.com

NETLAB+ is a registered trademark of Network Development Group, Inc.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries. Google is a registered trademark of Google, LLC.

Contents

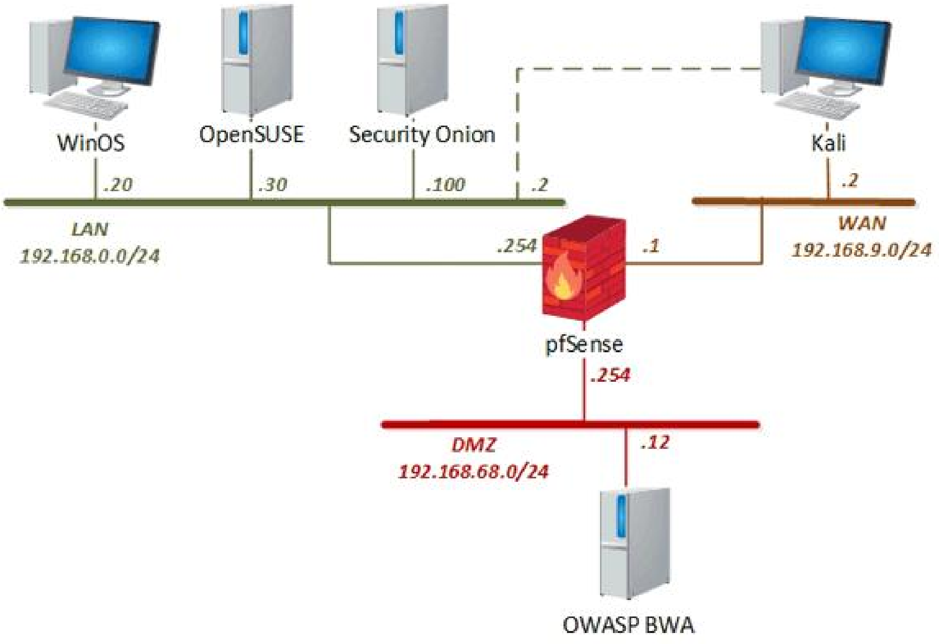
Introduction

As an ethical hacker or pen-tester, you will be called upon to perform port scanning, network scanning, and vulnerability scanning on the IP addresses obtained during the information-gathering phase. Scanning a network refers to a set of procedures for identifying hosts, ports, and services running in a network. In this lab, as the penetration tester, you will perform scanning to help you to identify IP/hostnames, live hosts, vulnerabilities, and services running on the target network.

Objectives

- Discover live host on the network, IP address, and open ports
- Identified running services using NMAP
- Identifying Network Topology
- Vulnerability Scan
- Banner Grabbing
- Packet Crafting

Lab Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

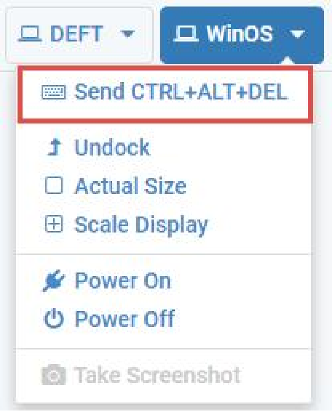
Virtual Machine	IP Address / Subnet Mask	Account (if needed)	Password (if needed)
WinOS	192.168.0.20	Administrator	Train1ng$
Kali Linux	192.168.9.2 192.168.0.2	root	toor

Copyright © 2021 Network Development Group, Inc. www.netdevgroup.com

1 Scanning a Network with Advanced IP Scanner

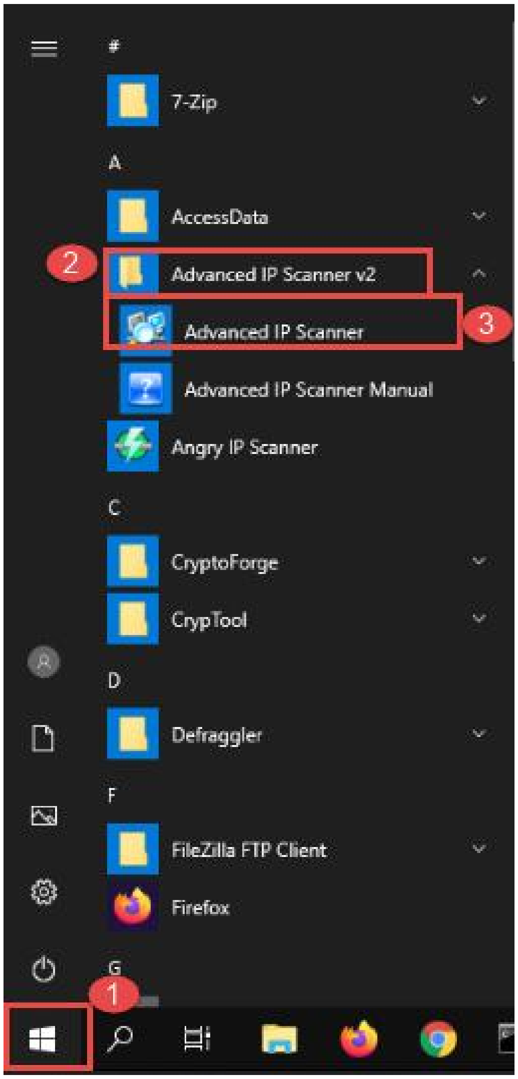
There are many great tools for scanning networks. In this exercise, we will be using a free tool called *Advanced IP Scanner* to perform scans and identify live devices on the network.

1. To begin, launch the **WinOS** virtual machine to access the graphical login screen.
   1.1. Select **Send CTRL+ALT+DEL** from the dropdown menu to be prompted with the login screen.

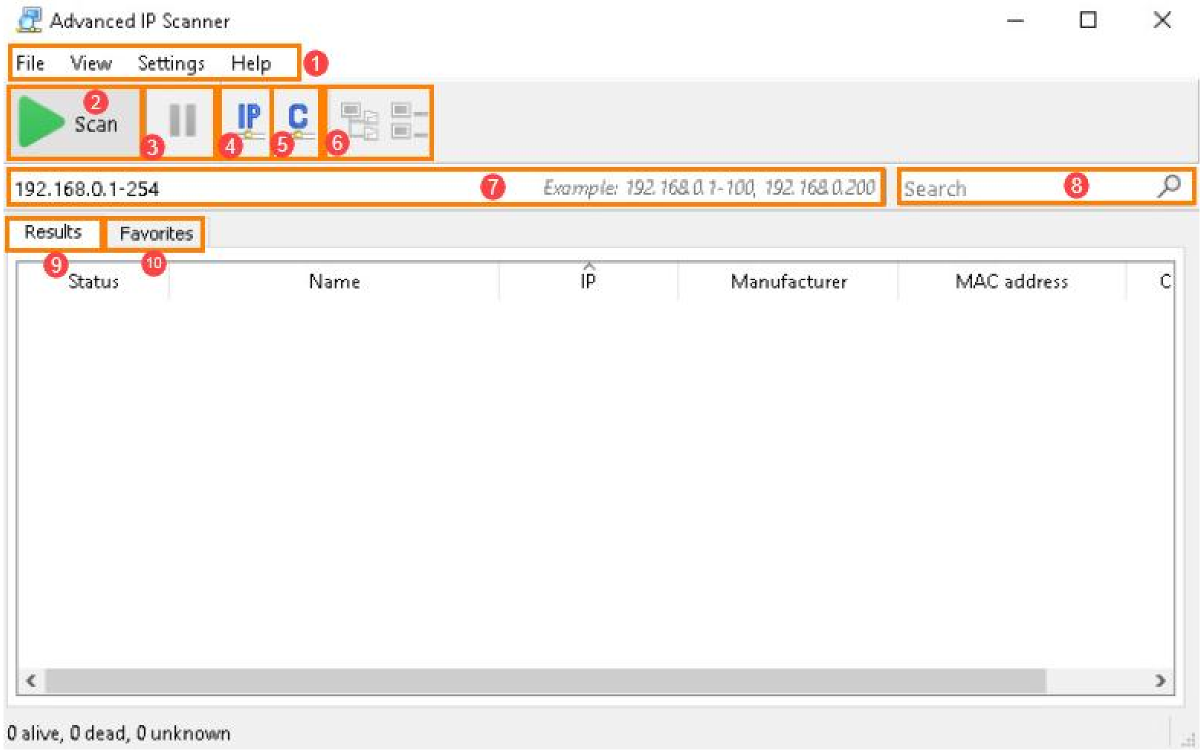


   1.2. Log in as `Administrator` using the password: `Train1ng$`

2. Once you are logged in, navigate to **Start Menu > Advanced IP Scanner v2 > Advanced IP Scanner** as seen in *items* **1, 2,** and **3** below:

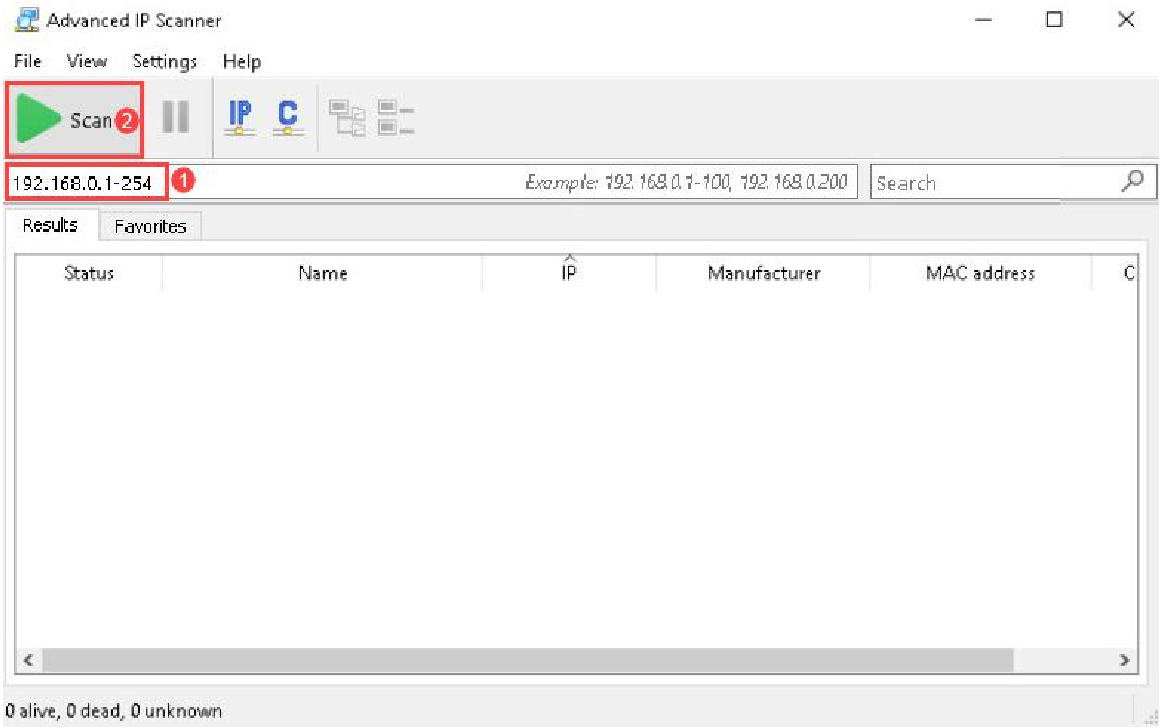


3. The *Advanced IP Scanner* GUI will appear. It is a simple-looking interface that allows you to gather some useful information about the live systems in the network. The table below the following screenshot provides details about the different features of this software.

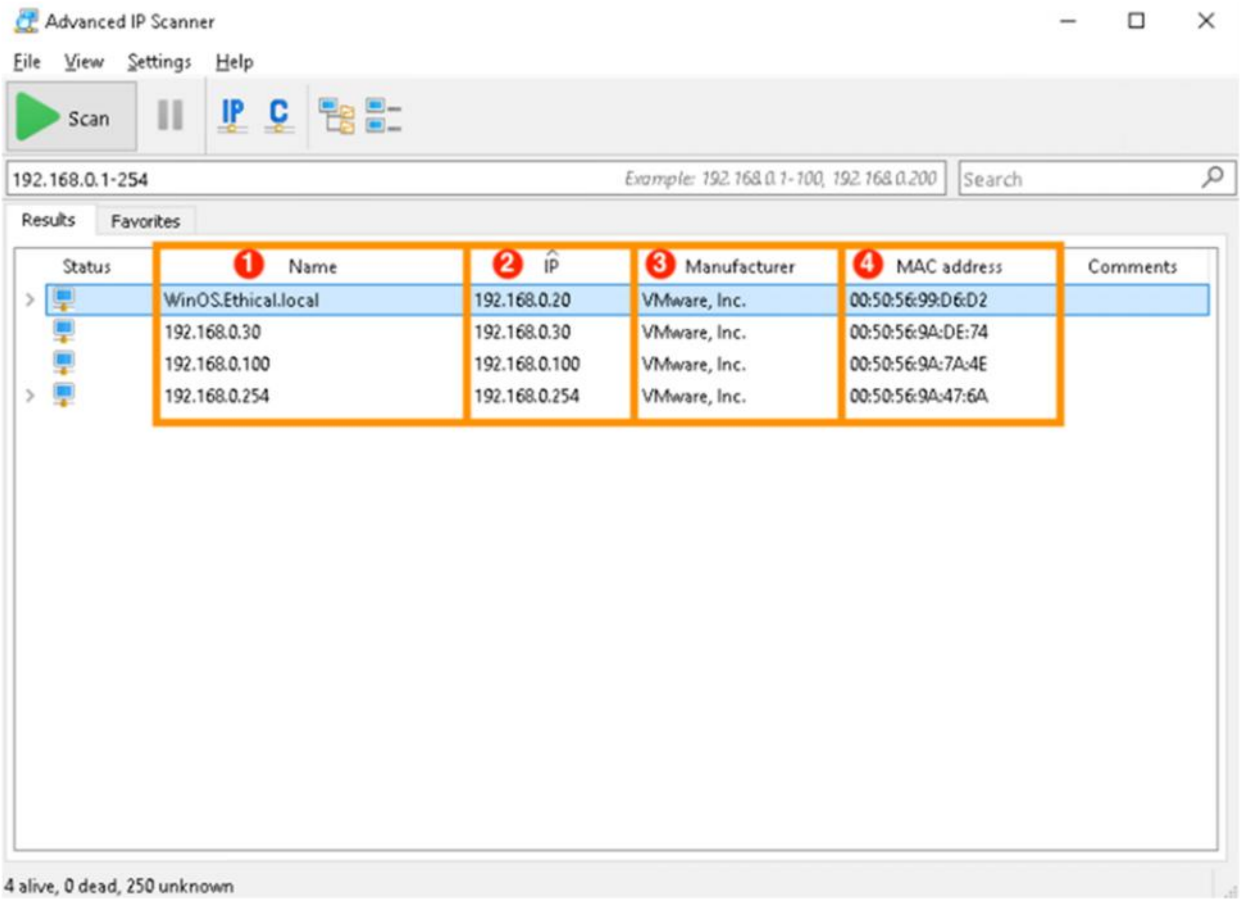


Item no.	Name	Description
1	Menu Bar	This contains dropdown lists that allow you to save, change view options, adjust settings, and get help
2	Scan	This button starts the scanning process
3	Pause	This button pauses the scanning process
4	IP	This button scans the subnet of the current computer
5	C	This button scans the class C subnet
6	Expand and contract all	These buttons allow you to expand and contract the results of the scan
7	IP Range	This field lists the IP address range to be scanned
8	Search	This allows you to search the results
9	Results	This pane lists the results of the scan in column view
10	Favorites	This tab lists the IP addresses that were favorited by the user

4. Now that you are familiar with the interface, let us jump right in and start scanning with *Advanced IP Scanner*. Let us begin by typing the IP range **`192.168.0.1 - 254`** in the *IP Range* field, as seen in *item* **1** below. Once you are done, click **Scan,** as seen in *item* **2.**



5. *Advanced IP Scanner* will begin scanning for live IP addresses, and the list will populate each time it identifies one. The column in *item* **1** below lists the IP address or hostname of the live system. *Item* **2** lists only the IP address, and *item* **3** details the name of the manufacturer of the network card for each live system. *Item* **4** is the MAC address of the live system. This information is extremely helpful in gathering information as it helps you to identify the systems and network services that they are running.

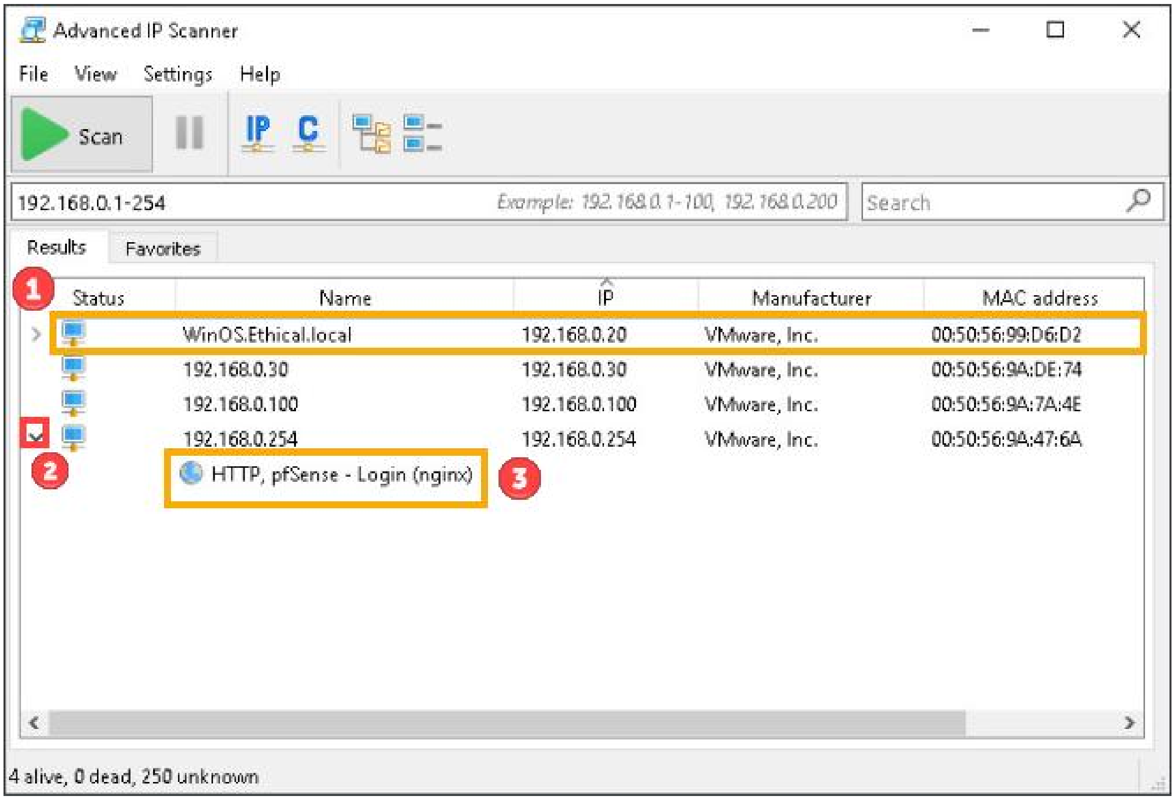


192.168.0.1-254

Status	Name	IP	Manufacturer	MAC address	Comments
	WinOS.Ethical.local	192.168.0.20	VMware, Inc.	00:50:56:99:D6:D2	
	192.168.0.30	192.168.0.30	VMware, Inc.	00:50:56:9A:DE:74	
	192.168.0.100	192.168.0.100	VMware, Inc.	00:50:56:9A:7A:4E	
	192.168.0.254	192.168.0.254	VMware, Inc.	00:50:56:9A:47:6A	

4 alive, 0 dead, 250 unknown

6. The *WinOS.Ethical.local* system seen in the list and in *item* **1** below is the hostname of Windows OS that you are currently using. There is some additional information that can be gained from one of the IP addresses in the list. We can determine that one of the systems is running a service by looking for the arrow beside it, as seen in *item* **2.** Click the arrow to expand the entry. The running service will appear below the entry, as seen in *item* **3.** This indicates that the system with IP address 192.168.0.254 is running HTTP service. This means you should be able to invoke a webpage by visiting the IP address with a web browser and attempt using the default password for *pfSense* or any other password you think could work.



192.168.0.1-254

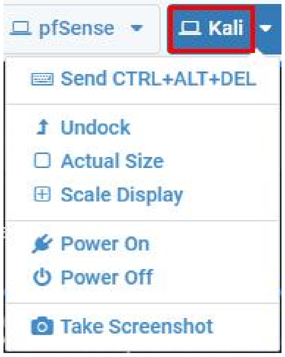
Status	Name	IP	Manufacturer	MAC address
	WinOS.Ethical.local	192.168.0.20	VMware, Inc.	00:50:56:99:D6:D2
	192.168.0.30	192.168.0.30	VMware, Inc.	00:50:56:9A:DE:74
	192.168.0.100	192.168.0.100	VMware, Inc.	00:50:56:9A:7A:4E
	192.168.0.254	192.168.0.254	VMware, Inc.	00:50:56:9A:47:6A
	HTTP, pfSense - Login (nginx)			

4 alive, 0 dead, 250 unknown

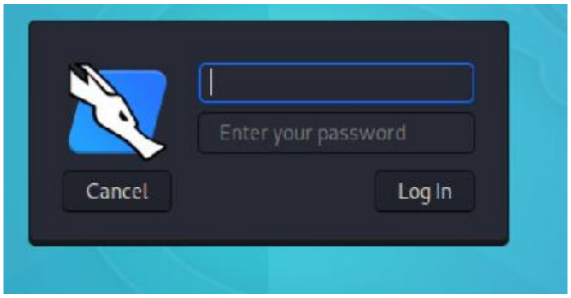
7. As you saw in this exercise, scanning with *Advanced IP Scanner* is great for getting data about systems in the network. It can be a great starting point when identifying a workstation to target.

2 Drawing Network Topology Using Simple Network Scans

1. Switch to the **Kali Linux** virtual machine to access the graphical login screen.

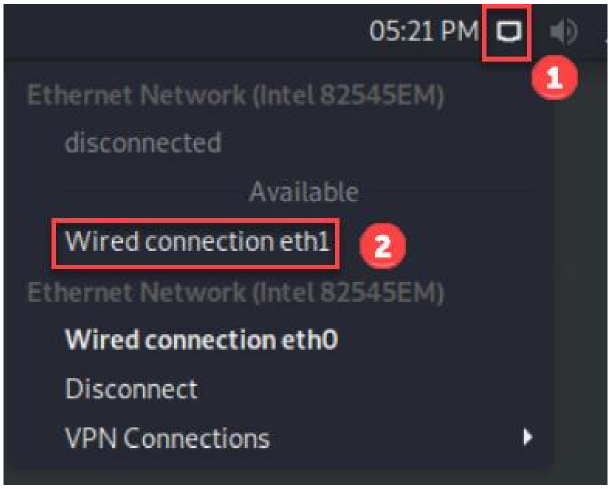


1.1. Log in as `root` using the password: `toor`



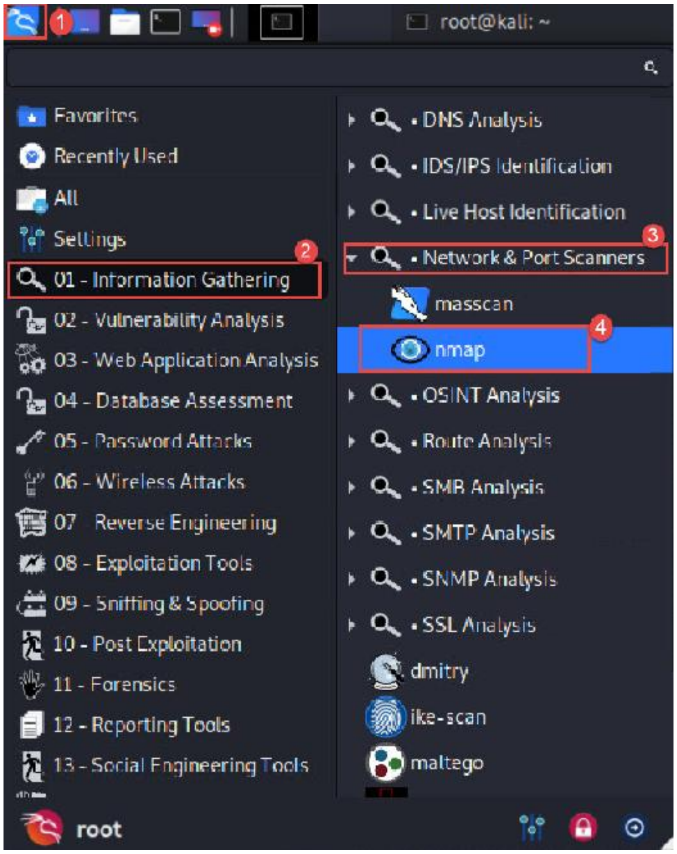
> If a terminal window appears when you log in, close it by clicking the **X** at the top-right corner of the window.

2. Before starting, we need to ensure that the host machine is on the same network as the target machine. Select the **Ethernet network connection** from the navigation panel, as seen in *item* 1. Then click **Wired connection eth1** to enable Kali Linux to configure with the IP address 192.168.0.2 as seen in *item* **2**.



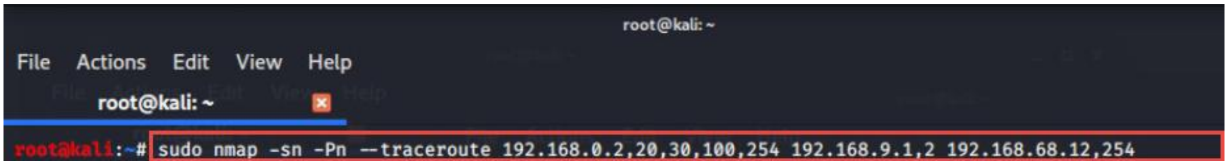
> STOP The target machine is on the network 192.168.0.0/24. The Kali Linux VM is configured with both IP addresses and can easily interchanged.

3. In this exercise, we will use *Nmap* to scan the network and create a map based on the feedback we get. A typical scan of multiple subnets would be ideal in this case; however, due to time constraints, we will assume that all the hosts on the network were already discovered. We will list the target IP addresses and perform the scan. Let us start by launching **nmap.** To do this, navigate to **Whisker Menu** > **Information Gathering** > **Network & Port Scanners** > **nmap** as seen in *items* **1**, **2**, **3,** and **4** below.



4. Once *Nmap* starts, type `sudo nmap -sn -Pn --traceroute 192.168.0.2,20,30,100,254 192.168.9.1,2 192.168.68.12,254` then press **Enter.**

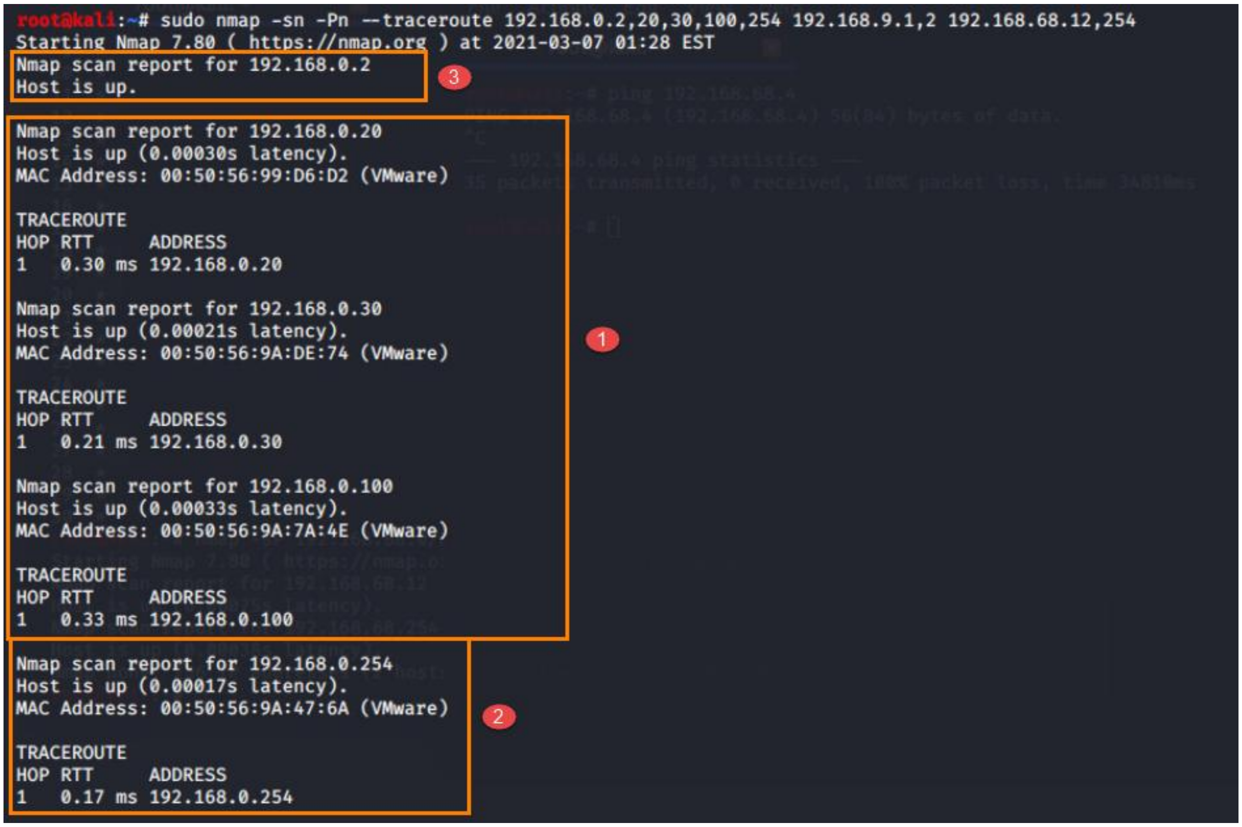
```
root@kali:~# sudo nmap -sn -Pn --traceroute 192.168.0.2,20,30,100,254
192.168.9.1,2 192.168.68.12,254
```



```
root@kali:~# sudo nmap -sn -Pn --traceroute 192.168.0.2,20,30,100,254 192.168.9.1,2 192.168.68.12,254
```

- Subnet **192.168.0.0/24** (192.168.0.2,20,30,100,254)
- Subnet **192.168.9.0/24** (192.168.9.1,2)
- Subnet **192.168.68.0/24** (192.168.68.12,254)

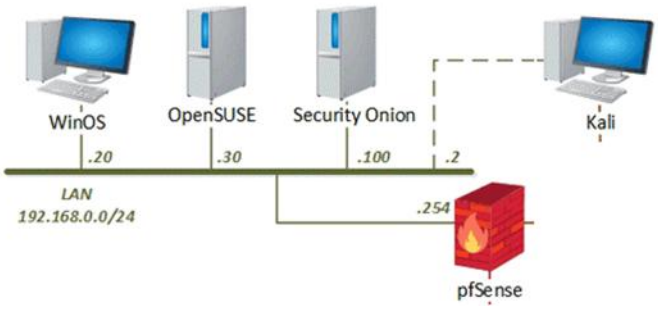
5. The command will list the round-trip Time (RTT) and the number of hops it takes to get to each host. This will allow us to determine the structure of the network, such as gateways, separate LANs, etc. Each scan can provide a different round-trip time and latency. Take the scan below, for example. As you can see from the screenshot, the IP addresses **192.168.0.20**, **192.168.0.30**, and **192.168.0.100** all have a round-trip time (RTT) between 0.21 and 0.33 milliseconds, as seen in *item* **1**. The node **192.168.0.254** has the shortest response time, 0.17 milliseconds, as seen in *item* **2,** which suggests that it is the network's gateway. Similarly, your scan should show results where the gateway has the shortest RTT. We also know from previous exercises that this IP address is a *pfSense* host. The node **192.168.0.2** only returned with the result, *host is up,* as seen in *item* **3**. This is because the host we are scanning is the *Kali Linux* host we are using.



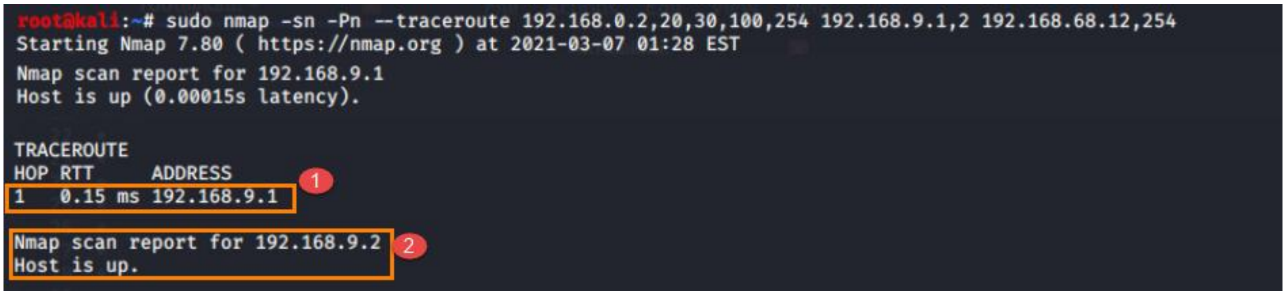
```
root@kali:~# sudo nmap -sn -Pn --traceroute 192.168.0.2,20,30,100,254 192.168.9.1,2 192.168.68.12,254
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-07 01:28 EST
Nmap scan report for 192.168.0.2
Host is up.

Nmap scan report for 192.168.0.20
Host is up (0.00030s latency).
MAC Address: 00:50:56:99:D6:D2 (VMware)

TRACEROUTE
HOP RTT     ADDRESS
1   0.30 ms 192.168.0.20

Nmap scan report for 192.168.0.30
Host is up (0.00021s latency).
MAC Address: 00:50:56:9A:DE:74 (VMware)

TRACEROUTE
HOP RTT     ADDRESS
1   0.21 ms 192.168.0.30

Nmap scan report for 192.168.0.100
Host is up (0.00033s latency).
MAC Address: 00:50:56:9A:7A:4E (VMware)

TRACEROUTE
HOP RTT     ADDRESS
1   0.33 ms 192.168.0.100
Nmap scan report for 192.168.0.254
Host is up (0.00017s latency).
MAC Address: 00:50:56:9A:47:6A (VMware)

TRACEROUTE
HOP RTT     ADDRESS
1   0.17 ms 192.168.0.254
```

Due to the way that Standard Transport Control Protocol (TCP) is defined, it inherently causes variations in RTT. The RTT is the amount of time it takes for a signal to be sent *plus* the amount of time it takes for it to be acknowledged having received it. Variations are very common in a live environment, and you should examine each on a case-by-case basis.

6. As you are aware, the results of the scans for the network 192.168.0.0/24 coincide with the network diagram for this lab, as seen in the diagram below.



7. The next section we will look at is the result for the network 192.168.9.0/24. As you can see in *item* **2**, The IP address *192.168.9.2* responded with *host is up,* which we know is our *Kali Linux* host. Since we know that the round-trip time (RTT) can vary, we will use the example in the screenshot below. The round-trip time for the IP address 192.168.9.1 as seen in *item* **1,** is 0.15 milliseconds. This is indicative that that node is a gateway as well. We can do further checks to see if it is the same *pfSense* host being used as a router.



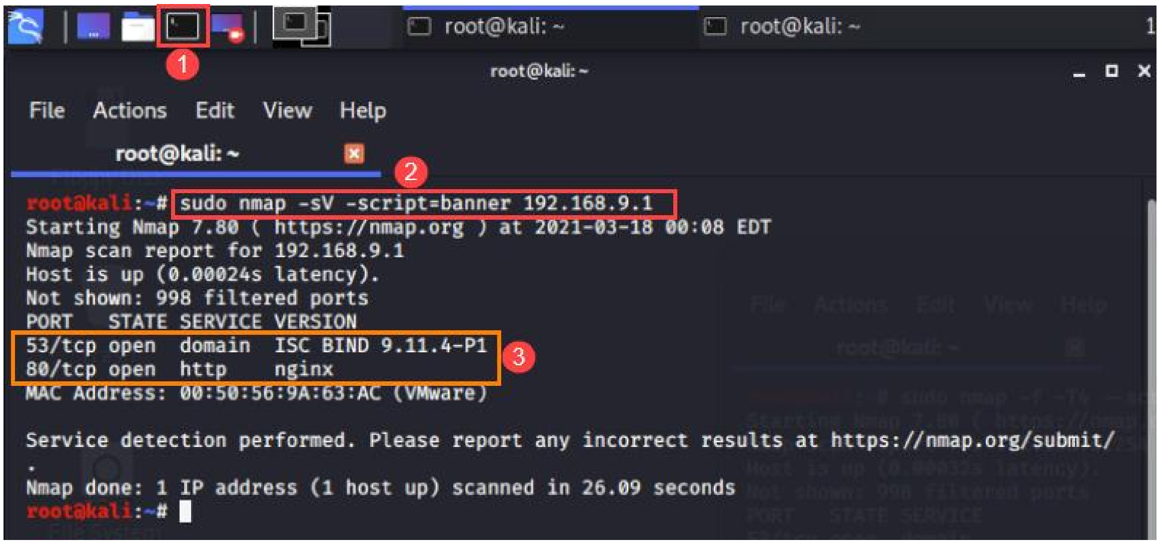
```
root@kali:~# sudo nmap -sn -Pn --traceroute 192.168.0.2,20,30,100,254 192.168.9.1,2 192.168.68.12,254
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-07 01:28 EST
Nmap scan report for 192.168.9.1
Host is up (0.00015s latency).

TRACEROUTE
HOP RTT      ADDRESS
1   0.15 ms 192.168.9.1

Nmap scan report for 192.168.9.2
Host is up.
```

Remember the Kali host has 2 network interfaces:

- 192.168.9.2
- 192.168.0.2

8. To do this, open a new **Terminal** window by clicking the icon from the navigator panel as seen in *item* **1** below. Then type `sudo nmap -sV -script=banner 192.168.9.1` and press **Enter,** as seen in *item* **2.** Look at the results to see what we can learn about this host. The command shows the open ports and the services that are running on the host. It is also supposed to get the operating system but is not always successful at this. If you notice, *item* **3** lists open HTTP (80) and DNS (53) ports and their versions Nginx and ISC BIND 9.11, respectively. *Nginx* is a webserver like *Apache* and is known to be used with *pfSense*. ISC BIND is an implementation of the Domain Name System (DNS). It performs both main DNS server roles and acts as an authoritative name server for domains. This is enough evidence to determine that it is the same *pfSense* host that we identified before.

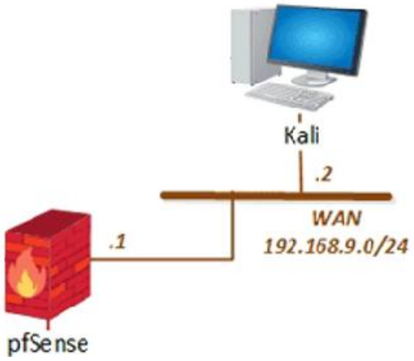
```
root@kali:~# sudo nmap -sV -script=banner 192.168.9.1
```



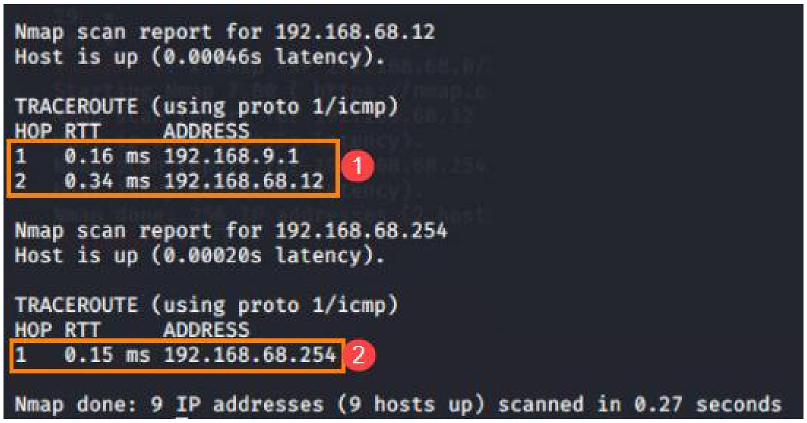
```
root@kali:~# sudo nmap -sV -script=banner 192.168.9.1
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-18 00:08 EDT
Nmap scan report for 192.168.9.1
Host is up (0.00024s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
53/tcp open  domain  ISC BIND 9.11.4-P1
80/tcp open  http    nginx
MAC Address: 00:50:56:9A:63:AC (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 26.09 seconds
root@kali:~# 
```

We can go further by browsing to the IP address, but that will not be covered in this lab. Feel free to type the IP address (192.168.9.1) and observe the results.

9. Looking at the diagram, we can determine that the results of the scans for the network 192.168.9.0/24 coincides with the network diagram for this lab, as seen in the diagram below.



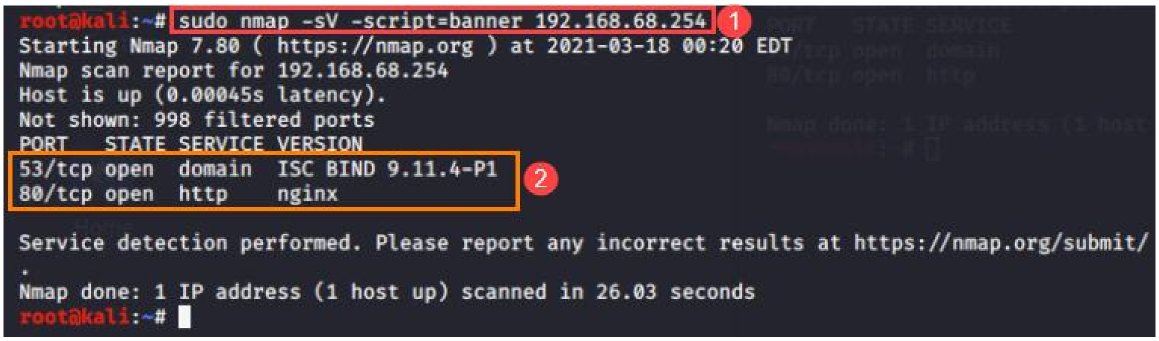
10. Switch back to the first terminal window to continue reviewing the results of the traceroute scan. The last part of the scan lists the results for the network 192.168.68.0/24. As you can see in *item* **1**, the scan shows that there are 2 hops to get to the host at IP Address *192.168.68.12*. This is normal since we know that the *Kali Linux* host does not have an interface directly connected to that network, so it must go through a gateway to get there. The next host in *item* **2** is *192.168.68.254,* but it is 1 hop away. This suggests that this is a gateway node. We can confirm this by taking a closer look at the round-trip time value. Remember that the round-trip time (RTT) can vary; we will use the example in the screenshot below. As you can see, the response time for *192.168.68.12* is 0.34 milliseconds, while the time for *192.168.68.254* is 0.15 milliseconds. Let us run the banner grabbing command we did in step 7 to learn more about it.



```
Nmap scan report for 192.168.68.12
Host is up (0.00046s latency).

TRACEROUTE (using proto 1/icmp)
HOP RTT     ADDRESS
1   0.16 ms 192.168.9.1
2   0.34 ms 192.168.68.12

Nmap scan report for 192.168.68.254
Host is up (0.00020s latency).

TRACEROUTE (using proto 1/icmp)
HOP RTT     ADDRESS
1   0.15 ms 192.168.68.254

Nmap done: 9 IP addresses (9 hosts up) scanned in 0.27 seconds
```

11. Switch back to the second terminal window we used to scan the *pfSense* host in step 7. Once there, type `sudo nmap -sV -script=banner 192.168.68.254` and press **Enter** as seen in *item* **1.** The results of the scan are the same as the ones from step 7. This means the host is the same *pfSense* host, and it has 3 IP addresses and is used as a router on this network.

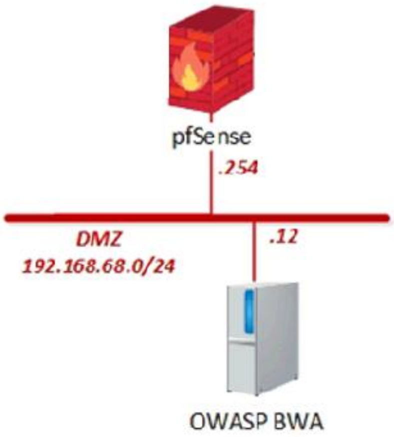
```
root@kali:~# sudo nmap -sV -script=banner 192.168.68.254
```



```
root@kali:~# sudo nmap -sV -script=banner 192.168.68.254
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-18 00:20 EDT
Nmap scan report for 192.168.68.254
Host is up (0.00045s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE VERSION
53/tcp open  domain  ISC BIND 9.11.4-P1
80/tcp open  http    nginx

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 26.03 seconds
root@kali:~# 
```

The option: **--script=banner** refers to the technique used to gain information about a computer system on a network and the services running on its open ports. It is referred to as **Banner grabbing**.

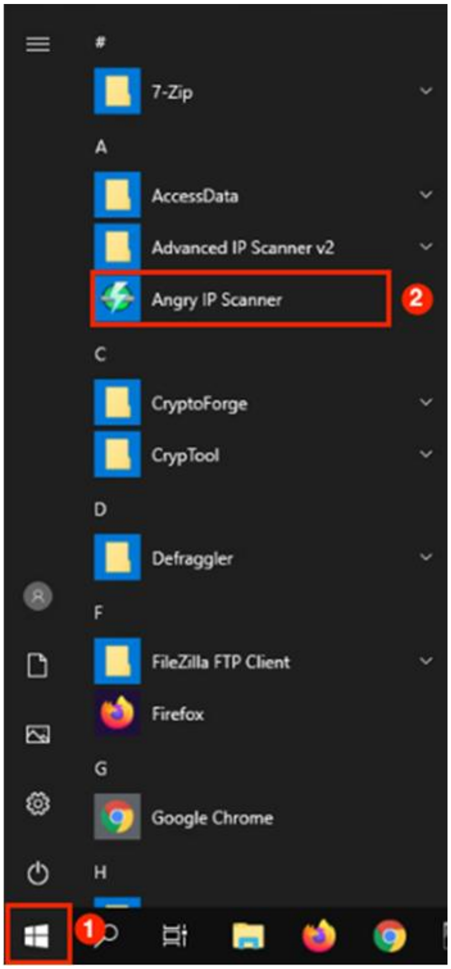
12. Going back to our diagram, we can determine that the results of the scans for the network 192.168.68.0/24 coincides with the network diagram for this lab, as seen in the diagram below.



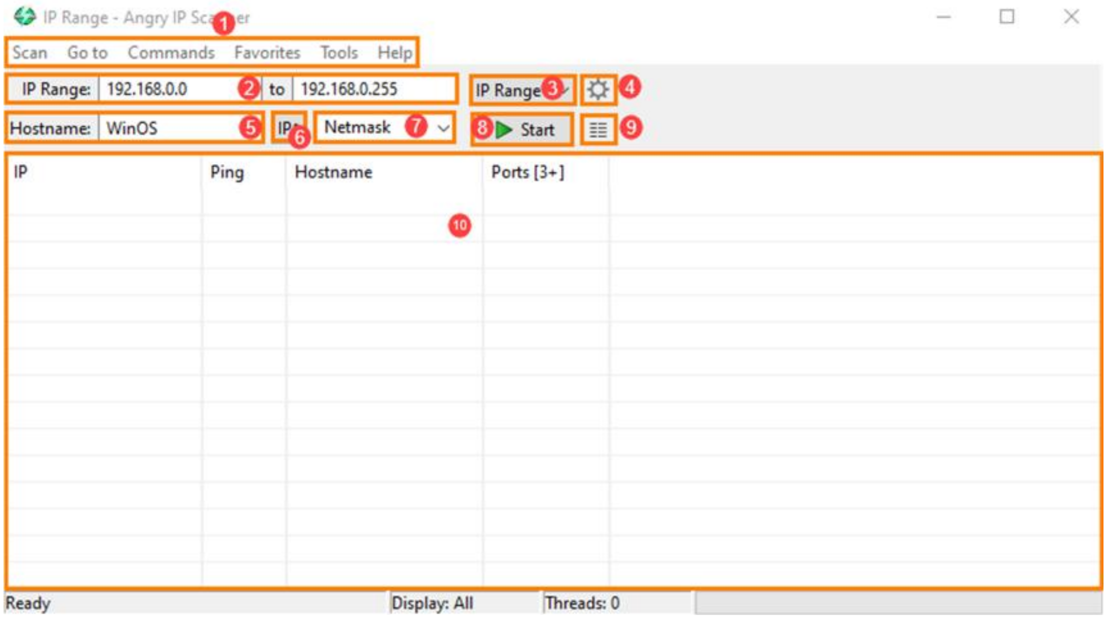
13. This completes our network diagram and gives us more insight into the time of hosts running on the network. It also gives us an idea of the best routes to take and helps us determine how difficult the target will be to hack.

3 Scanning a Network with Angry IP Scanner

1. Switch back to the **WinOS** virtual machine. Another great scanning tool is *Angry IP Scanner*. Let us jump right into it by navigating to **Start Menu > Angry IP Scanner,** as seen in *items* **1** and **2** below:

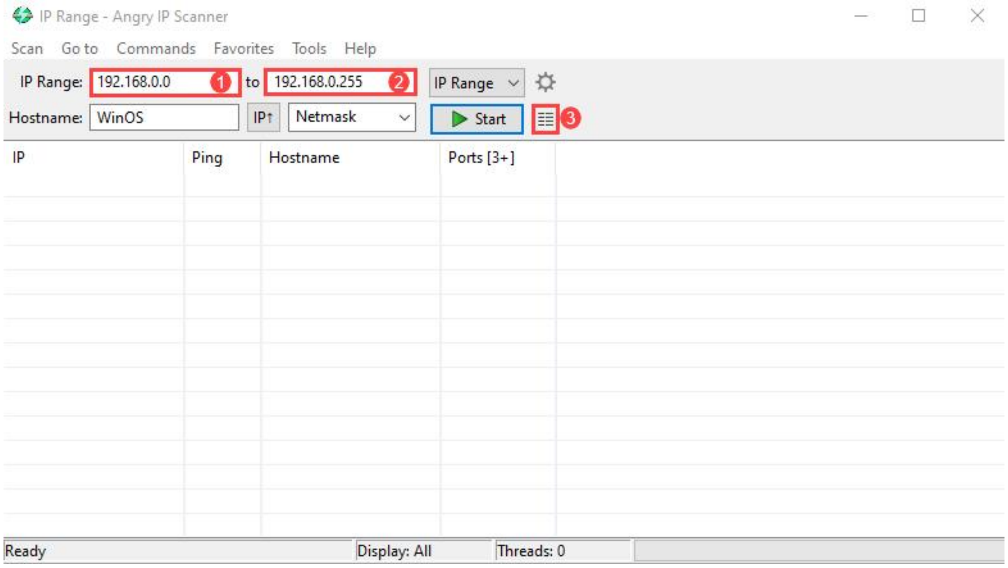


2. The *Angry IP Scanner* GUI will appear. Like the last scanner you used, this is a simple-looking interface as well. The table below the following screenshot provides details about the different features of this software.

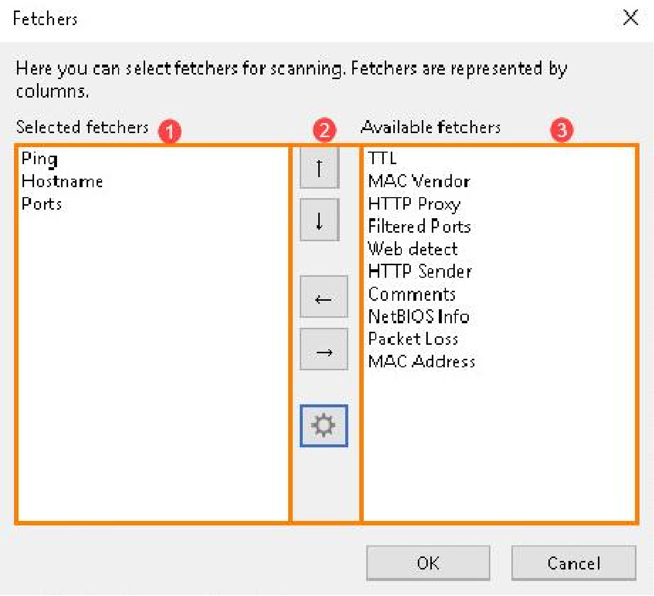


Item no.	Name	Description
1	Menu Bar	This contains dropdown lists that allow you to save, change view options, adjust settings and get help.
2	IP Range	This field lists the IP address range to be scanned
3	IP Feeder menu	This dropdown menu lists the different IP sources. You can type the range or switch to a text file with IP addresses in it or choose a random IP.
4	Preferences	This button contains additional settings.
5	Hostname	This field contains the target hostname
6	IP Source	This button allows you to choose IP addresses from the network adapters in the computer.
7	Netmask	This dropdown menu allows you to choose the subnet mask for the intended target range.
8	Start	This button starts the scanning process.
9	Fetchers	This button provides options to recover different types of data from the scanned hosts.
10	List	This pane lists the results of the scan in column view.

3. Now that you are familiar with the interface, let us start scanning with *Angry IP Scanner*. Let us begin by typing the IP address **192.168.0.0** in the first *IP Range* field seen *item* **1** below. Next, type the IP address **192.168.0.255** in the second *IP Range* field seen in *item* **2** below. Once you are done, click the **Fetchers** button seen in *item* **3** so we can choose what kind of data we want to scan for**.**



4. The *Fetchers* window will open. The items listed in the left pane, seen in *item* **1,** are the default and currently selected *fetchers*. You can use the buttons in *item* **2** to select, remove, reorder, and configure the different *fetchers*. The pane on the right, seen in *item* **3,** contains the available *fetchers*. The results for each *fetcher* will appear as additional columns in the main GUI results pane.


Fetchers

Here you can select fetchers for scanning. Fetchers are represented by columns.

Selected fetchers 1

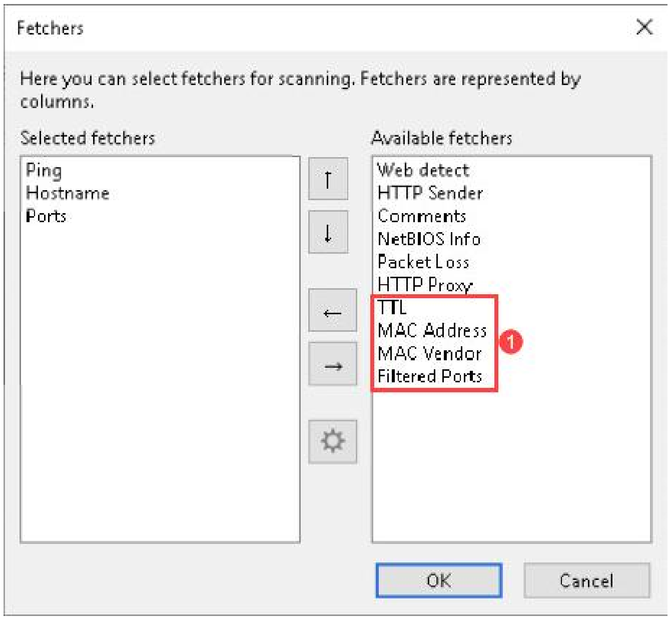
Ping
Hostname
Ports

2

Available fetchers 3

TTL
MAC Vendor
HTTP Proxy
Filtered Ports
Web detect
HTTP Sender
Comments
NetBIOS Info
Packet Loss
MAC Address

5. Now let us select 4 of these *fetchers* for our scan. If your *selected fetchers* already include **TTL, MAC Address, MAC Vendor,** and **Filtered Ports**, please jump this step and go to step 6. If they are not presented in the *selected fetchers,* double-click on each of the following *fetchers* **TTL, MAC Address, MAC Vendor,** and **Filtered Ports.** They are also highlighted as *item* **1** below. Once you select them, they will move to the *Selected fetchers* pane on the left side.



Fetchers

Here you can select fetchers for scanning. Fetchers are represented by columns.

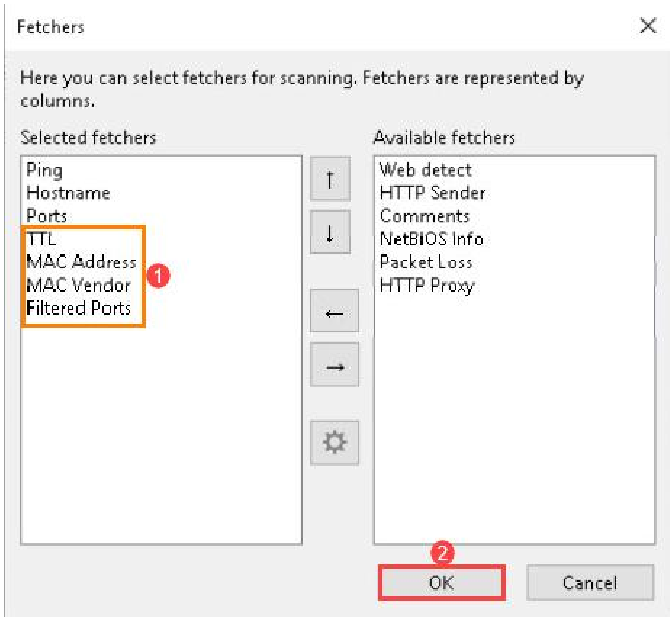
Selected fetchers

Ping
Hostname
Ports

Available fetchers

Web detect
HTTP Sender
Comments
NetBIOS Info
Packet Loss
HTTP Proxy
TTL
MAC Address
MAC Vendor
Filtered Ports

6. Your *Selected fetchers* pane should look like the one in *item* **1** below. Once you verify that they are all there, click **OK,** as seen in *item* **2** below.



Fetchers

Here you can select fetchers for scanning. Fetchers are represented by columns.

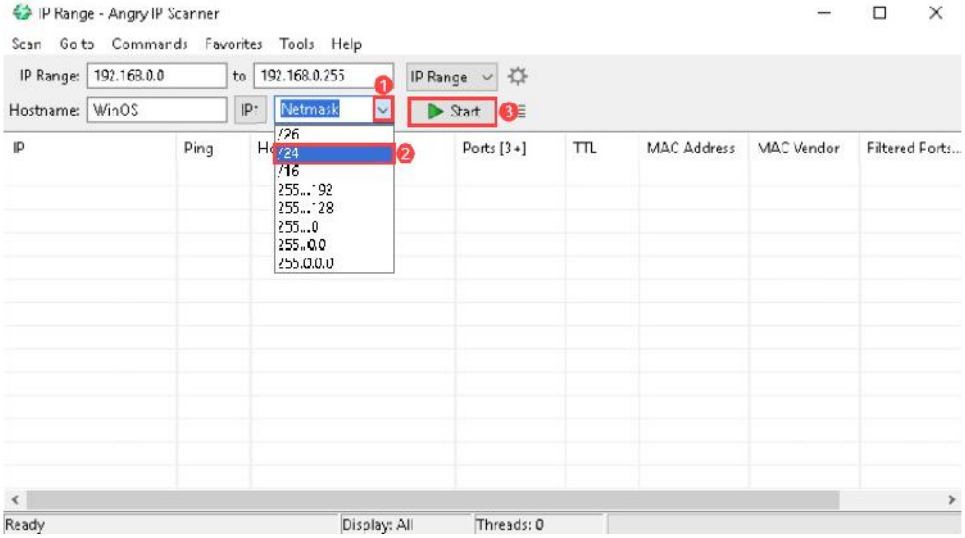
Selected fetchers

Ping
Hostname
Ports
TTL
MAC Address
MAC Vendor
Filtered Ports

Available fetchers

Web detect
HTTP Sender
Comments
NetBIOS Info
Packet Loss
HTTP Proxy

7. The last thing we will do is select the subnet mask. To do this, click the dropdown arrow **Netmask** as seen in *item* **1** below. Select **/24** from the dropdown list, as seen in *item* **2.** Once that is done, we can go ahead and start the scan by clicking **Start,** as seen in *item* **3.**



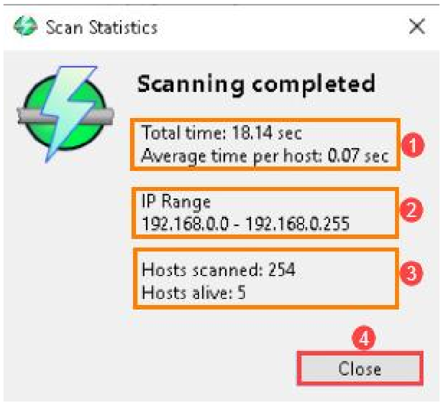
IP Range: 192.168.0.0 to 192.168.0.255

Hostname: WinOS

IP↑ Netmask

Ready | Display: All | Threads: 0

8. This scan is normally pretty quick. Once it is done, the *Scan Statistics* window will appear. It provides a summary of the scan, including the scanning speed seen in *item* **1,** the scanned IP range seen in *item* **2,** and the status of the hosts seen in *item* **3.** Click **Close** as seen in *item* **4** to review the findings.



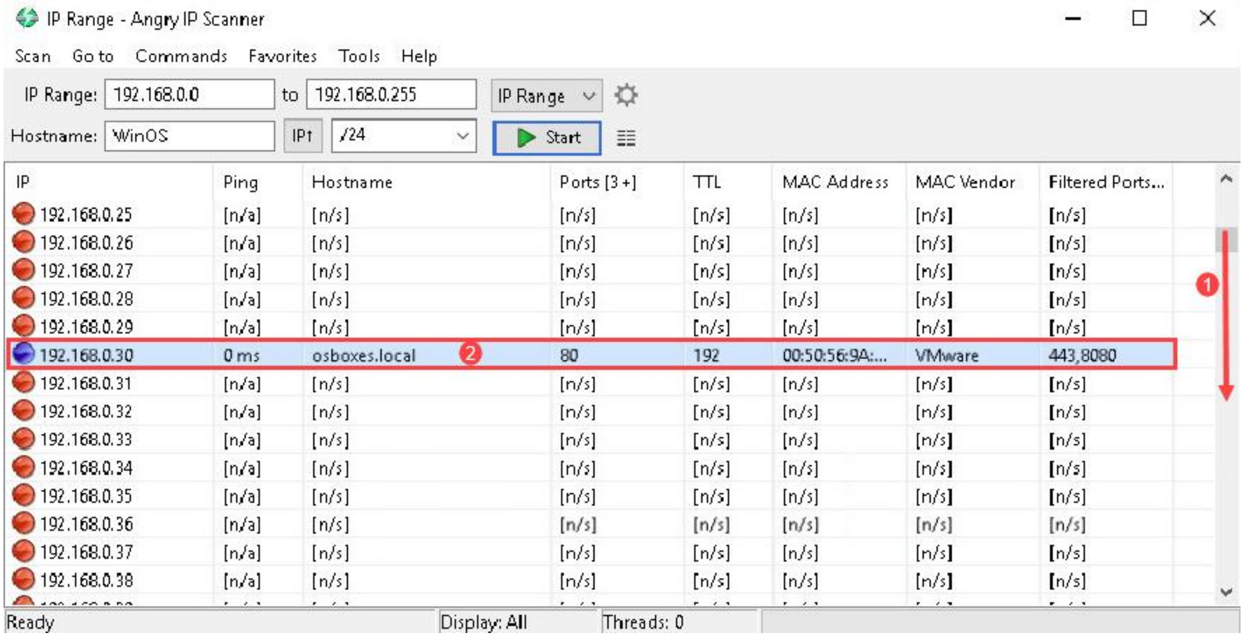
Scanning completed

Total time: 18.14 sec
Average time per host: 0.07 sec

IP Range
192.168.0.0 - 192.168.0.255

Hosts scanned: 254
Hosts alive: 5

9. Let us look at the results. We will look at 1 of the live systems. As you can see, the live hosts appear as a blue circle in the *IP* column. Let us scroll down using the arrow or scroll bar, seen in *item* **1,** until you get to the IP address **192.168.0.30,** as seen in *item* **2** below. Once there, double-click the row to open an information window.



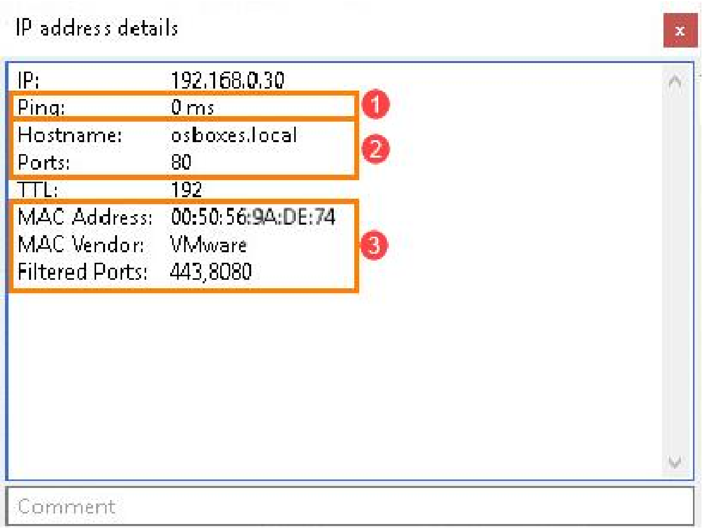
IP Range: 192.168.0.0 to 192.168.0.255

Hostname: WinOS /24

IP	Ping	Hostname	Ports [3+]	TTL	MAC Address	MAC Vendor	Filtered Ports...
● 192.168.0.25	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.26	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.27	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.28	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.29	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.30	0 ms	osboxes.local	80	192	00:50:56:9A:...	VMware	443,8080
● 192.168.0.31	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.32	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.33	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.34	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.35	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.36	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.37	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]
● 192.168.0.38	[n/a]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]	[n/s]

Ready Display: All Threads: 0

10. The *IP address details* window will appear and list all the information that the *fetchers* collected from this system. The important thing to note in this window is the response time of the ping. This one is *0ms***,** seen in *item* **1,** and can give you an idea of the workstation's location on the network. The *hostname* field provides the name for the system, and the *Ports* field indicates the open ports, as seen in *item* **2**. The *MAC address* and *Vendor* fields, seen in *item* **3,** provide the physical address of the network card and the possible manufacturer. Finally, the *Filtered Ports* field, also seen in *item* **3,** will provide the list of ports that are monitored by a firewall or router.



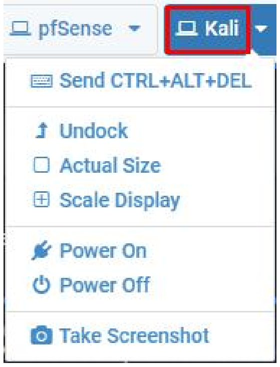
IP: 192.168.0.30
Ping: 0 ms
Hostname: osboxes.local
Ports: 80
TTL: 192
MAC Address: 00:50:56:9A:DE:74
MAC Vendor: VMware
Filtered Ports: 443,8080

11. As you can see, these seemingly simple scanning tools contain powerful information-gathering capabilities. We are now done with this exercise. Move on to the next section, where you will find some more advanced scanning tools.

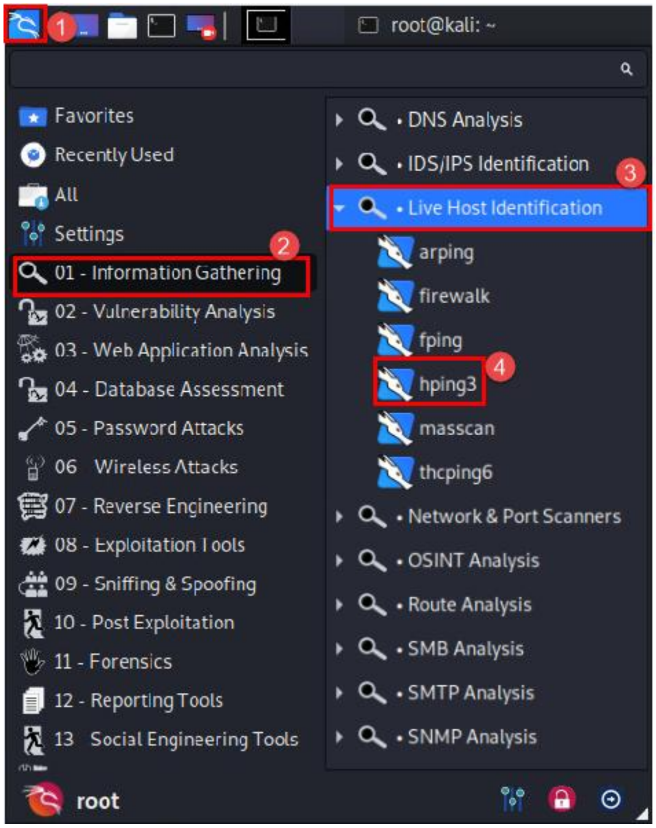
4 TCP and UDP Packet Crafting Using HPING3

As you continue to scan the network, we will now use the tool **HPING3**. This tool is a scriptable program that uses TCL language, and packets can be received and sent via binary or string representation describing the packets.

1. Switch to the **Kali Linux** virtual machine to access the graphical login screen.
   1.1. Log in as `root` using the password: `toor`, if prompted.



2. Let us start by launching **HPING3.** To do this, navigate to **Whisker Menu** > **Information Gathering** > **Live Host identification** > **hping3**, as seen in *items* **1**, **2**, **3,** and **4** below.

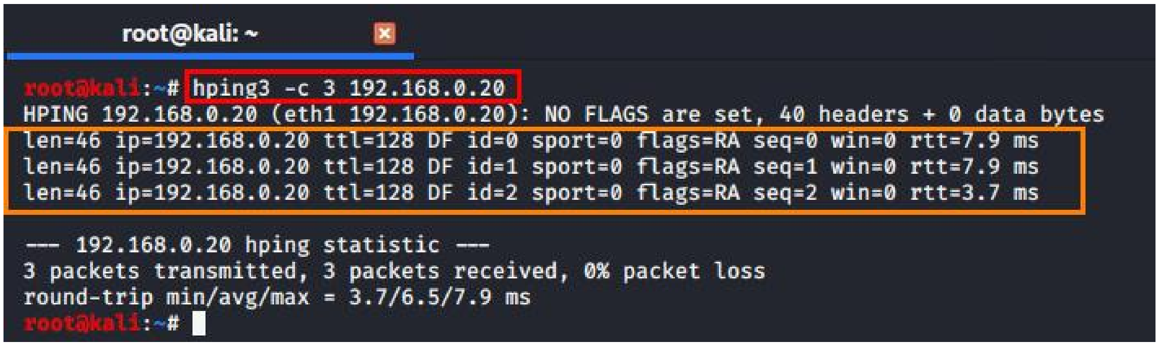


3. Now, type `hping3 -c 3 <IP address of the target machine>` and then press **Enter.** The target machine will be the *Windows* machine IP address.

```
root@kali:~# hping3 -c 3 192.168.0.20
```

> **-c** --count means stop after sending (and receiving) count response packets. So, -c 3 means that we will only want to send 3 packets to the target machine.

4. The output from the above command should have three packets sent and received, as seen below.



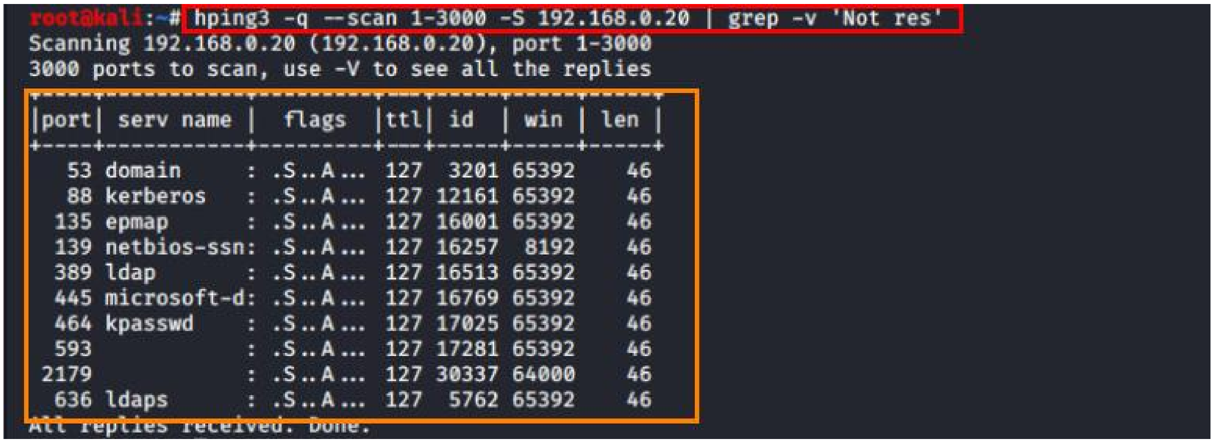
```
root@kali:~# hping3 -c 3 192.168.0.20
HPING 192.168.0.20 (eth1 192.168.0.20): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.0.20 ttl=128 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=1 sport=0 flags=RA seq=1 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=2 sport=0 flags=RA seq=2 win=0 rtt=3.7 ms

--- 192.168.0.20 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.7/6.5/7.9 ms
root@kali:~#
```

5. Now, let us scan the target machine, type `hping3 -q --scan 1-3000 -S 192.168.0.20 | grep -v ‘Not res’`. The results can be overwhelming; as such, we will pipe a grep function into the string using **-v --invert-match** to select the non-matching lines **‘Not res’;** this will only display the active replies.

```
root@kali:~# hping3 -q --scan 1-3000 -S 192.168.0.20 | grep -v ‘Not res’
```

> **--scan** parameter defines the port range to scan and the **-S** represents **SYN** flag.

6. The output from the above command should have sent and received a packet similar to the example shown below.

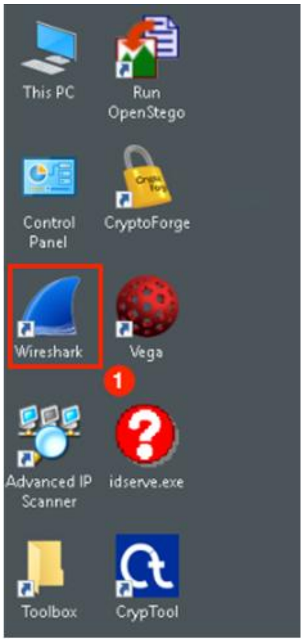


```
root@kali:~# hping3 -q --scan 1-3000 -S 192.168.0.20 | grep -v 'Not res'
Scanning 192.168.0.20 (192.168.0.20), port 1-3000
3000 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
   53 domain     : .S..A... 127  3201 65392    46
   88 kerberos   : .S..A... 127 12161 65392    46
  135 epmap      : .S..A... 127 16001 65392    46
  139 netbios-ssn: .S..A... 127 16257  8192    46
  389 ldap       : .S..A... 127 16513 65392    46
  445 microsoft-d: .S..A... 127 16769 65392    46
  464 kpasswd    : .S..A... 127 17025 65392    46
  593            : .S..A... 127 17281 65392    46
 2179            : .S..A... 127 30337 64000    46
  636 ldaps      : .S..A... 127  5762 65392    46
All replies received. Done.
```

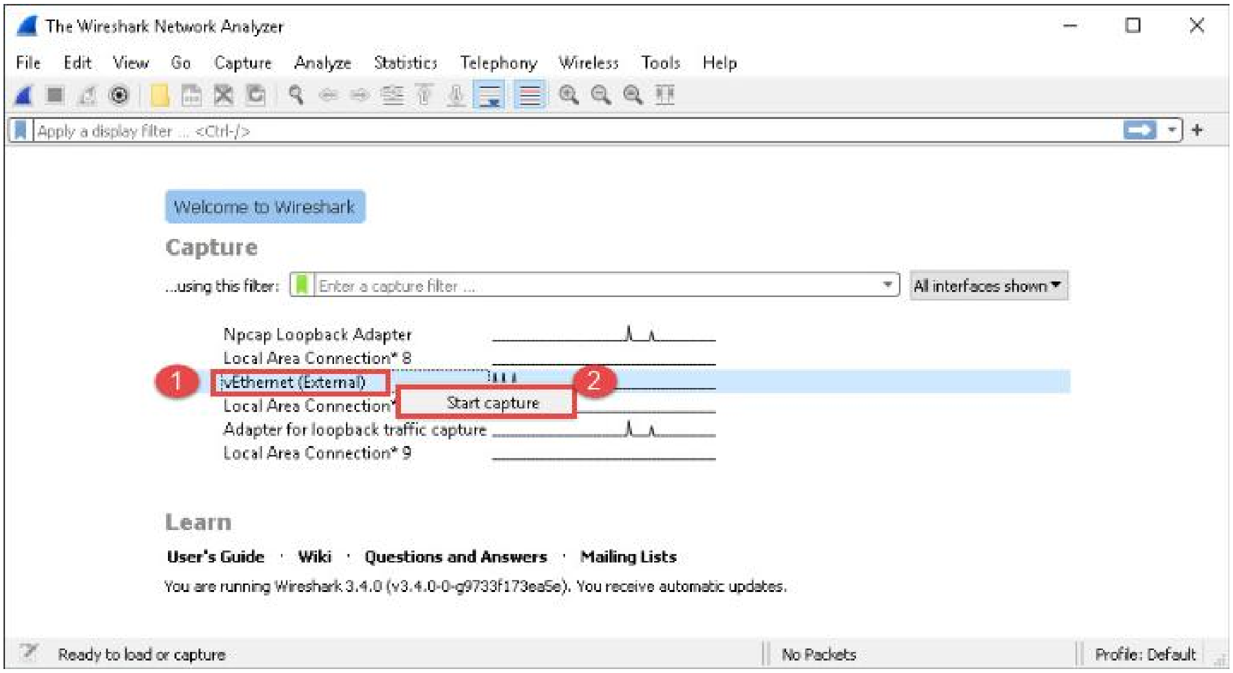
7. Now that we a familiar with hping3, let's perform UDP packet crafting. Packet crafting is a technique that allows the Pen Tester to probe firewall rule sets and find entry points into a targeted system or network. This is done by manually generating packets to test the network devices instead of using existing network traffic.
8. Type **`hping3 192.168.0.20 --udp --rand-source --data 500`.**

```
root@kali:~# hping3 192.168.0.20 --udp --rand-source --data 500
```

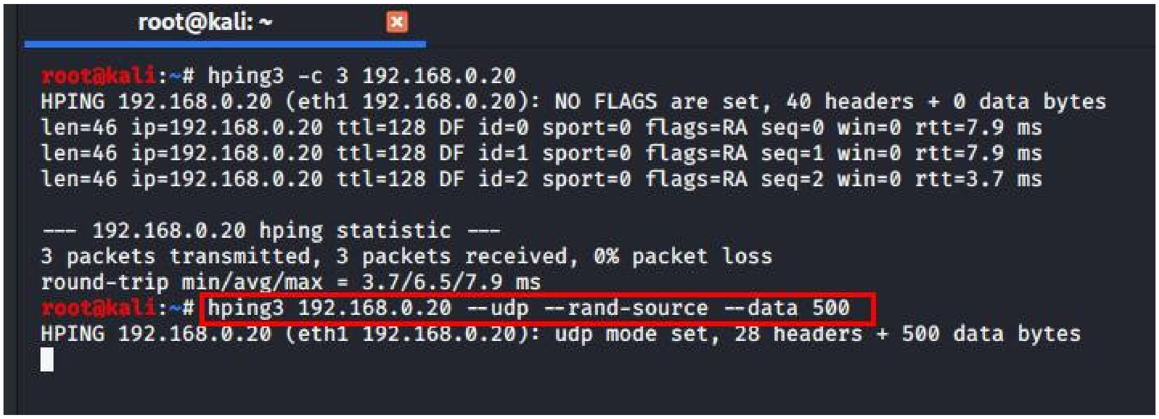
9. Now, log into Windows VM, if not already logged in, and launch Wireshark to start capturing the packets. Double-click **Wireshark** on the Desktop to start the software, as seen in *item* **1** below.



10. Once Wireshark has started, select **vEthernet (External),** then right-click and choose **Start capture** from the welcome screen, as seen in *items* **1** and **2** below.

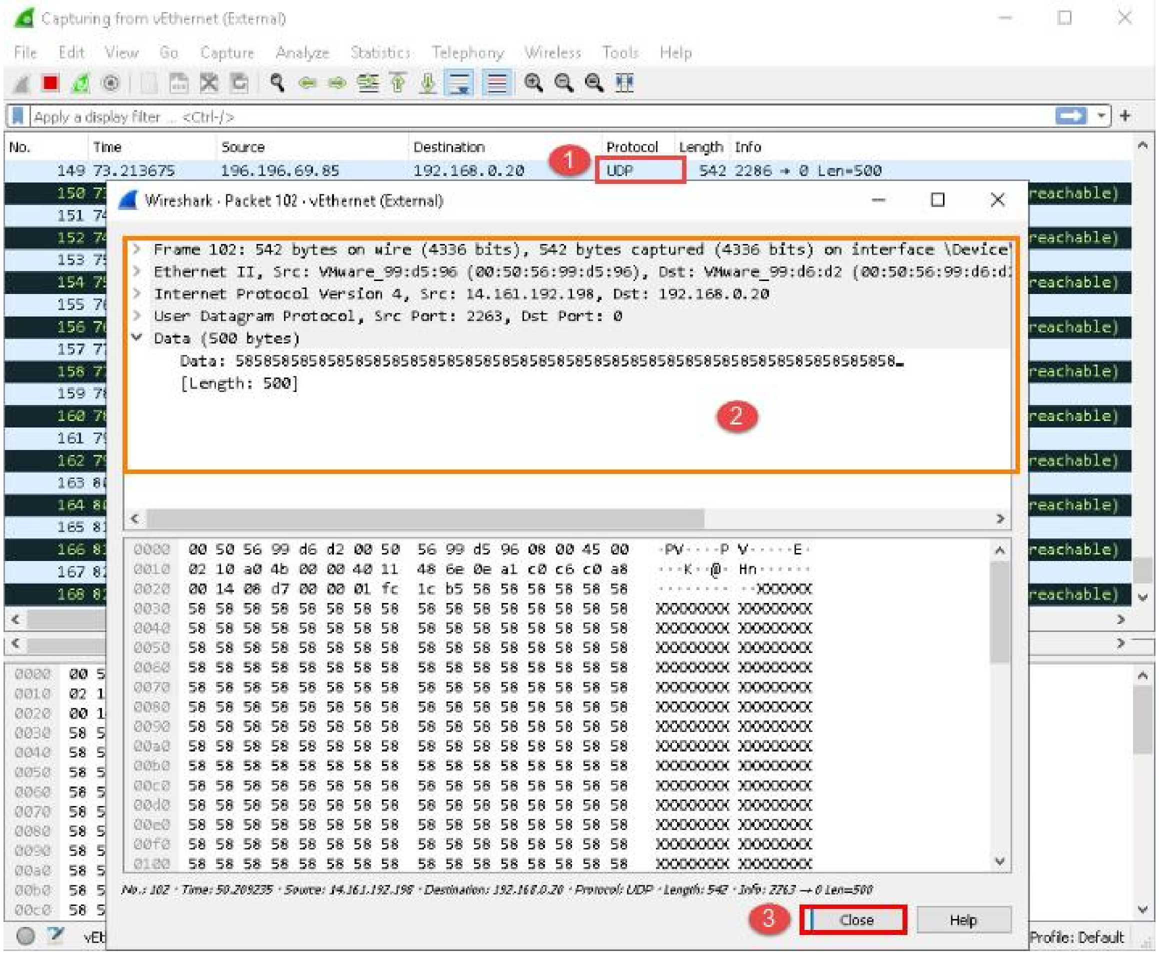

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Npcap Loopback Adapter
Local Area Connection* 8
vEthernet (External)
Local Area Connection*
Adapter for loopback traffic capture
Local Area Connection* 9
Start capture
1
2
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
You are running Wireshark 3.4.0 (v3.4.0-0-g9733f173ea5e). You receive automatic updates.
Ready to load or capture
No Packets
Profile: Default

11. Switch to Kali Linux and execute the previously typed command. If the terminal was closed, open it again and retype the command at step 9 above.



```
root@kali: ~
root@kali:~# hping3 -c 3 192.168.0.20
HPING 192.168.0.20 (eth1 192.168.0.20): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.0.20 ttl=128 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=1 sport=0 flags=RA seq=1 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=2 sport=0 flags=RA seq=2 win=0 rtt=3.7 ms

--- 192.168.0.20 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.7/6.5/7.9 ms
root@kali:~# hping3 192.168.0.20 --udp --rand-source --data 500
HPING 192.168.0.20 (eth1 192.168.0.20): udp mode set, 28 headers + 500 data bytes
```

12. The output from the above command will be seen on the target machine. Switch back to the **WinOS** machine to observe the UDP packets in Wireshark. Double-click any UDP packet and observe the details as seen in *items* **1** and **2**. After viewing the packet, click **Close,** as seen in *item* **3** below.


Capturing from vEthernet (External)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Wireshark · Packet 102 · vEthernet (External)
1
2
3
Close
Help
Profile: Default

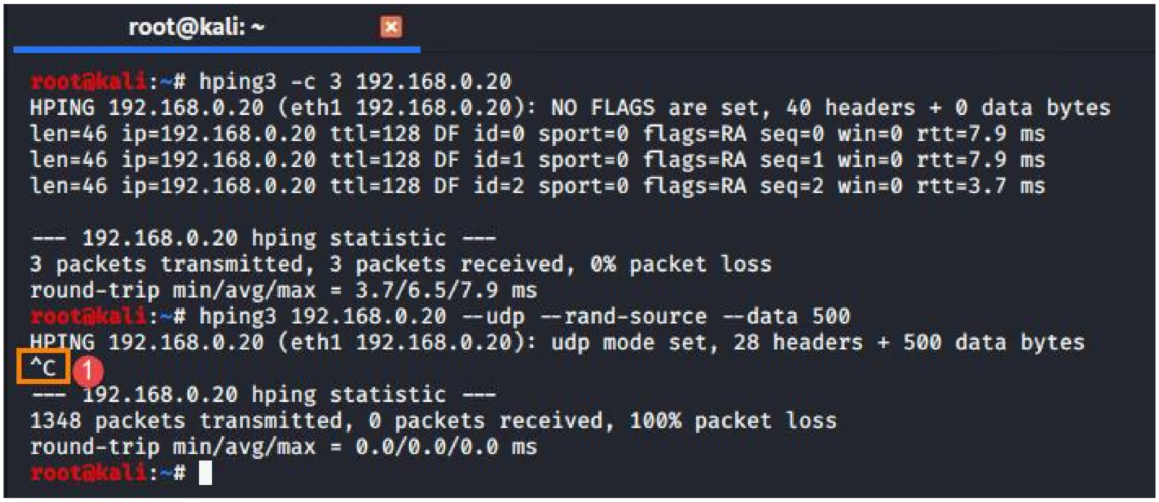
```
No.   Time        Source          Destination    Protocol  Length  Info
  149 73.213675   196.196.69.85   192.168.0.20   UDP          542  2286 → 0 Len=500
```

```
> Frame 102: 542 bytes on wire (4336 bits), 542 bytes captured (4336 bits) on interface \Device
> Ethernet II, Src: VMware_99:d5:96 (00:50:56:99:d5:96), Dst: VMware_99:d6:d2 (00:50:56:99:d6:d
> Internet Protocol Version 4, Src: 14.161.192.198, Dst: 192.168.0.20
> User Datagram Protocol, Src Port: 2263, Dst Port: 0
v Data (500 bytes)
    Data: 5858585858585858585858585858585858585858585858585858585858585858585858…
    [Length: 500]
```

```
0000  00 50 56 99 d6 d2 00 50  56 99 d5 96 08 00 45 00   ·PV····P V·····E·
0010  02 10 a0 4b 00 00 40 11  48 6e 0e a1 c0 c6 c0 a8   ···K··@· Hn······
0020  00 14 08 d7 00 00 01 fc  1c b5 58 58 58 58 58 58   ········ ··XXXXXX
0030  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0080  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0090  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00a0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00b0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00c0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00d0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00e0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
00f0  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
0100  58 58 58 58 58 58 58 58  58 58 58 58 58 58 58 58   XXXXXXXX XXXXXXXX
```

No.: 102 · Time: 50.209235 · Source: 14.161.192.198 · Destination: 192.168.0.20 · Protocol: UDP · Length: 542 · Info: 2263 → 0 Len=500

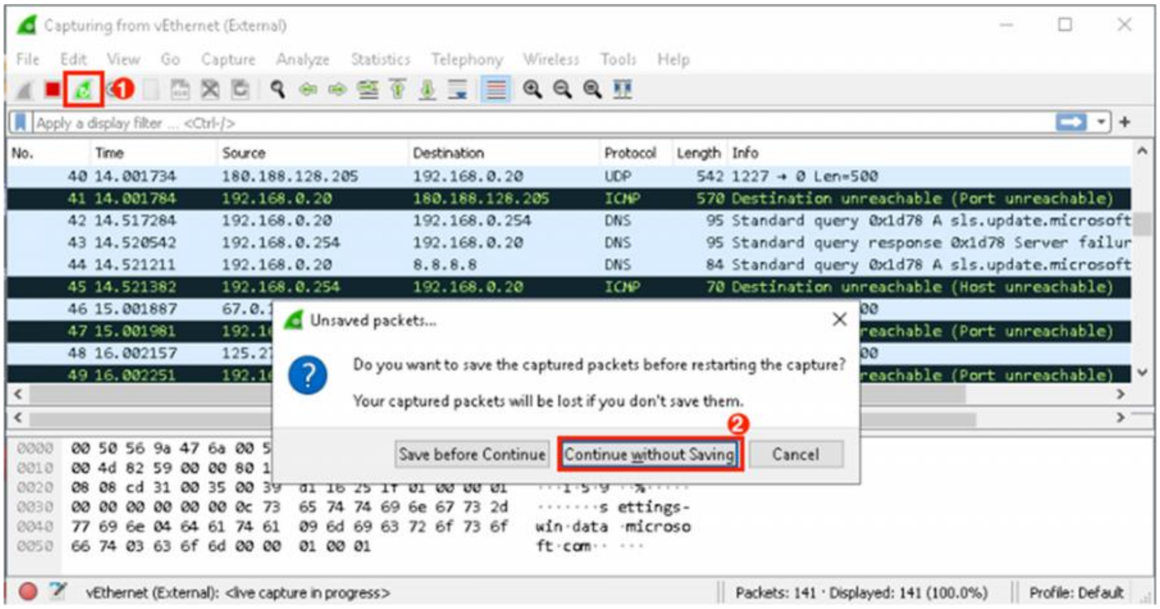
13. The UDP packets are still being sent to the target machine. Let us stop this before proceeding to the next task. Change focus to the **Kali** machine and press **Ctrl+C** to stop the command, as seen in *item* **1** below.



```
root@kali:~# hping3 -c 3 192.168.0.20
HPING 192.168.0.20 (eth1 192.168.0.20): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.0.20 ttl=128 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=1 sport=0 flags=RA seq=1 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=2 sport=0 flags=RA seq=2 win=0 rtt=3.7 ms

--- 192.168.0.20 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.7/6.5/7.9 ms
root@kali:~# hping3 192.168.0.20 --udp --rand-source --data 500
HPING 192.168.0.20 (eth1 192.168.0.20): udp mode set, 28 headers + 500 data bytes
^C
--- 192.168.0.20 hping statistic ---
1348 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# 
```

If this is not done before starting the next task, Wireshark will continue to capture the sent UDP packets from the Kali Linux machine.

14. Let us move on to the next task. Switch to the **WinOS** machine and browse to the Wireshark navigation panel and select the **Restart current capture** icon. When prompted to save, click **Continue without Saving** as seen in *items* **1** and **2** below.


Capturing from vEthernet (External)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
1
Apply a display filter ... <Ctrl-/>
Unsaved packets...
2
Save before Continue
Continue without Saving
Cancel

```
No.    Time       Source           Destination      Protocol Length Info
    40 14.001734  180.188.128.205  192.168.0.20     UDP       542 1227 → 0 Len=500
    41 14.001784  192.168.0.20     180.188.128.205  ICMP      570 Destination unreachable (Port unreachable)
    42 14.517284  192.168.0.20     192.168.0.254    DNS        95 Standard query 0x1d78 A sls.update.microsoft
    43 14.520542  192.168.0.254    192.168.0.20     DNS        95 Standard query response 0x1d78 Server failur
    44 14.521211  192.168.0.20     8.8.8.8          DNS        84 Standard query 0x1d78 A sls.update.microsoft
    45 14.521382  192.168.0.254    192.168.0.20     ICMP       70 Destination unreachable (Host unreachable)
    46 15.001887  67.0.1                                                                00
    47 15.001981  192.1                                                                 reachable (Port unreachable)
    48 16.002157  125.2                                                                 00
    49 16.002251  192.1                                                                 reachable (Port unreachable)

0000  00 50 56 9a 47 6a 00 5
0010  00 4d 82 59 00 00 80 1
0020  08 08 cd 31 00 35 00 39  d1 16 25 1f 01 00 00 01   ···1·5·9 ··%·····
0030  00 00 00 00 00 00 0c 73  65 74 74 69 6e 67 73 2d   ·······s ettings-
0040  77 69 6e 04 64 61 74 61  09 6d 69 63 72 6f 73 6f   win·data ·microso
0050  66 74 03 63 6f 6d 00 00  01 00 01                  ft·com·· ···
```

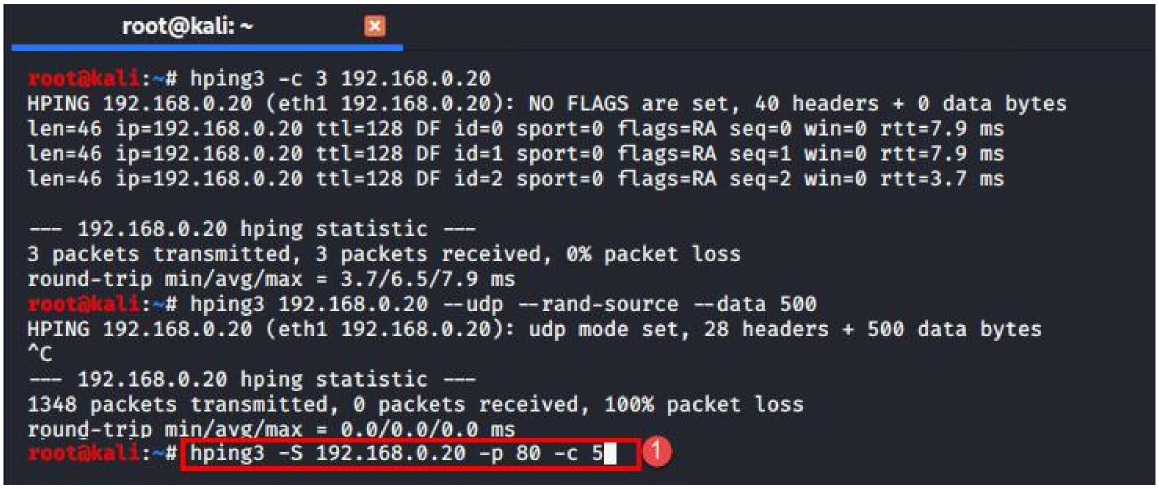
Do you want to save the captured packets before restarting the capture?

Your captured packets will be lost if you don't save them.

vEthernet (External): <live capture in progress> | Packets: 141 · Displayed: 141 (100.0%) | Profile: Default

15. Next, we will be sending a **TCP SYN Request** to the target machine. To start, go back to the **Kali** machine, type `hping3 -S 192.168.0.20 -p 80 -c 5` and press **Enter,** as seen in *item* **1** below**.**

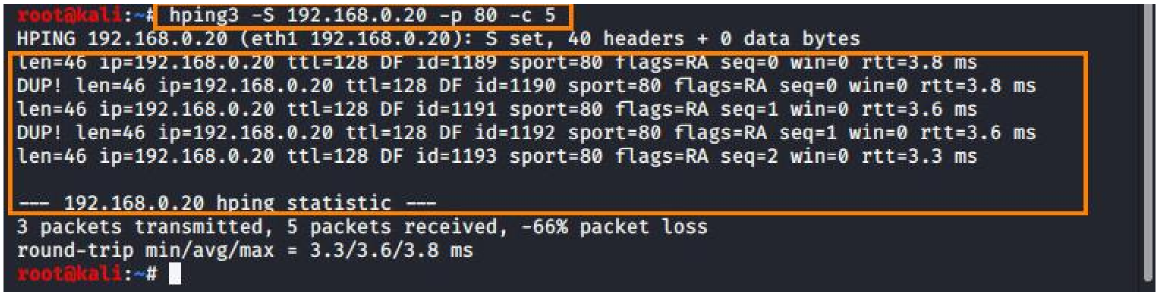
```
root@kali:~# hping3 -S 192.168.0.20 -p 80 -c 5
```



```
root@kali:~# hping3 -c 3 192.168.0.20
HPING 192.168.0.20 (eth1 192.168.0.20): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.0.20 ttl=128 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=1 sport=0 flags=RA seq=1 win=0 rtt=7.9 ms
len=46 ip=192.168.0.20 ttl=128 DF id=2 sport=0 flags=RA seq=2 win=0 rtt=3.7 ms

--- 192.168.0.20 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.7/6.5/7.9 ms
root@kali:~# hping3 192.168.0.20 --udp --rand-source --data 500
HPING 192.168.0.20 (eth1 192.168.0.20): udp mode set, 28 headers + 500 data bytes
^C
--- 192.168.0.20 hping statistic ---
1348 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# hping3 -S 192.168.0.20 -p 80 -c 5
```

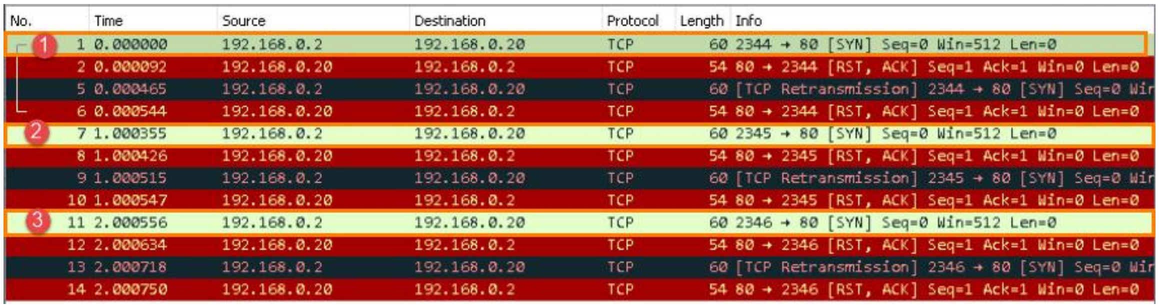
-S will perform a **TCP SYN request** on the target machine **-p** will pass the traffic through which port is assigned, and **-c** is the count of the packets being sent.

16. The output from the above command shows that five TCP packets were sent through port 80 to the target machine.



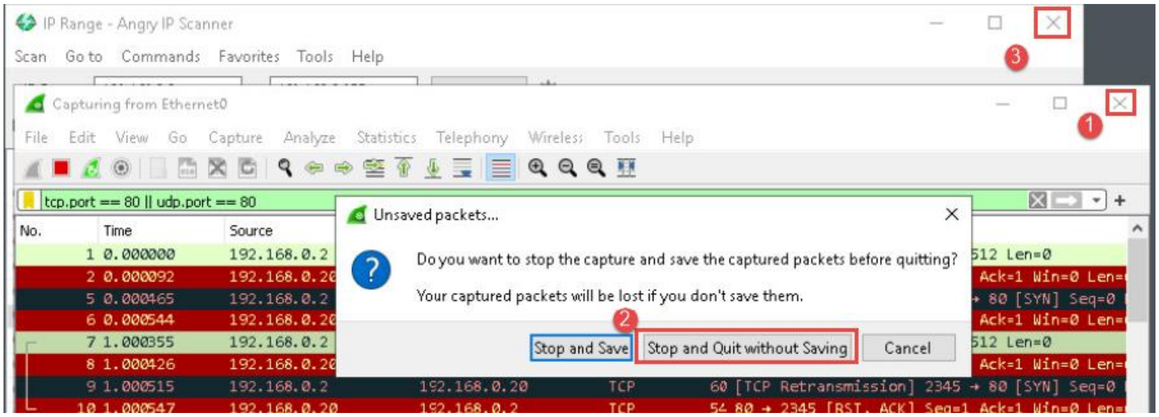
```
root@kali:~# hping3 -S 192.168.0.20 -p 80 -c 5
HPING 192.168.0.20 (eth1 192.168.0.20): S set, 40 headers + 0 data bytes
len=46 ip=192.168.0.20 ttl=128 DF id=1189 sport=80 flags=RA seq=0 win=0 rtt=3.8 ms
DUP! len=46 ip=192.168.0.20 ttl=128 DF id=1190 sport=80 flags=RA seq=0 win=0 rtt=3.8 ms
len=46 ip=192.168.0.20 ttl=128 DF id=1191 sport=80 flags=RA seq=1 win=0 rtt=3.6 ms
DUP! len=46 ip=192.168.0.20 ttl=128 DF id=1192 sport=80 flags=RA seq=1 win=0 rtt=3.6 ms
len=46 ip=192.168.0.20 ttl=128 DF id=1193 sport=80 flags=RA seq=2 win=0 rtt=3.3 ms

--- 192.168.0.20 hping statistic ---
3 packets transmitted, 5 packets received, -66% packet loss
round-trip min/avg/max = 3.3/3.6/3.8 ms
root@kali:~#
```

17. Let us confirm the receipt of the packets on the target machine. The output from the above command shows that three TCP packets were sent through port 80 to the target machine, as seen in *items* **1**, **2,** and **3** below.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.2	192.168.0.20	TCP	60	2344 → 80 [SYN] Seq=0 Win=512 Len=0
2	0.000092	192.168.0.20	192.168.0.2	TCP	54	80 → 2344 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
5	0.000465	192.168.0.2	192.168.0.20	TCP	60	[TCP Retransmission] 2344 → 80 [SYN] Seq=0 Wir
6	0.000544	192.168.0.20	192.168.0.2	TCP	54	80 → 2344 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
7	1.000355	192.168.0.2	192.168.0.20	TCP	60	2345 → 80 [SYN] Seq=0 Win=512 Len=0
8	1.000426	192.168.0.20	192.168.0.2	TCP	54	80 → 2345 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
9	1.000515	192.168.0.2	192.168.0.20	TCP	60	[TCP Retransmission] 2345 → 80 [SYN] Seq=0 Wir
10	1.000547	192.168.0.20	192.168.0.2	TCP	54	80 → 2345 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
11	2.000556	192.168.0.2	192.168.0.20	TCP	60	2346 → 80 [SYN] Seq=0 Win=512 Len=0
12	2.000634	192.168.0.20	192.168.0.2	TCP	54	80 → 2346 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
13	2.000718	192.168.0.2	192.168.0.20	TCP	60	[TCP Retransmission] 2346 → 80 [SYN] Seq=0 Wir
14	2.000750	192.168.0.20	192.168.0.2	TCP	54	80 → 2346 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

18. Wireshark detects the TCP packets sent by the attacker's machine. This is the end of the exercise. Close all windows; if prompted to save, click **Stop and Quit without saving,** as seen in *items* **1**, **2,** and **3** below.


IP Range - Angry IP Scanner
Scan Go to Commands Favorites Tools Help
3
Capturing from Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
1
Unsaved packets...
2
Stop and Save
Stop and Quit without Saving
Cancel

```
tcp.port == 80 || udp.port == 80
No.   Time       Source
  1 0.000000   192.168.0.2                                          512 Len=0
  2 0.000092   192.168.0.20                                         Ack=1 Win=0 Len=
  5 0.000465   192.168.0.2                                          → 80 [SYN] Seq=0
  6 0.000544   192.168.0.20                                         Ack=1 Win=0 Len=
  7 1.000355   192.168.0.2                                          512 Len=0
  8 1.000426   192.168.0.20                                         Ack=1 Win=0 Len=
  9 1.000515   192.168.0.2    192.168.0.20   TCP   60 [TCP Retransmission] 2345 → 80 [SYN] Seq=0
 10 1.000547   192.168.0.20   192.168.0.2    TCP   54 80 → 2345 [RST, ACK] Seq=1 Ack=1 Win=0 Len=
```

Do you want to stop the capture and save the captured packets before quitting?

Your captured packets will be lost if you don't save them.

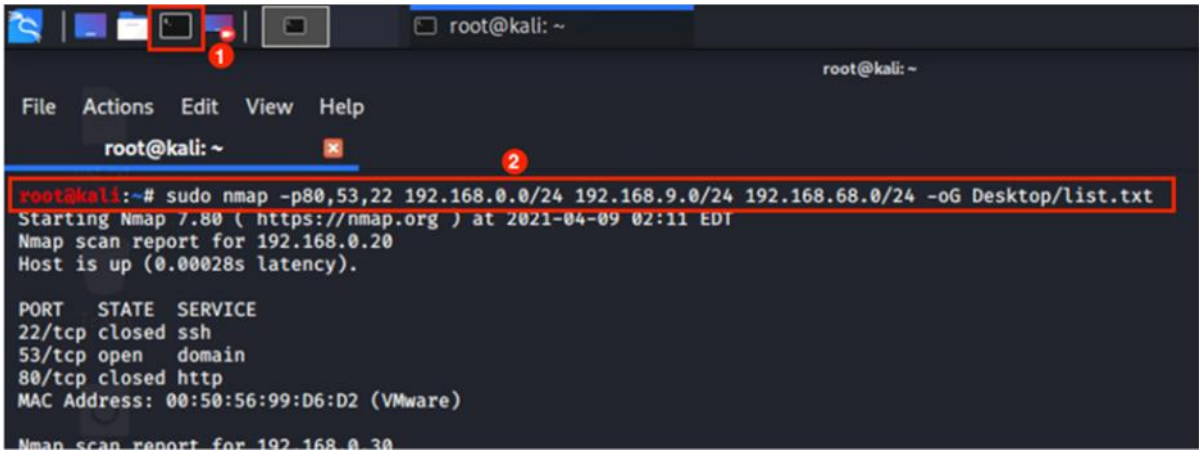
19. Switch to **Kali** and move on to the next lab.

5 Vulnerability Scanning with Nikto

We will now perform a vulnerability scan to see what kind of vulnerabilities we can identify on the hosts on this network. Let us begin by using *Nmap* to create a list of hosts that we can feed to *Nikto.*

1. To begin, open a **Terminal** window by clicking the icon from the navigator panel, as seen in *item* **1.** Once there, type `sudo nmap -p80,53,22 192.168.0.0/24 192.168.9.0/24 192.168.68.0/24 -oG Desktop/list.txt` and press **Enter.** This command will output the results of the scan to a text file called *list.txt* on the Desktop.

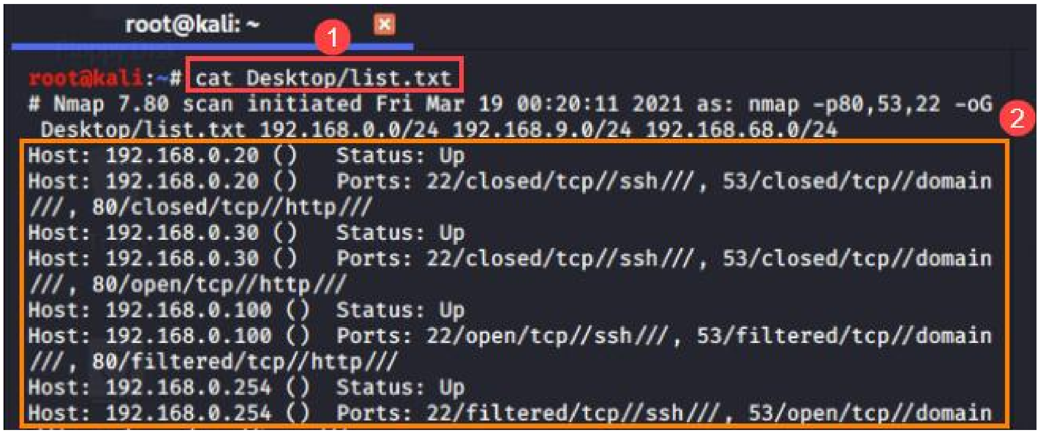
```
root@kali:~# sudo nmap -p80,53,22 192.168.0.0/24 192.168.9.0/24
192.168.68.0/24 -oG Desktop/list.txt
```



```
root@kali:~# sudo nmap -p80,53,22 192.168.0.0/24 192.168.9.0/24 192.168.68.0/24 -oG Desktop/list.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-09 02:11 EDT
Nmap scan report for 192.168.0.20
Host is up (0.00028s latency).

PORT   STATE  SERVICE
22/tcp closed ssh
53/tcp open   domain
80/tcp closed http
MAC Address: 00:50:56:99:D6:D2 (VMware)

Nmap scan report for 192.168.0.30
```

2. Now that the scan is complete, we can parse the list to prepare it to be ingested by *Nikto.* Type the command `cat Desktop/list.txt` and press **Enter** as seen in *item* **1.** You will see what the list in greppable format looks like, as shown in *item* **2.**

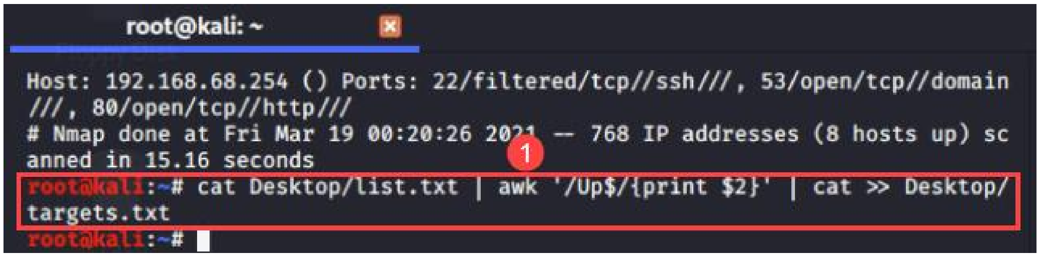
```
root@kali:~# cat Desktop/list.txt
```



```
root@kali:~# cat Desktop/list.txt
# Nmap 7.80 scan initiated Fri Mar 19 00:20:11 2021 as: nmap -p80,53,22 -oG
 Desktop/list.txt 192.168.0.0/24 192.168.9.0/24 192.168.68.0/24
Host: 192.168.0.20 ()   Status: Up
Host: 192.168.0.20 ()   Ports: 22/closed/tcp//ssh///, 53/closed/tcp//domain
///, 80/closed/tcp//http///
Host: 192.168.0.30 ()   Status: Up
Host: 192.168.0.30 ()   Ports: 22/closed/tcp//ssh///, 53/closed/tcp//domain
///, 80/open/tcp//http///
Host: 192.168.0.100 ()  Status: Up
Host: 192.168.0.100 ()  Ports: 22/open/tcp//ssh///, 53/filtered/tcp//domain
///, 80/filtered/tcp//http///
Host: 192.168.0.254 ()  Status: Up
Host: 192.168.0.254 ()  Ports: 22/filtered/tcp//ssh///, 53/open/tcp//domain
```

3. Next, enter the following command to remove the unwanted data `cat Desktop/list.txt | awk '/Up$/{print $2}' | cat >> Desktop/targets.txt` and press **Enter** as seen in *item* **1.** This will parse only the IP addresses of the hosts that are up and save the results to a file on the Desktop called *targets.txt.*

```
root@kali:~# cat Desktop/list.txt | awk '/Up$/{print $2}' | cat >>
Desktop/targets.txt
```

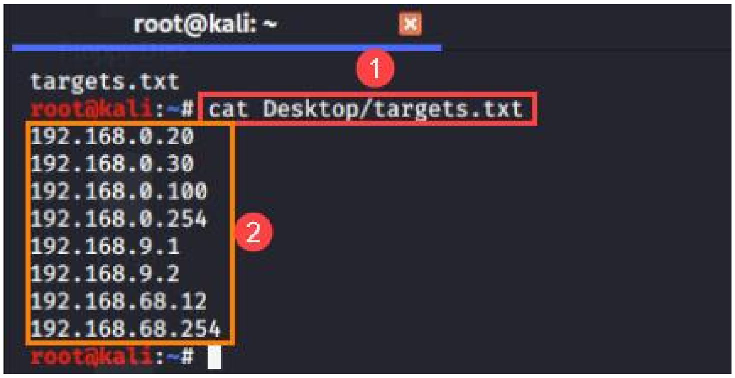


```
Host: 192.168.68.254 () Ports: 22/filtered/tcp//ssh///, 53/open/tcp//domain
///, 80/open/tcp//http///
# Nmap done at Fri Mar 19 00:20:26 2021 -- 768 IP addresses (8 hosts up) sc
anned in 15.16 seconds
root@kali:~# cat Desktop/list.txt | awk '/Up$/{print $2}' | cat >> Desktop/
targets.txt
root@kali:~# 
```

awk is an awesome programming language that scans the input (**list.txt**) for lines that match any of the set patterns, in this case the machine Status (**Up**)

4. Now, type `cat Desktop/targets.txt` and press **Enter** as seen in *item* **1,** to see what the output looks like. As you can see in *item* **2,** the list is significantly condensed to just the IP addresses from the previous file.

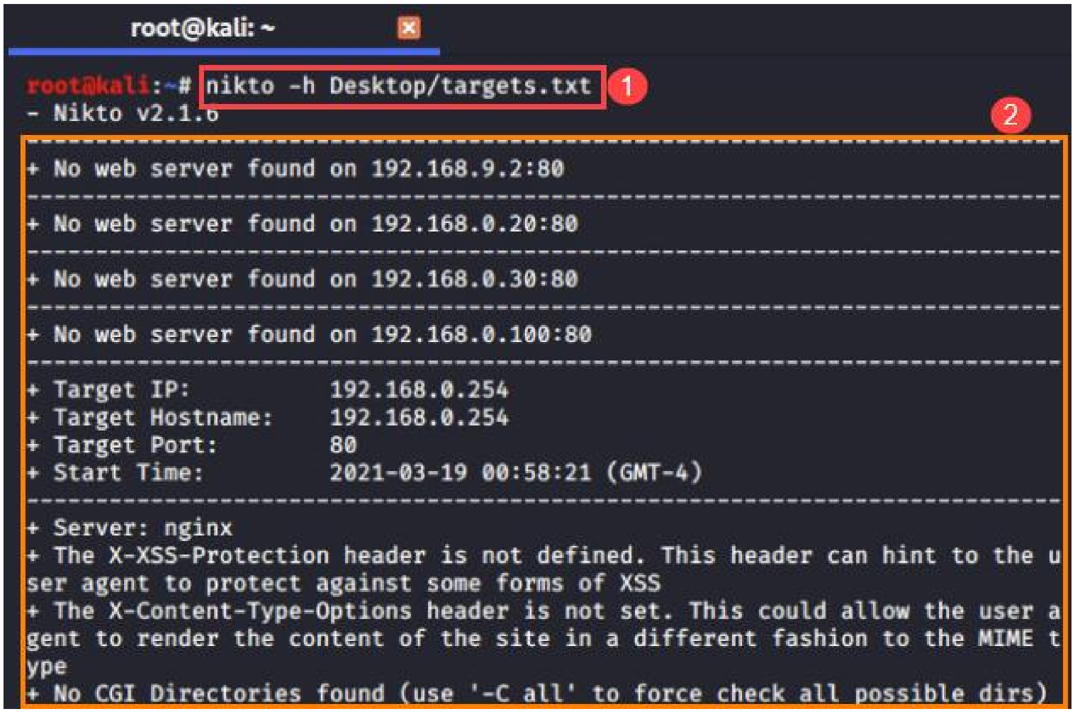
```
root@kali:~# cat Desktop/targets.txt
```



```
root@kali: ~
targets.txt
root@kali:~# cat Desktop/targets.txt
192.168.0.20
192.168.0.30
192.168.0.100
192.168.0.254
192.168.9.1
192.168.9.2
192.168.68.12
192.168.68.254
root@kali:~#
```

5. Next, we will use *Nikto* to scan the hosts in our *targets.txt* file. To do this, type the command `nikto -h Desktop/targets` and press **Enter** as seen in *item* **1.** This scan will take some time to complete. As it progresses, you will see the results populate the terminal window, as seen in *item* **2.**

```
root@kali:~# nikto -h Desktop/targets.txt
```



```
root@kali: ~
root@kali:~# nikto -h Desktop/targets.txt
- Nikto v2.1.6
---------------------------------------------------------------------------
+ No web server found on 192.168.9.2:80
---------------------------------------------------------------------------
+ No web server found on 192.168.0.20:80
---------------------------------------------------------------------------
+ No web server found on 192.168.0.30:80
---------------------------------------------------------------------------
+ No web server found on 192.168.0.100:80
---------------------------------------------------------------------------
+ Target IP:          192.168.0.254
+ Target Hostname:    192.168.0.254
+ Target Port:        80
+ Start Time:         2021-03-19 00:58:21 (GMT-4)
---------------------------------------------------------------------------
+ Server: nginx
+ The X-XSS-Protection header is not defined. This header can hint to the u
ser agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user a
gent to render the content of the site in a different fashion to the MIME t
ype
+ No CGI Directories found (use '-C all' to force check all possible dirs)
```

6. The results will be lengthy, but you can scroll through them to see what kind of vulnerabilities it found for the associated IP addresses. These vulnerabilities will be useful to note when moving to the next stages of the penetration test.

> The scan was limited to ports **80**, **53** and **22,** however; you may add or substitute other common ports such as **20**, **21**, **23**, **25**, **110**, and **443** and run the scan again and observe the results.

7. This is the end of the lab; please close all windows and terminals to complete the lab.