



CySA+ Lab Series

Lab 01: Network Enumeration

Document Version: 2022-10-10

Material in this Lab Aligns to the Following	
CompTIA CySA+ (CS0-002) Exam Objectives	1.2 - Given a scenario, utilize threat intelligence to support organizational security 1.3 - Given a scenario, perform vulnerability management activities 1.4 - Given a scenario, analyze the output from vulnerability tools 1.7 - Given a scenario, implement controls to mitigate attacks and software vulnerabilities
All-In-One CompTIA CySA+ Second Edition ISBN-13: 978-1260464306 Chapters	2: Threat Intelligence in Support of Organizational Security 3: Vulnerability Management Activities 4: Vulnerability Assessment Tools 7: Mitigating Controls for Attacks and Software Vulnerabilities

Copyright © 2022 Network Development Group, Inc.
www.netdevgroup.com

NETLAB+ is a registered trademark of Network Development Group, Inc.
KALI LINUX™ is a trademark of Offensive Security.
ALIEN VAULT OSSIM V is a trademark of AlienVault, Inc.
Microsoft®, Windows®, and Windows Server® are trademarks of the Microsoft group of companies.
Greenbone is a trademark of Greenbone Networks GmbH.
SECURITY ONION is a trademark of Security Onion Solutions LLC.
Android is a trademark of Google LLC.
pfSense® is a registered mark owned by Electric Sheep Fencing LLC ("ESF").
All trademarks, logos, and brand names are the property of their respective owners.

Contents

Introduction	3
Objective	3
Lab Topology.....	4
Lab Settings.....	5
1 Utilizing Netstat to Perform System Scans	6
2 Performing Network Scans with nmap.....	14
2.1 Basic nmap Features/Functions and Scanning from Internal Hosts	14
2.2 Perform OS Fingerprinting from nmap	27
2.3 Using nmap to Scan Hosts from an External Network.....	32
2.4 Exporting Scan Logs from nmap.....	39
3 Performing Host Scans with Legion	44

Introduction

This lab will explore various network enumeration and reconnaissance scanning methods. This lab will introduce several common tools, *netstat* (monitor connections on a network), *nmap* (perform port scanning), and *legion* (a GUI network host and port scanner).

In addition, you will see how a firewall can protect resources on the network by blocking scans.

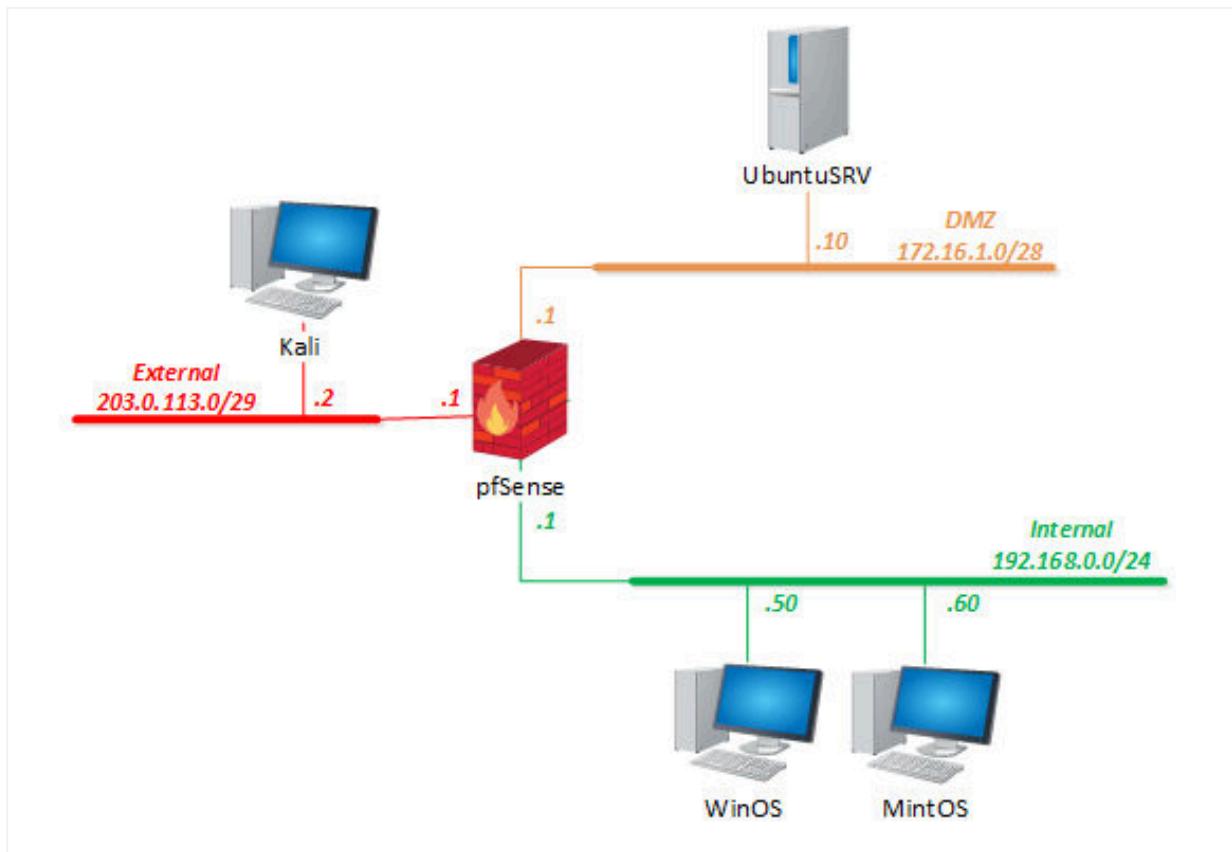
Objective

In this lab, you will be conducting network and port scans using various tools. You will be performing the following tasks:

- Perform Host Ports Scans with *NETSTAT*
- Perform Network Scans with *NMAP*
- Perform OS Fingerprinting with *NMAP*
- Perform Host Scans with *LEGION*

You will also be looking at how firewalls can protect resources on the network by conducting scans with and without firewalls.

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	Account	Password
WinOS (Server 2019)	192.168.0.50	Administrator	NDGlabpass123!
MintOS (Linux Mint)	192.168.0.60	sysadmin	NDGlabpass123!
OSSIM (Alien Vault)	172.16.1.2	root	NDGlabpass123!
UbuntuSRV (Ubuntu Server)	172.16.1.10	sysadmin	NDGlabpass123!
Kali	203.0.113.2	sysadmin	NDGlabpass123!
pfSense	203.0.113.1 172.16.1.1 192.168.0.1	admin	NDGlabpass123!

1 Utilizing Netstat to Perform System Scans

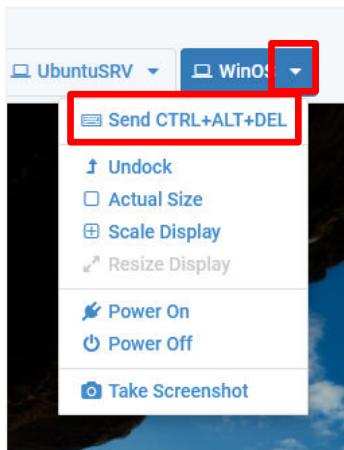
The *netstat* (network statistics) command-line utility is used for displaying network connections in TCP/IP networks. It can display network connections for routing tables, network interfaces, transport layer protocols, and multicast memberships. In addition, security analysts can use *netstat* to discover masquerade connections, sockets, and listening states. The tool was originally a Unix command, but has been ported to Linux, Windows, and macOS. *netstat* will only display the information on the local computer where you run the program.

In this task, you will use *netstat* to scan your machine to discover which ports are open, and which processes with their respective Process Identifiers (PIDs) are listening on those ports.

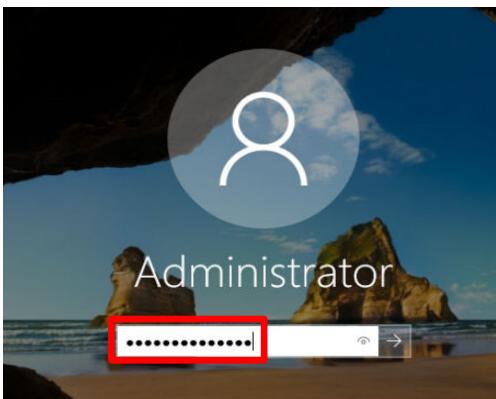


Process Identifiers or PIDs are numbers that are used in Unix/Linux, Windows, and MacOS (also Android and iOS, but not as easy to show) to identify an active process. In this way, multiple instances of a program can run concurrently, each one running with its own PID.

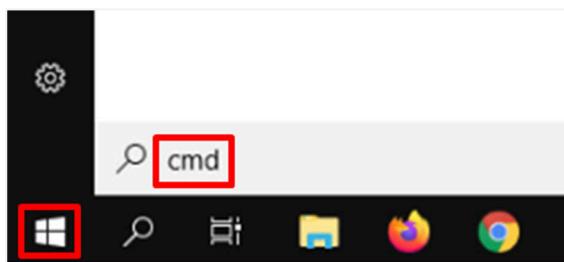
1. Change the focus to the **WinOS** host.
2. Bring up the login window by sending a Ctrl + Alt + Delete. To do this, click the **WinOS** dropdown menu and click **Send CTRL+ALT+DEL**.



3. Log in as **Administrator** using the password: **NDGLabpass123!**



4. Click on the **Windows Start** button in the bottom-left corner, type **cmd** and then press **Enter** to bring up the command prompt window.



5. In the command prompt window, enter the following *netstat* command to check which *TCP* and *UDP* ports are *LISTENING* for a connection and which have an *ESTABLISHED* connection. A port that is *LISTENING* is a potential “open door” that can be exploited.

```
netstat -a | more
```

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:445	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:5985	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:47001	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49664	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49665	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49666	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49667	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49668	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49669	WIN-E3AIDIHECNG:0	LISTENING
TCP	192.168.0.50:139	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:135	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:445	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:5985	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:47001	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49664	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49665	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49666	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49667	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49668	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49669	WIN-E3AIDIHECNG:0	LISTENING
UDP	0.0.0.0:123	*:*	
UDP	0.0.0.0:5353	*:*	
UDP	0.0.0.0:5355	*:*	
UDP	0.0.0.0:53725	*:*	

-- More --



The **-a** option displays all connections and listening ports.

The **| more** (reads as “pipe to more”) is a good way of displaying output one screen at a time. If the output is more than one screen’s worth, the output will pause. Pressing the Spacebar will continue the output to the next screen.

The first page of the output displayed above shows that *TCP Port 139* is *LISTENING*. This port is used for the *NetBIOS Session Service*. This protocol is used for *Windows File and Print Sharing*. Since there are a number of vulnerabilities associated with the implementation of the protocol, a security analyst needs to be aware of the risks of leaving this port open.

6. If the **-- More --** is shown on the last line of the page, press the **Spacebar** to show the next page.

```

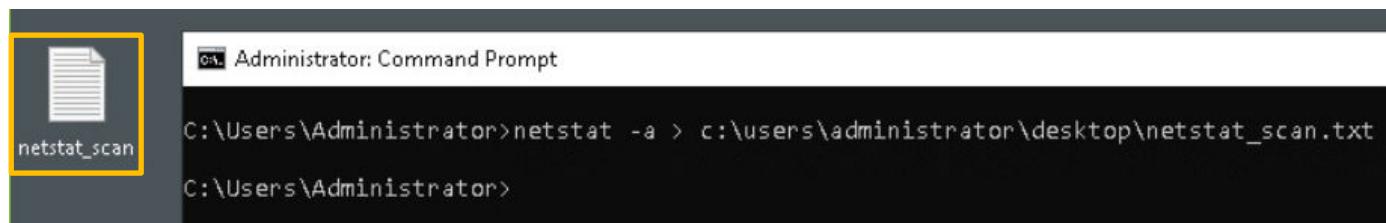
UDP      127.0.0.1:53563      *:*
UDP      192.168.0.50:137      *:*
UDP      192.168.0.50:138      *:*
UDP      [::]:123              *:*
UDP      [::]:5353              *:*
UDP      [::]:5355              *:*
UDP      [::]:53725             *:*

```

Notice that for *UDP*, *Port 137*, *NetBIOS Name Service* and *Port 138*, *NetBIOS Datagram Service* is shown in the list indicating that the port is open, but they do not show the *LISTENING* status. This is because *TCP* is a connection-oriented protocol that needs to establish a connection after the *Three-Way Handshake* (*SYN*, *SYN/ACK*, *ACK*), whereas *UDP* is connectionless and just opens the port and waits for datagrams to arrive. These open ports are also vulnerable and can be exploited.

7. To capture the output of the *netstat* scan, use the redirect operator (**>**) to redirect output to a text file which can be used for more detailed analysis and reporting. Type the following command, which will take the output and redirect it to a text file that will be saved on the desktop.

```
netstat -a > c:\users\administrator\Desktop\netstat_scan.txt
```



8. Open the text file by double-clicking on the **netstat_scan** icon, and you will see the results in **Notepad**.

netstat_scan - Notepad

File Edit Format View Help

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:445	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:5985	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:47001	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49664	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49665	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49666	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49667	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49668	WIN-E3AIDIHECNG:0	LISTENING
TCP	0.0.0.0:49669	WIN-E3AIDIHECNG:0	LISTENING
TCP	192.168.0.50:139	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:135	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:445	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:5985	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:47001	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49664	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49665	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49666	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49667	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49668	WIN-E3AIDIHECNG:0	LISTENING
TCP	[::]:49669	WIN-E3AIDIHECNG:0	LISTENING
UDP	0.0.0.0:123	*:*	
UDP	0.0.0.0:5353	*:*	
UDP	0.0.0.0:5355	*:*	
UDP	127.0.0.1:53563	*:*	
UDP	192.168.0.50:137	*:*	
UDP	192.168.0.50:138	*:*	
UDP	[::]:123	*:*	
UDP	[::]:5353	*:*	

9. After looking at the report, you can close **Notepad**.

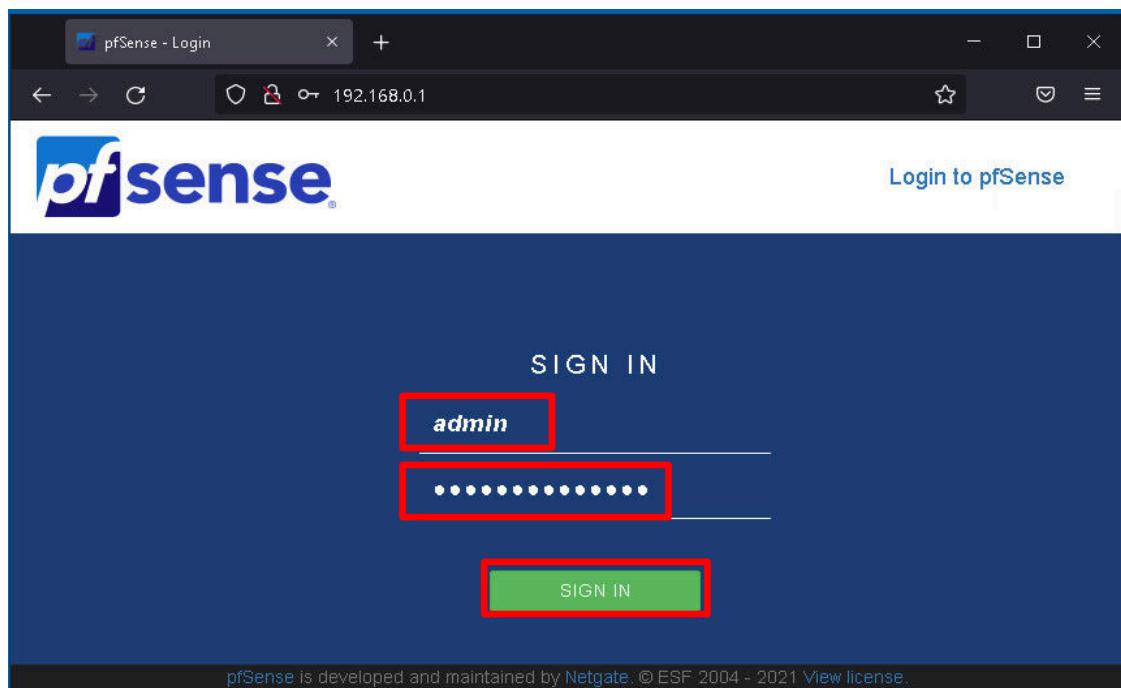
Netstat can also show connections that have been made to TCP/UDP ports on other devices.

10. Click on the **Firefox** browser icon in the taskbar to open a web browser.

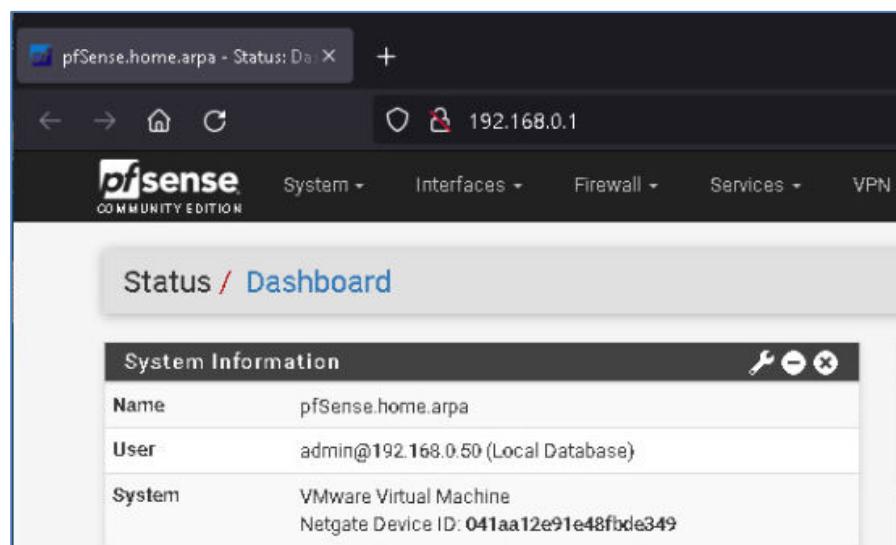


11. In the address bar of the browser, type 192.168.0.1, the IP address of the *pfSense* server.

12. Log in as **admin** using the password **NDGLabpass123!** and click the **SIGN IN** button.



13. You will be presented with the *Dashboard* for the *pfSense* firewall.



System Information	
Name	pfSense.home.arpa
User	admin@192.168.0.50 (Local Database)
System	VMware Virtual Machine Netgate Device ID: 041aa12e91e48fbde349

14. Set the focus to the command prompt window and run *netstat* again by typing:

```
netstat -a -p tcp
```

TCP	192.168.0.50:139	WIN-E3AIDIHECNG:0	LISTENING
TCP	192.168.0.50:52222	pfSense:http	ESTABLISHED
TCP	192.168.0.50:52253	pfSense:http	ESTABLISHED

Looking at the results, the *WinOS* computer has an *ESTABLISHED* connection with the pfSense web server on *http*: (Port 80) and *https*: (Port 443).



The **-p tcp** option is used to specify that you are only interested in *TCP* ports.

15. Type the following *netstat* command to examine which processes are utilizing certain ports. Note that the service name (*firefox.exe*) is listed on the left side for each port with an *ESTABLISHED* status.

```
netstat -b
```

TCP	192.168.0.50:52222	pfSense:http	ESTABLISHED
[firefox.exe]			
TCP	192.168.0.50:52265	pfSense:http	ESTABLISHED
[firefox.exe]			

16. Type the following *netstat* command to see the *TCP port* and the *Process ID* that was given to the *pfSense:http* web server listed in the previous scan. The *Process ID* is a unique number that is used to identify each process that is currently running in memory. Notice you are provided with the process ID for each *ESTABLISHED* port on the right side.

```
netstat -o
```

Active Connections					
Proto	Local Address	Foreign Address	State	PID	
TCP	127.0.0.1:52200	WIN-E3AIDIHECNG:52201	ESTABLISHED	388	
TCP	127.0.0.1:52201	WIN-E3AIDIHECNG:52200	ESTABLISHED	388	
TCP	127.0.0.1:52202	WIN-E3AIDIHECNG:52203	ESTABLISHED	5856	
TCP	127.0.0.1:52203	WIN-E3AIDIHECNG:52202	ESTABLISHED	5856	
TCP	127.0.0.1:52204	WIN-E3AIDIHECNG:52205	ESTABLISHED	5904	
TCP	127.0.0.1:52205	WIN-E3AIDIHECNG:52204	ESTABLISHED	5904	
TCP	127.0.0.1:52208	WIN-E3AIDIHECNG:52209	ESTABLISHED	5048	
TCP	127.0.0.1:52209	WIN-E3AIDIHECNG:52208	ESTABLISHED	5048	
TCP	127.0.0.1:52212	WIN-E3AIDIHECNG:52213	ESTABLISHED	6140	
TCP	127.0.0.1:52213	WIN-E3AIDIHECNG:52212	ESTABLISHED	6140	
TCP	127.0.0.1:52219	WIN-E3AIDIHECNG:52220	ESTABLISHED	4728	
TCP	127.0.0.1:52220	WIN-E3AIDIHECNG:52219	ESTABLISHED	4728	
TCP	127.0.0.1:52232	WIN-E3AIDIHECNG:52233	ESTABLISHED	2104	
TCP	127.0.0.1:52233	WIN-E3AIDIHECNG:52232	ESTABLISHED	2104	
TCP	127.0.0.1:52236	WIN-E3AIDIHECNG:52237	ESTABLISHED	5936	
TCP	127.0.0.1:52237	WIN-E3AIDIHECNG:52236	ESTABLISHED	5936	
TCP	192.168.0.50:52222	pfSense:http	ESTABLISHED	388	

17. The following *netstat* command is used to check the routing table. The routing table will tell you which destination traffic is being directed through which interface, along with the metric for that destination. You will also see that the default route for this device is listed as *192.168.0.1*, which is configured as the default gateway connecting to the *pfSense* router.

```
netstat -r
```

```
C:\Users\Administrator>netstat -r
=====
Interface List
12...00 50 56 99 56 8c .....vmxnet3 Ethernet Adapter
 1.....Software Loopback Interface 1
=====

IPv4 Route Table
=====
Active Routes:
Network Destination      Netmask     Gateway       Interface Metric
          0.0.0.0        0.0.0.0   192.168.0.1  192.168.0.50    271
         127.0.0.0      255.0.0.0   On-link        127.0.0.1    331
         127.0.0.1      255.255.255.255  On-link        127.0.0.1    331
 127.255.255.255  255.255.255.255  On-link        127.0.0.1    331
         192.168.0.0      255.255.255.0  On-link        192.168.0.50    271
         192.168.0.50      255.255.255.255  On-link        192.168.0.50    271
         192.168.0.255     255.255.255.255  On-link        192.168.0.50    271
         224.0.0.0        240.0.0.0   On-link        127.0.0.1    331
         224.0.0.0        240.0.0.0   On-link        192.168.0.50    271
 255.255.255.255  255.255.255.255  On-link        127.0.0.1    331
 255.255.255.255  255.255.255.255  On-link        192.168.0.50    271
=====
Persistent Routes:
Network Address      Netmask     Gateway Address Metric
          0.0.0.0        0.0.0.0   192.168.0.1  Default
```



You can also use the command:
`route print`
to display the routing table.

18. Close the command prompt window and minimize the web browser.

2 Performing Network Scans with nmap

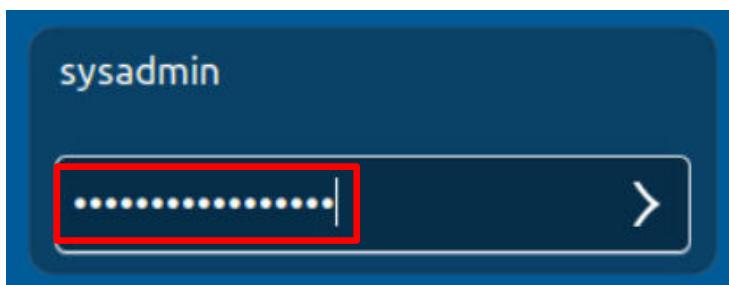
2.1 Basic nmap Features/Functions and Scanning from Internal Hosts

The *nmap* tool is often used in security analysis for network discovery and security auditing. *nmap* uses raw IP packets to determine hosts that are available on the network, services that those hosts are offering, operating systems and versions, packet filters and firewalls that are being used, and many other useful bits of information.

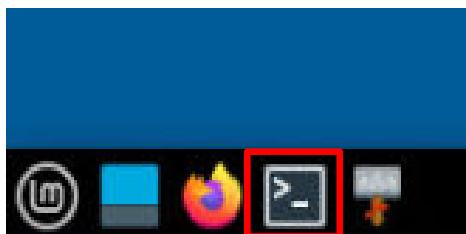
nmap is so powerful and ubiquitous in the security and forensics fields (not to mention the hacker community) that it was named the “Security Product of the Year” by various journals. The tool is so commonly seen that it has been used in several movies such as “The Matrix”, “Die Hard 4”, “Girl with the Dragon Tattoo” and “The Bourne Ultimatum”.

In this task, you will use *nmap* to scan hosts that are on the LAN and DMZ networks and discover which hosts are available, what ports they have open and listening, what OS they are running, and other details that an intruder may be able to detect with a rogue device.

1. Change focus to the **MintOS** computer.
2. Log in as the **sysadmin** using the password: NDGLabpass123!



3. Open a terminal session by clicking on the **Terminal** icon located at the bottom of the page.



4. Open and review *nmap*'s manual by typing the command below.

```
man nmap
```

```
Terminal - sysadmin@mintos: ~
File Edit View Terminal Tabs Help
NMAP(1) Nmap Reference Guide NMAP(1)

NAME
nmap - Network exploration tool and security / port scanner

SYNOPSIS
nmap [Scan Type...] [options] {target specification}

DESCRIPTION
Nmap ("Network Mapper") is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that
Manual page nmap(1) line 1 (press h for help or q to quit)
```



nmap has many options, including its own scripting engine. Review the man pages to get familiar with the switches and options. Press the Spacebar to go to the next page or press Enter to go to the next line.

5. Once finished reviewing the man page, press **Q** to quit and bring the shell prompt back.
6. Let's scan the *WinOS* host using an *nmap* scan with no options by entering the command:

```
nmap 192.168.0.50
```

```
sysadmin@mintos:~$ nmap 192.168.0.50
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-26 18:53 EDT
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.04 seconds
```

7. The host seems to be down, but it might actually be up and blocking the host discovery packets. Perform the following *nmap* scan, but this time with the **-Pn** option, which will assume the *WinOS* computer is up and will scan the first 1000 TCP/UDP ports:

```
nmap 192.168.0.50 -Pn
```

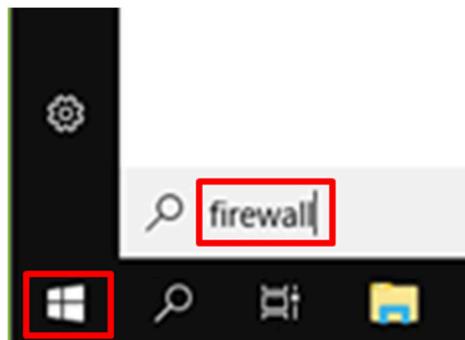
```
sysadmin@mintos:~$ nmap 192.168.0.50 -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2021-10-15 12:39 EDT
Nmap scan report for 192.168.0.50
Host is up.
All 1000 scanned ports on 192.168.0.50 are filtered
Nmap done: 1 IP address (1 host up) scanned in 201.34 seconds
```



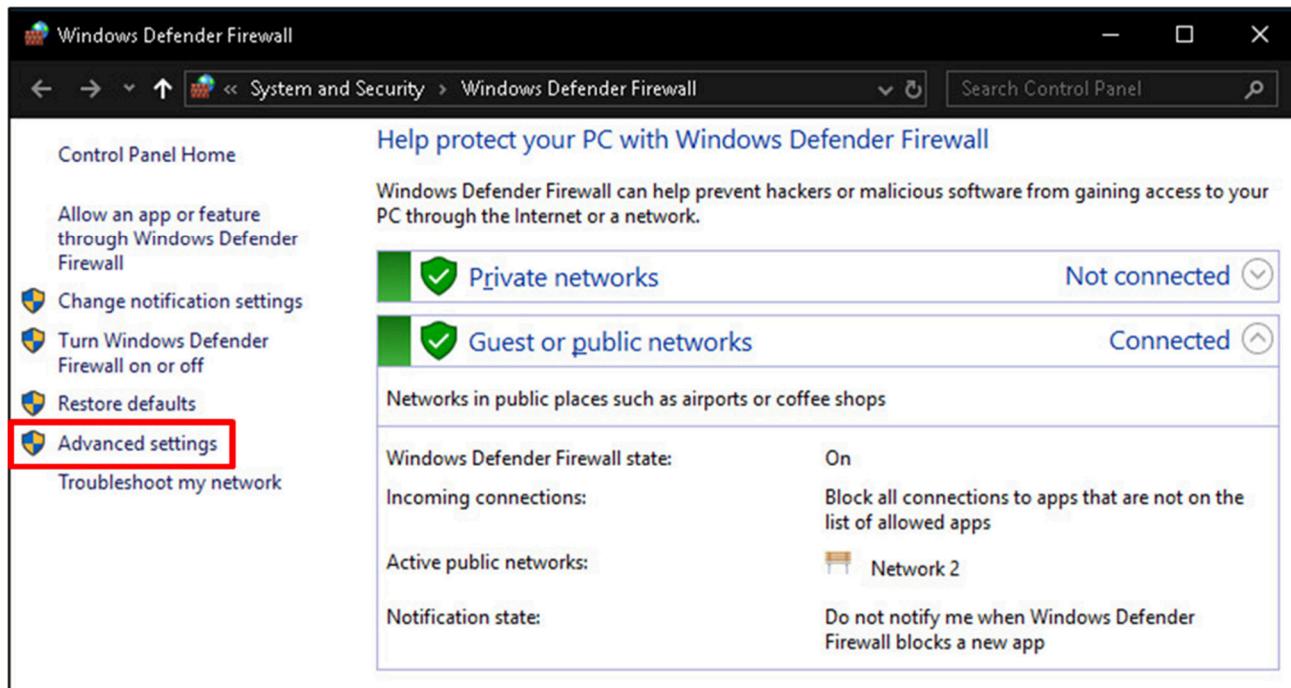
The *nmap* scan might take as long as 4 minutes to complete.

This time, *nmap* confirms that the host is up, but the Windows Defender Firewall is blocking all inbound traffic. This is the purpose of a firewall, namely to block incoming and outgoing traffic based on IP addresses, TCP/UDP ports, and/or applications.

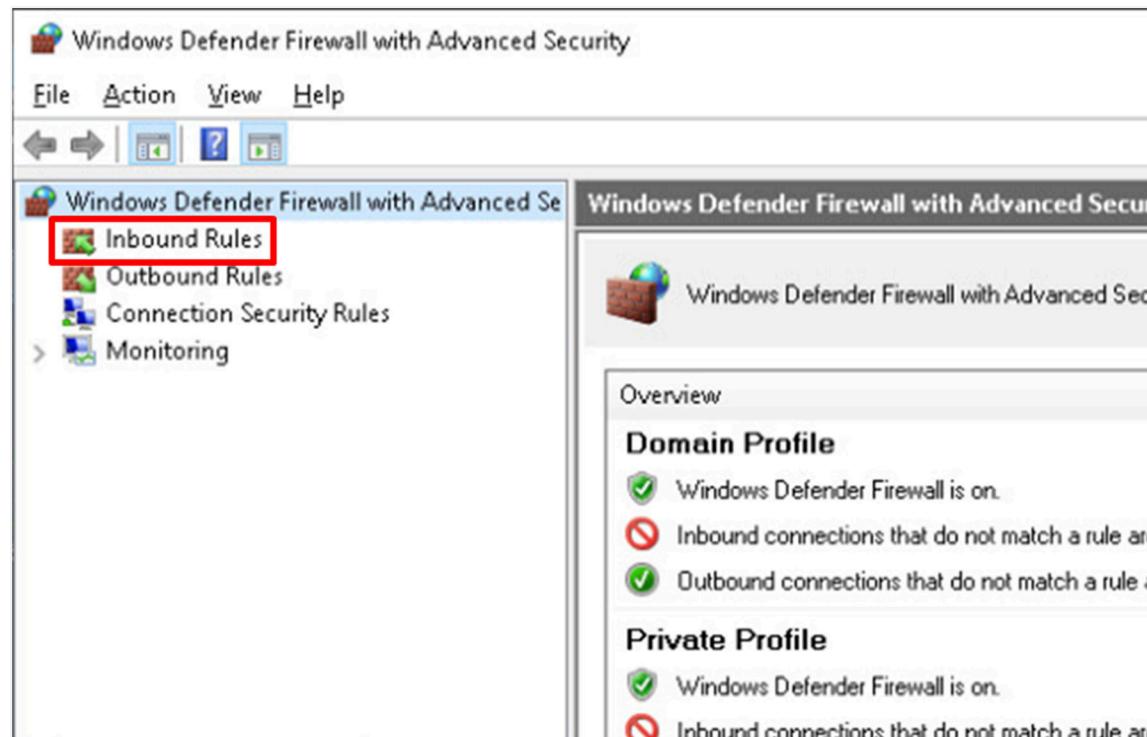
8. The firewall uses a set of rules, to control the traffic. In the next step, we will add a rule to the Firewall Rule Table that will allow traffic from the IP address of the *MintOS* computer only. Change focus to the *WinOS* computer.
9. Click on the **Windows Start** button in the bottom-left corner, type **firewall** and press **Enter**.



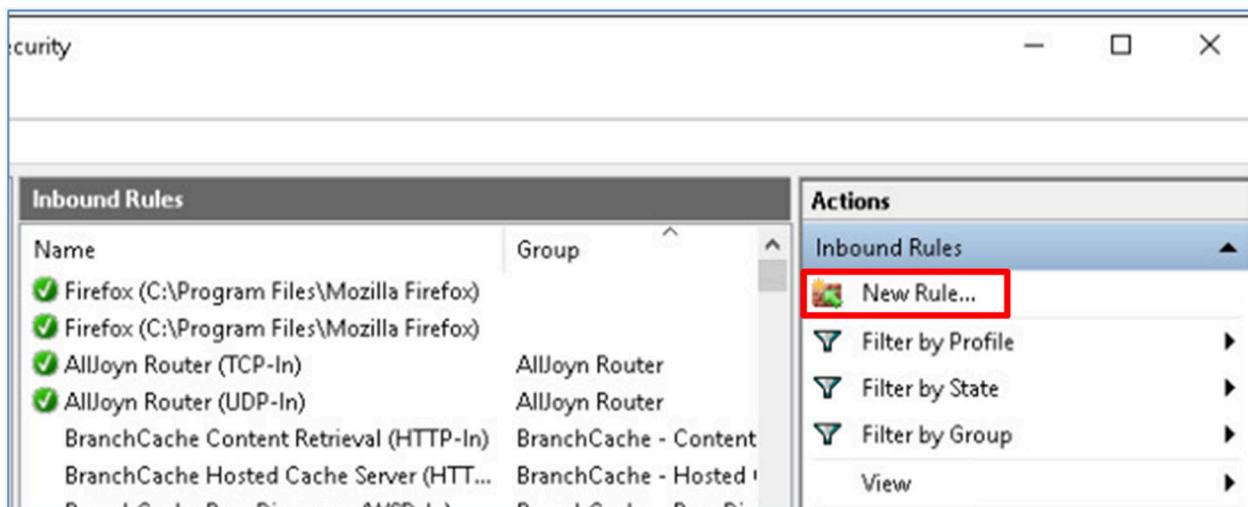
10. On the left side of the window, click on **Advanced Settings**.



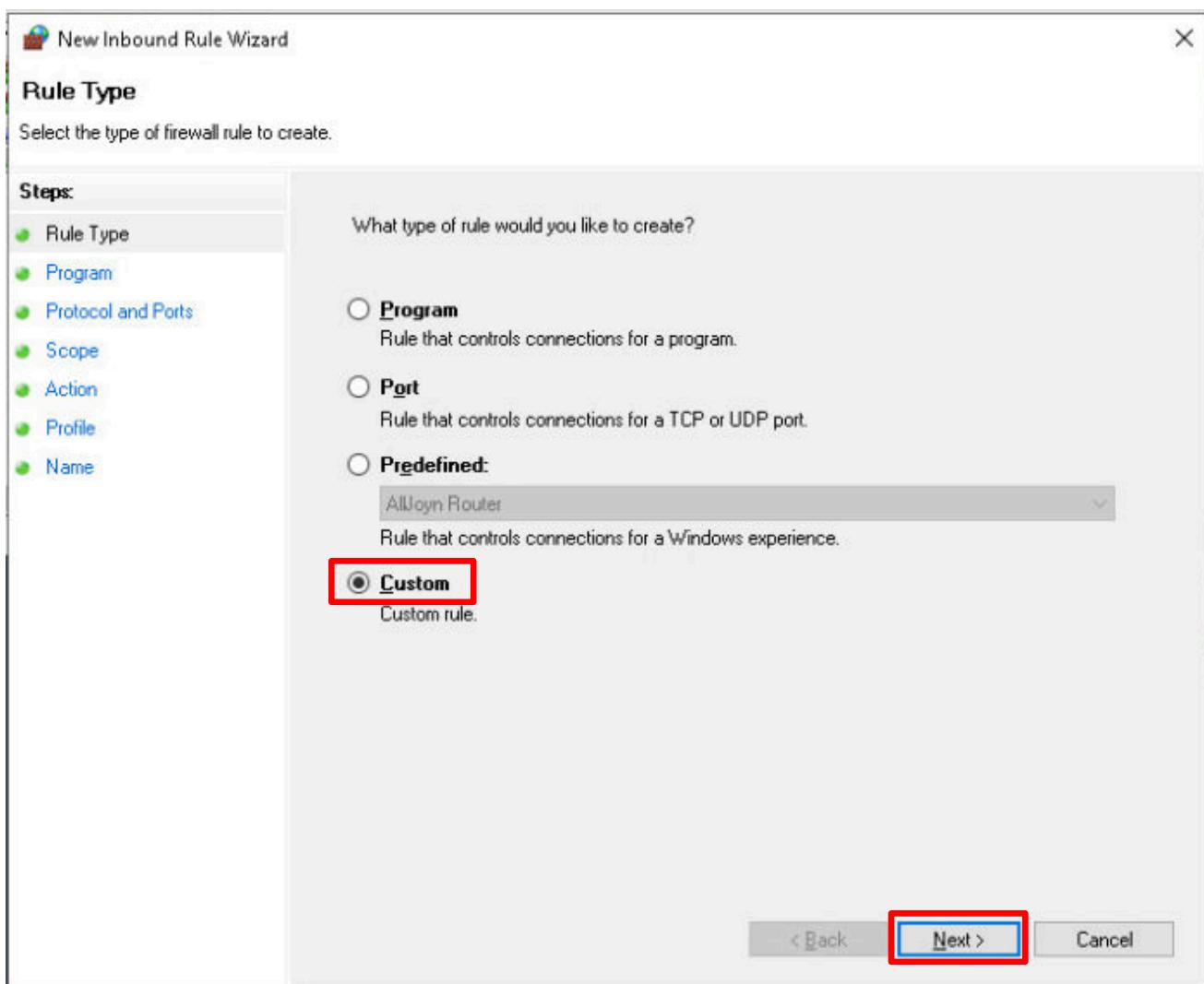
11. On the left panel, click on **Inbound Rules**.



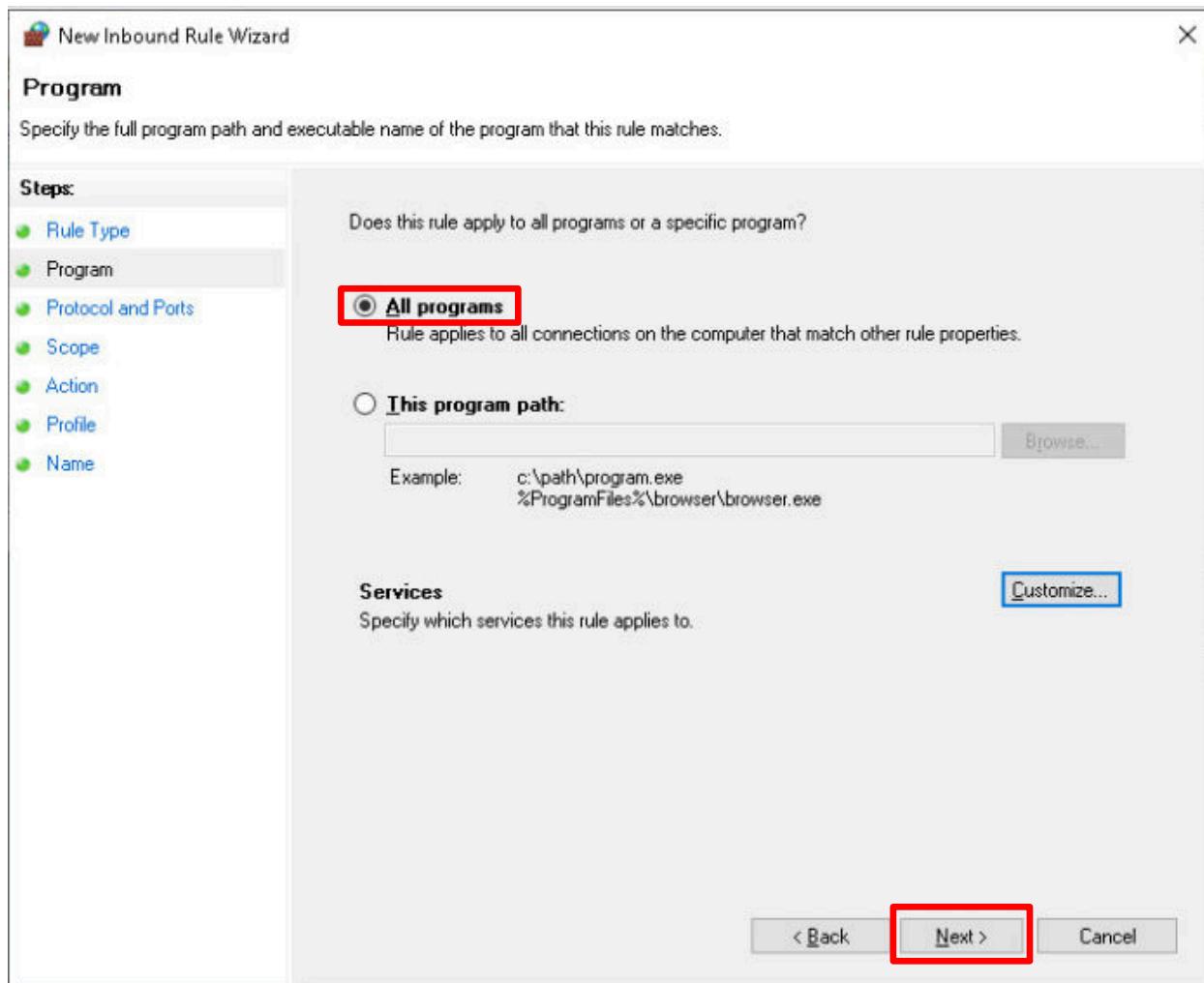
12. On the right panel, click on **New Rule...**.



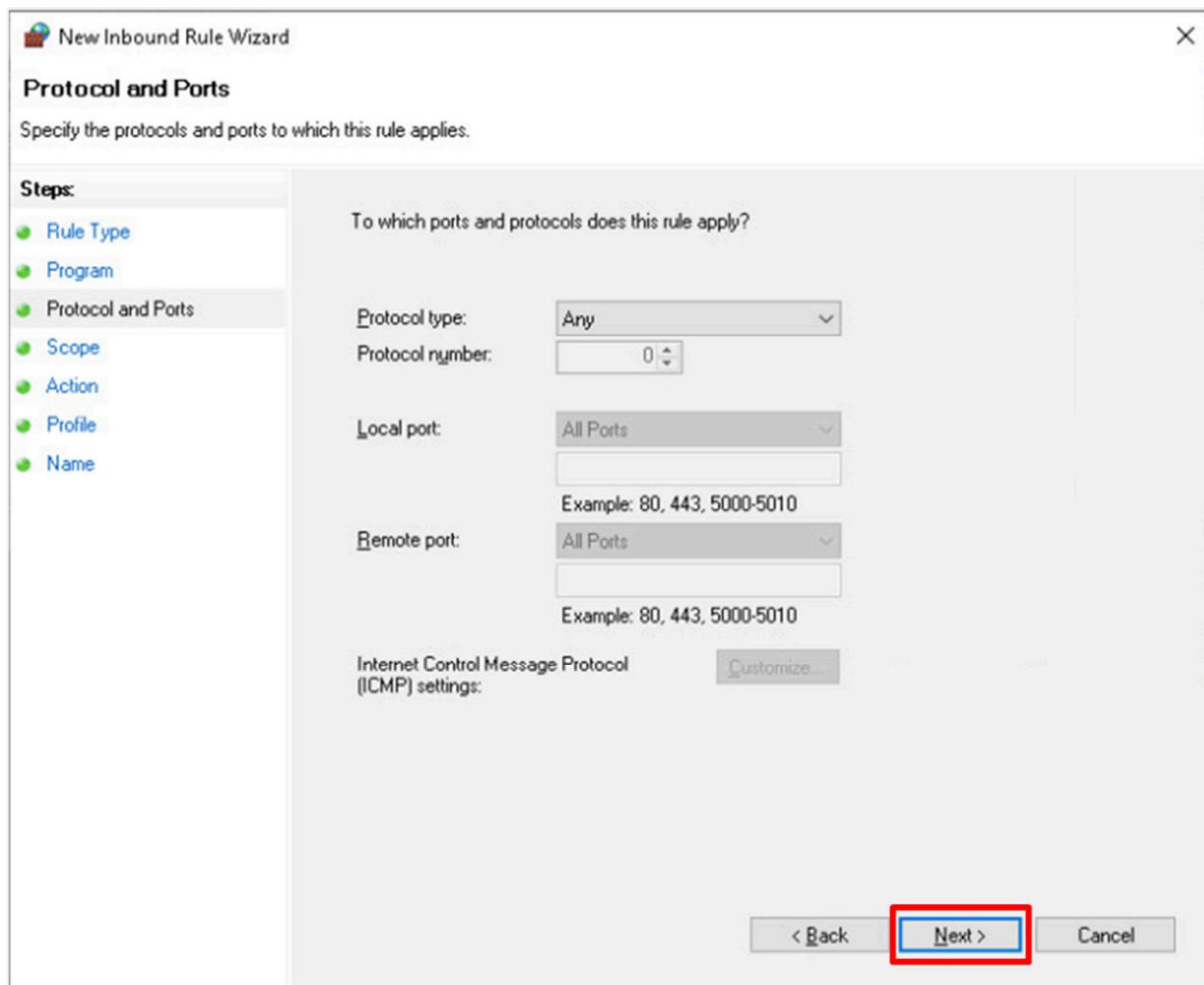
13. In the *New Inbound Rule Wizard*, in the *Rule Type* step, click on the **Custom** radio button and then click the **Next >** button.



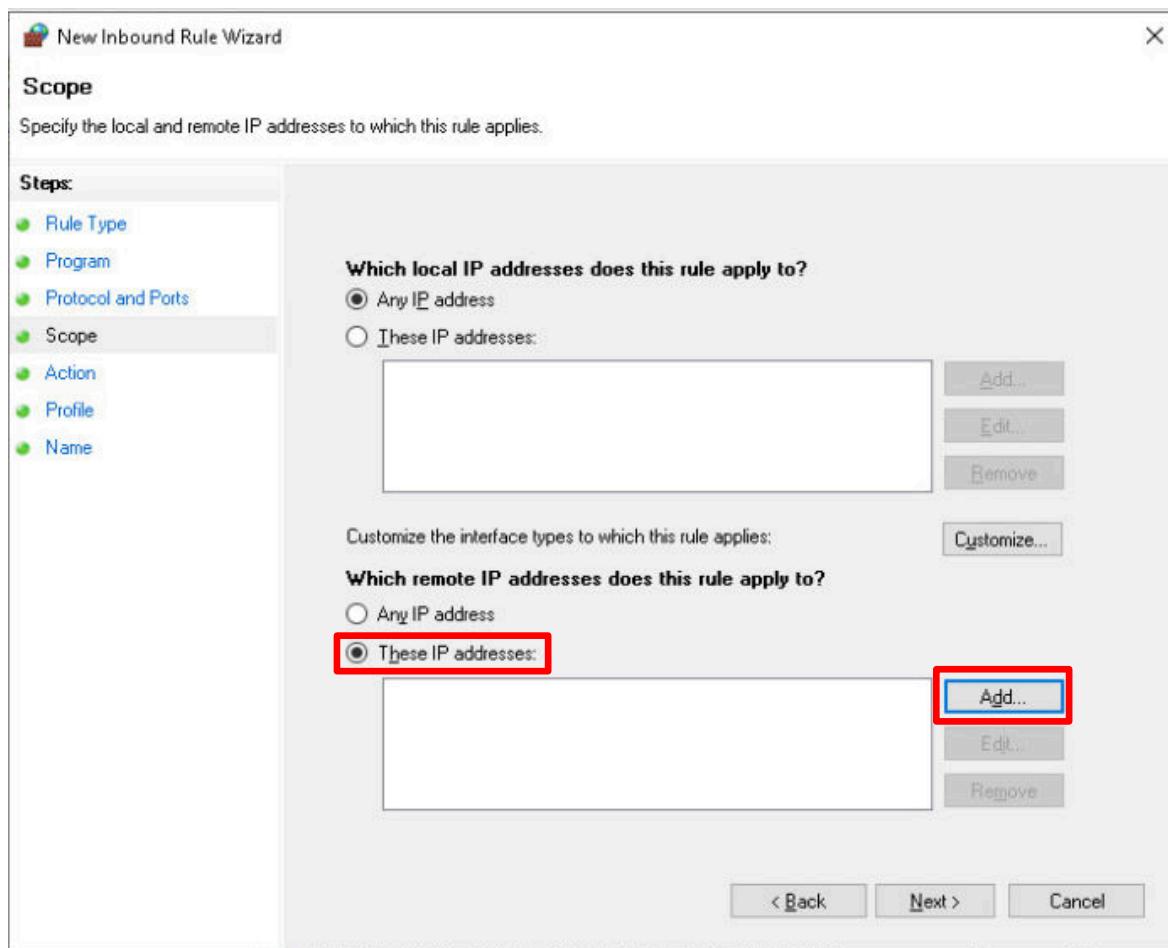
14. In the *Program* step, if it is not already selected, click on the **All Programs** radio button and then click the **Next** button.



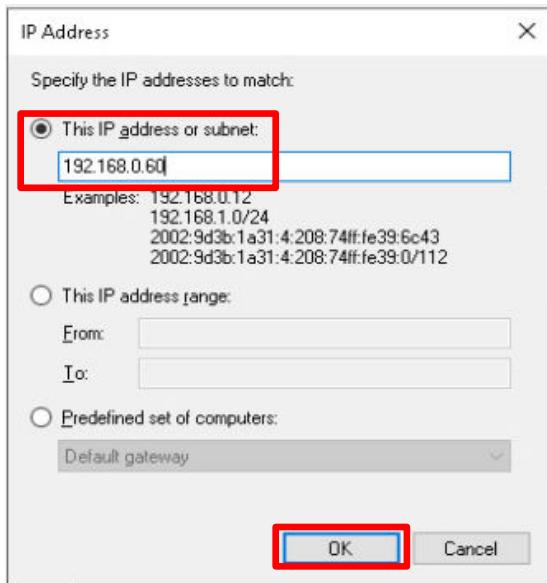
15. In the *Protocol and Ports* step, leave all the entries at their default values and click the **Next** button.



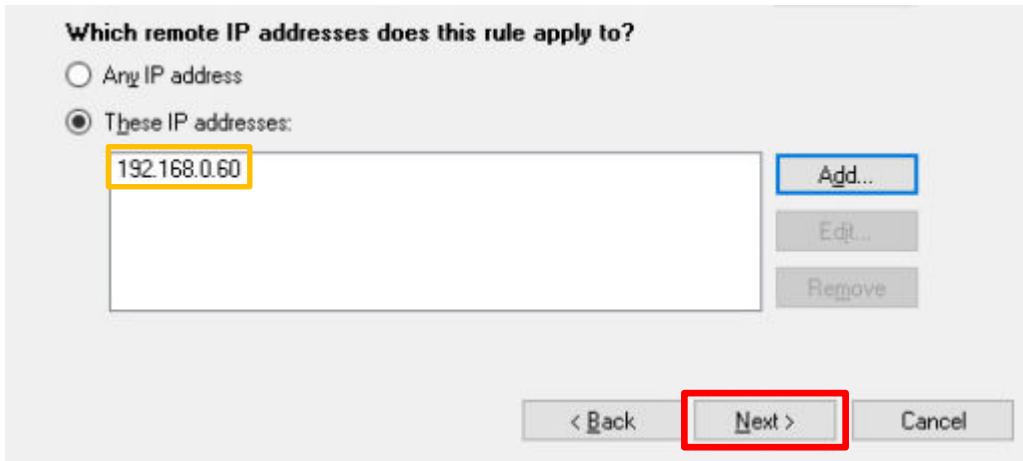
16. In the *Scope* step, in the lower-right pane under the *Which remote IP addresses does this rule apply to?* section, click the **These IP addresses** radio button and then click the **Add** button:



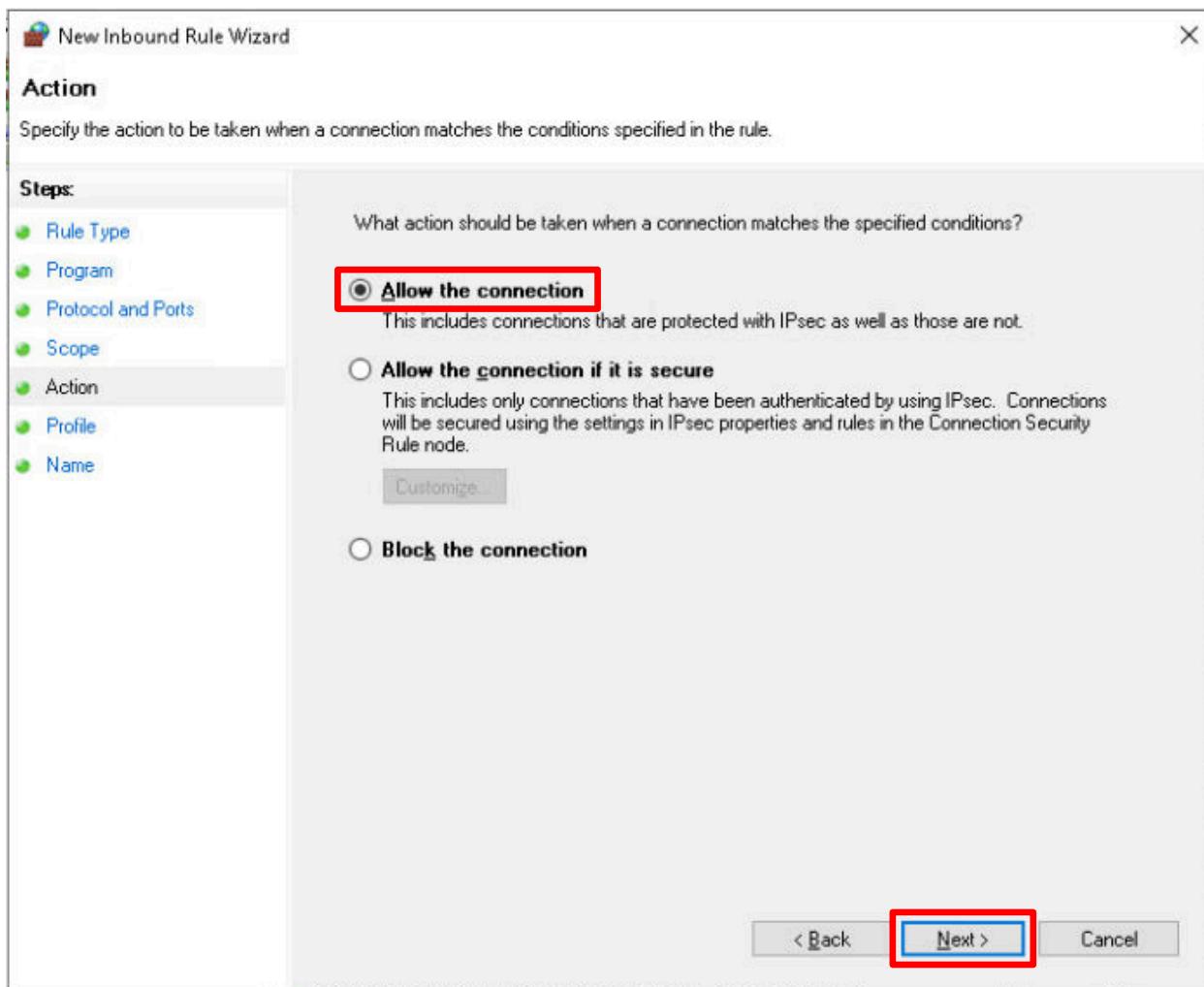
17. In the *IP Address* popup box, confirm the **This IP address or subnet** radio button is selected and enter 192.168.0.60, which is the IP address of the *MintOS* computer, and click **OK**.



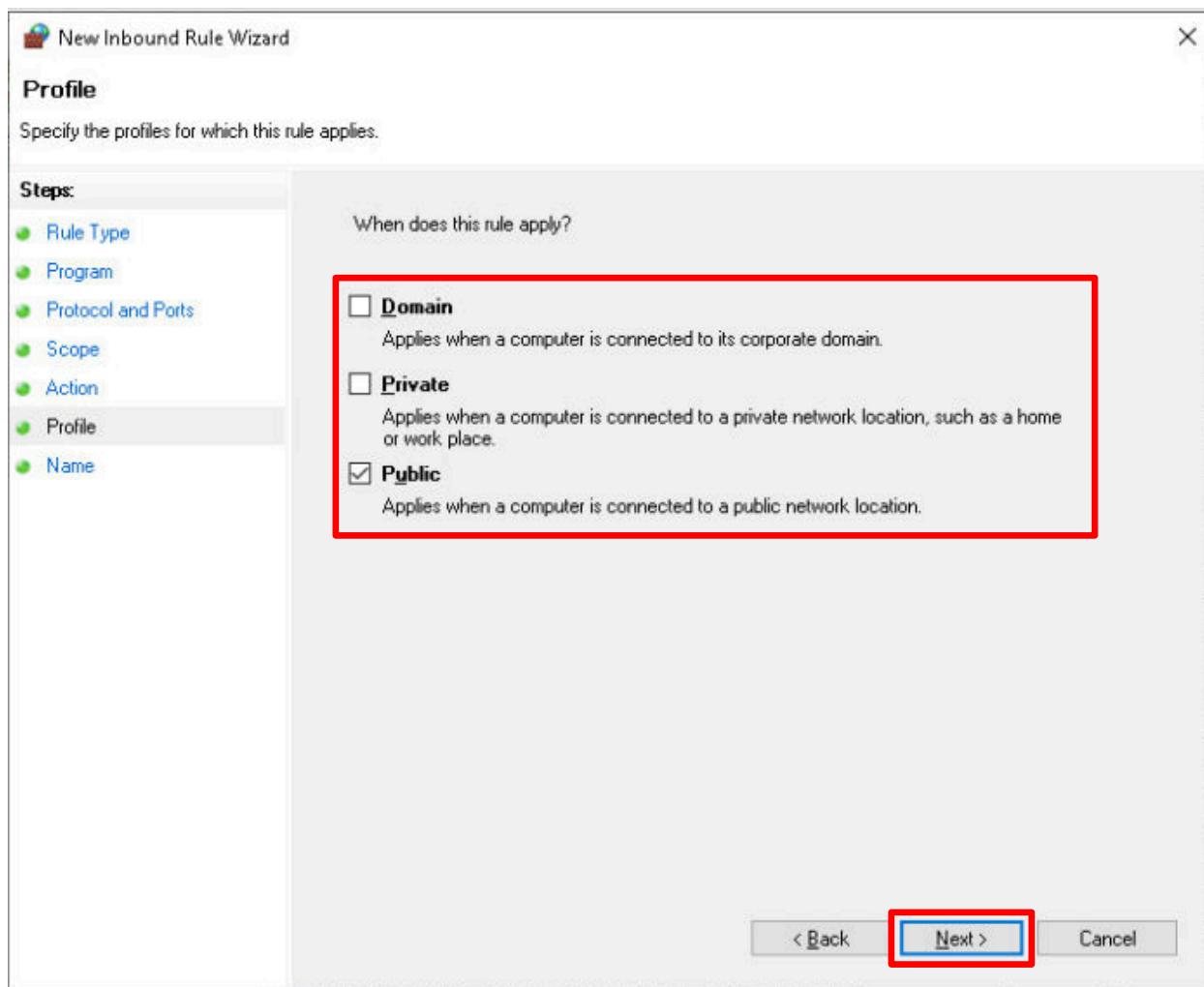
18. When you return to the *Scope* window, confirm that the **MintOS** IP address is in the box and click the **Next** button.



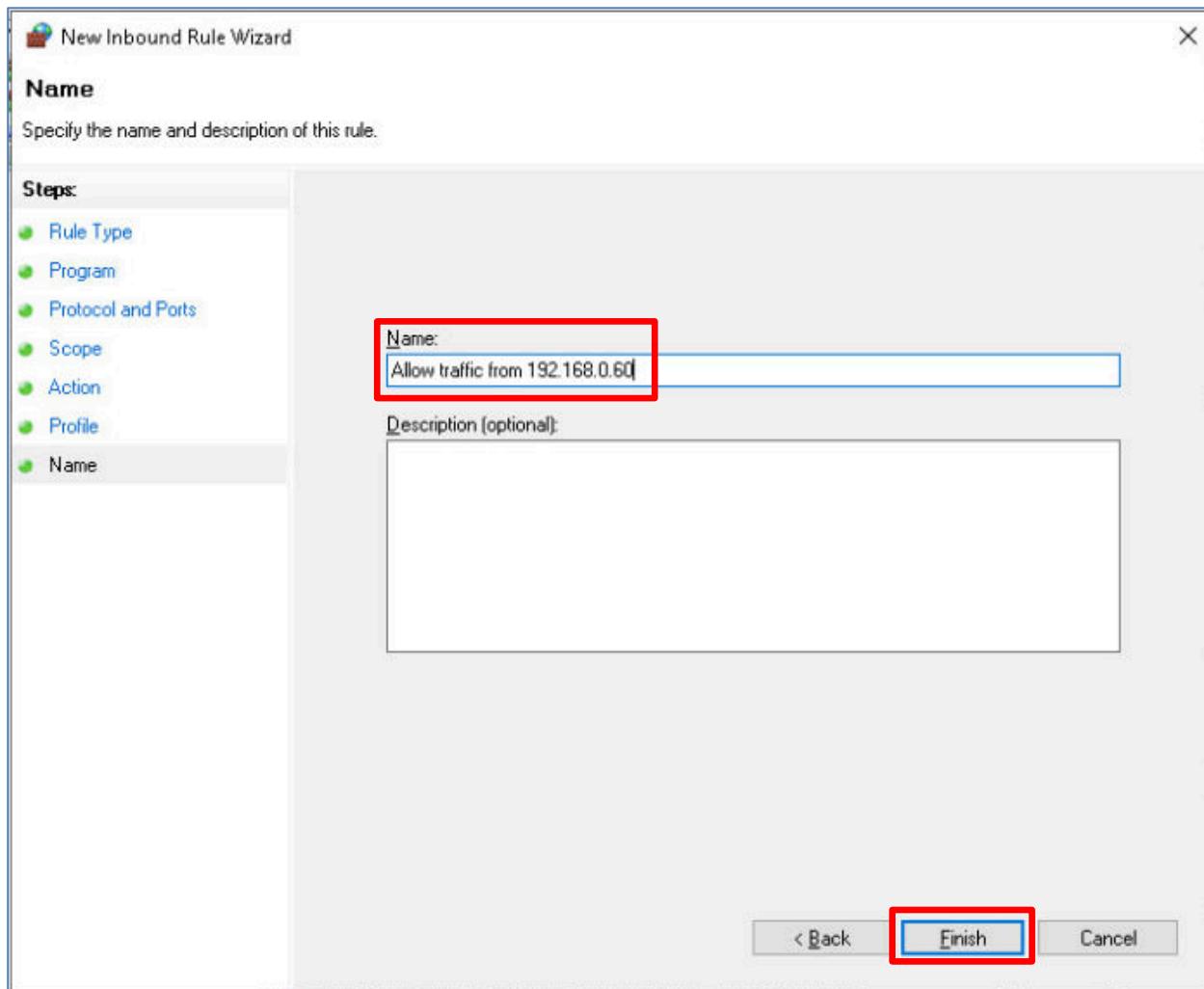
19. On the **Action** step, confirm the **Allow the connection** radio button is selected and click the **Next** button.



20. In the **Profile** step, uncheck the **Domain** and **Private** checkboxes (leaving the **Public** checkbox selected) and click **Next**.



21. In the *Name* step, enter Allow traffic from 192.168.0.60 and click **Finish**.



22. Close the **Windows Defender Firewall with Advanced Security** window, and then close the **Windows Defender Firewall** window.

23. Return to the **MintOS** computer.

24. Now execute the *nmap* command with the **-Pn** option again targeting the **WinOS** host.

```
nmap 192.168.0.50 -Pn
```

```
sysadmin@mintos:~$ nmap 192.168.0.50 -Pn
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-31 14:52 EDT
Nmap scan report for 192.168.0.50
Host is up (0.00064s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
```

This time, *nmap* will identify open ports on the *WinOS* host since the firewall table has been set to allow connections from the *MintOS* computer. Notice there are very few ports that are open on the *WinOS* host. You want as few ports open on a host as possible. Every open port presents an opportunity for hackers to break into your system. It's like leaving the front door wide open.

25. Let's try running *nmap* against **UbuntuSRV** in the DMZ. Type the following command:

```
nmap 172.16.1.10
```

```
sysadmin@mintos:~$ nmap 172.16.1.10
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-02 18:57 EDT
Nmap scan report for 172.16.1.10
Host is up (0.00017s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Once again, notice that there are only three ports, **TCP Port 22**, which is used for **SSH Remote Access**, **TCP Port 80**, which is **HTTP**; and **TCP Port 443**, which is **HTTPS**:



The *UbuntuSRV* shows that **Port 80** and **Port 443** is open since the Apache web server is running.

26. Remain on the *MintOS* computer and continue to the next step.

2.2 Perform OS Fingerprinting from nmap

Remote operating system detection is a common way for a potential hacker to discover what types of hosts are running on the network. One method is called **OS Fingerprinting**, which collects detailed information from the TCP/IP stack of the hosts. *nmap* gathers fingerprint data by simply adding the **-O** option to the scan. You can also view the processes from the scan with the **-v** (verbose mode) option.

1. Enter the following command to run an *nmap* fingerprint against the **Kali** computer. You must use **sudo** because this *nmap* option requires root privileges. When asked for the **[sudo] password**, enter: NDGLabpass123!

```
sudo nmap -O -v 203.0.113.2/32
```

```
sysadmin@mintos:~$ sudo nmap -O -v 203.0.113.2
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-02 17:09 EDT
Initiating Ping Scan at 17:09
Scanning 203.0.113.2 [4 ports]
Completed Ping Scan at 17:09, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:09
Completed Parallel DNS resolution of 1 host. at 17:09, 0.00s elapsed
Initiating SYN Stealth Scan at 17:09
Scanning 203.0.113.2 [1000 ports]
Completed SYN Stealth Scan at 17:09, 0.05s elapsed (1000 total ports)
Initiating OS detection (try #1) against 203.0.113.2
Nmap scan report for 203.0.113.2
Host is up (0.00029s latency).
All 1000 scanned ports on 203.0.113.2 are closed
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.18
OS details: Linux 2.6.18, Linux 2.6.30
Network Distance: 2 hops

Read data files from: /usr/bin/../share/nmap
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.10 seconds
Raw packets sent: 1016 (45.346KB) | Rcvd: 1005 (40.762KB)
```

Notice a *Linux* operating system was detected.

2. Type the following command to scan and fingerprint the *WinOS* host (you must use **sudo** because this *nmap* option requires root privileges).

```
sudo nmap -O -v 192.168.0.50/32
```

Notice that the *nmap* fingerprint does not get an OS match.

```
sysadmin@gmintos:~$ sudo nmap -O -v 192.168.0.50/32
Starting Nmap 7.80 ( https://nmap.org ) at 2022-08-01 00:43 EDT
Initiating ARP Ping Scan at 00:43
Scanning 192.168.0.50 [1 port]
Completed ARP Ping Scan at 00:43, 0.04s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:43
Completed Parallel DNS resolution of 1 host. at 00:43, 0.00s elapsed
Initiating SYN Stealth Scan at 00:43
Scanning 192.168.0.50 [1000 ports]
Discovered open port 445/tcp on 192.168.0.50
Discovered open port 139/tcp on 192.168.0.50
Discovered open port 135/tcp on 192.168.0.50
Completed SYN Stealth Scan at 00:43, 4.45s elapsed (1000 total ports)
Initiating OS detection (try #1) against 192.168.0.50
Retrying OS detection (try #2) against 192.168.0.50
Nmap scan report for 192.168.0.50
Host is up (0.00041s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
MAC Address: 00:50:56:99:56:8C (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=257 (Good luck!)
IP ID Sequence Generation: Incremental
```

3. This time, scan and fingerprint the *UbuntuSRV* computer.

```
sudo nmap -O 172.16.1.10
```

```
sysadmin@mintos:~$ sudo nmap -O 172.16.1.10
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-02 19:35 EDT
Nmap scan report for 172.16.1.10
Host is up (0.00046s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ )
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=5/2%OT=22%CT=1%CU=36282%PV=Y%DS=2%DC=I%G=Y%TM=62706AC3
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=F5%GCD=1%ISR=104%TI=Z%II=I%TS=A)OPS(01=M5B
OS:4ST11Nw7%02=M5B4ST11Nw7%03=M5B4NNT11Nw7%04=M5B4ST11Nw7%05=M5B4ST11Nw7%06
OS:=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF
OS:=Y%T=40%W=FAF0%O=M5B4NNSNw7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=AS%RD=0%
OS:Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6
OS:(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RU
OS:D=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.44 seconds
```

This time, *nmap* generated a special ASCII-encoded version of the operating system.



The fingerprint contains relevant information that *nmap* was able to glean from the host. It is formatted in such a way that it can be submitted to nmap.org where it can be used to add the OS information to the *nmap* database. With some simple editing, the fingerprint can be “cleaned-up” and easier to decipher.

4. Type the following *nmap* command with the **-A** option to show additional information about the *WinOS* computer:

```
sudo nmap -A 192.168.0.50
```

```
sysadmin@mintos:~$ sudo nmap -A 192.168.0.50
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-02 19:16 EDT
Nmap scan report for 192.168.0.50
Host is up (0.00028s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE          VERSION
135/tcp    open  msrpc            Microsoft Windows RPC
139/tcp    open  netbios-ssn       Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds?
902/tcp    open  ssl/vmware-auth  VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
912/tcp    open  vmware-auth     VMware Authentication Daemon 1.0 (Uses VNC, SOAP)
MAC Address: 00:50:56:99:56:8C (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: WIN-E3AIDIHECNG, NetBIOS user: <unknown>, NetBIOS MAC: 00:50:56:99:56:8c (VMware)
| smb2-security-mode:
|   2.02:
|_  Message signing enabled but not required
| smb2-time:
|   date: 2022-05-02T23:16:50
|_ start_date: N/A

TRACEROUTE
HOP RTT      ADDRESS
1  0.28 ms  192.168.0.50

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 114.95 seconds
```

In the example, you will see that *nmap* has identified the *Windows* operating system (using the Service Info), the *NetBIOS* name of the computer, the *MAC Address* and the *NIC*'s manufacturer, and even the version of SMB that is running on the *WinOS* computer. That's quite a bit of information that can be gleaned.

5. Now, let's use *nmap* to try to identify versions of software running on the ports on the *UbuntuSRV*. Type the following command:

```
sudo nmap -A 172.16.1.10
```

Once again, you will see that *nmap* has identified the operating system as Ubuntu Linux and the version details on the service ports that were discovered.

```
sysadmin@mintos:~$ sudo nmap -A 172.16.1.10
[sudo] password for sysadmin:
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-02 20:02 EDT
Nmap scan report for 172.16.1.10
Host is up (0.00052s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  ssl/http   Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
| ssl-cert: Subject: commonName=172.16.1.10/organizationName=NDG/stateOrProvinceName=California/countryName=US
| Not valid before: 2021-09-14T05:44:10
| Not valid after:  2022-09-14T05:44:10
|_tls-alpn:
|_ http/1.1
443/tcp   open  ssl/http   Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
| ssl-cert: Subject: commonName=172.16.1.10/organizationName=NDG/stateOrProvinceName=California/countryName=US
| Not valid before: 2021-09-14T05:44:10
| Not valid after:  2022-09-14T05:44:10
|_tls-alpn:
|_ http/1.1
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ )
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=5/2%OT=22%CT=1%CU=30112%PV=Y%DS=2%DC=T%G=Y%TM=62707129
OS:%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10F%TI=Z%II=I%TS=A)OPS(O1=M5
OS:B4ST11NW7%02=M5B4ST11NW7%03=M5B4NNT11NW7%04=M5B4ST11NW7%05=M5B4ST11NW7%0
OS:6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%D
OS:F=Y%T=40%W=FAF0%O=M5B4NSMW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+F=AS%RD=0
OS:%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=%RD=0%Q=)T
OS:6(R=N)T7(R=N)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%R
OS:UD=G)IE(R=Y%DFI=N%T=40%CD=S)

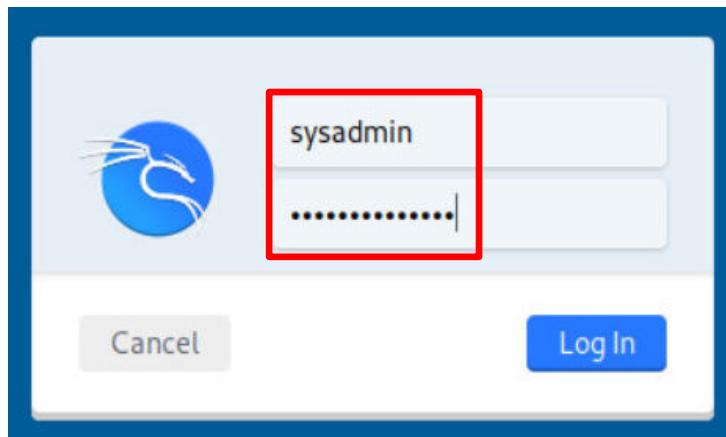
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 8888/tcp)
HOP RTT      ADDRESS
1  0.24 ms  pfSense.home.arpa (192.168.0.1)
2  0.65 ms  172.16.1.10
```

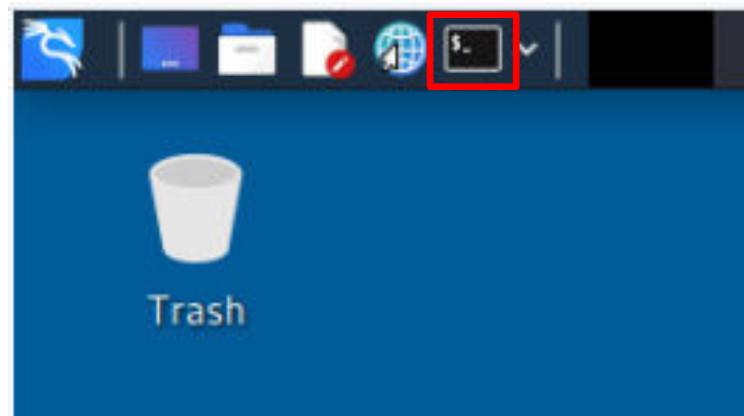
2.3 Using nmap to Scan Hosts from an External Network

Hackers will most likely perform reconnaissance from outside an organization's LAN and/or DMZ. This section will use a *Kali Linux* attack computer located on the exterior network to probe the hosts on the LAN and DMZ networks.

1. Change the focus to the **Kali** computer.
2. Log in as **sysadmin** using the password: **NDGLabpass123!**



3. Click on the **Terminal** icon in the taskbar at the top of the screen.



4. Let's scan the **192.168.0.0/24** network using *nmap*.

```
nmap 192.168.0.0/24
```

```
(sysadmin㉿kali)-[~]
$ nmap 192.168.0.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-26 20:56 EDT
Nmap done: 256 IP addresses (0 hosts up) scanned in 104.30 seconds
```

There will be no response from the **192.168.0.0/24** network. This is because there is a firewall between the External network and the LAN and DMZ. This firewall will block all traffic using RFC1918 which stops all traffic from public IP addresses from being routed to private IP addresses.



RFC1918 was originally intended to allow for conservation of IPv4 address space by allowing organizations to have networks of non-internet routable addresses and when combined with the NAT/PAT protocol could allow inside, private address hosts to route packets to the public internet and receive directed replies.

It also provided a natural firewall that could be used to block all unsolicited traffic from the public network into the private network.

This is an excellent first wall of defense.

- Let's try the scan on the 172.16.1.0/28 network by entering the command:

```
nmap 172.16.1.0/28
```

```
(sysadmin㉿kali)-[~]
$ nmap 172.16.1.0/28
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-24 13:34 EDT
Nmap scan report for 172.16.1.10
Host is up (0.00052s latency).
Not shown: 996 filtered ports
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    open     http
443/tcp   open     https
15000/tcp closed   hydap

Nmap done: 16 IP addresses (1 host up) scanned in 18.99 seconds
```

The scan will show that there are open ports on the 172.16.1.10 host (*UbuntuSRV*). Even though this is a private network, the firewall has been configured to forward packets on ports 22, 80, 443, and 15000. The *UbuntuSRV* computer is hosting a Web Server on a DMZ network and will need ports 22 (SSH), 80 (HTTP), and 443 (HTTPS) open for access from the internet. Port 15000 has been forwarded by the firewall to accommodate a lab that will exploit an exposed and unprotected port.

For this next part of the task, we will need to allow traffic from the public network into the 192.168.0.0/24 private network.

6. Change focus to the **WinOS** computer.
7. Restore the **Firefox** web browser.



You should **NEVER** consider doing this on a production firewall as it will expose your entire private network to all kinds of threats.

You should see the **Dashboard** for the *pfSense* firewall.

The screenshot shows the pfSense Status / Dashboard interface. At the top, there's a header bar with the pfSense logo, navigation links (System, Interfaces, Firewall, Services, VPN), and a status message "Status: Da". Below the header is a search bar with the IP address "192.168.0.1". The main content area has a title "Status / Dashboard". Underneath is a "System Information" table:

System Information	
Name	pfSense.home.arpa
User	admin@192.168.0.50 (Local Database)
System	VMware Virtual Machine Netgate Device ID: 041aa12e91e48fbde349
BIOS	Vendor: Phoenix Technologies LTD Version: 6.00 Release Date: Mon Sep 21 2015
Version	2.5.2-RELEASE (amd64) built on Fri Jul 02 15:33:00 EDT 2021 FreeBSD 12.2-STABLE
<p>The system is on the latest version. Version information updated at Fri Aug 27 1:25:19 UTC 2021 ↻</p>	
CPU Type	Intel(R) Xeon(R) D-2146NT CPU @ 2.30GHz AES-NI CPU Crypto: Yes (inactive) QAT Crypto: No
Hardware crypto	
Kernel PTI	Enabled

We need to add a rule to allow inbound and outbound traffic to be passed to and from the External network to the LAN and DMZ.

8. Click on the **Firewall** menu item and then click on **Rules**.

The screenshot shows the pfSense interface. The top navigation bar has a red box around the 'Firewall' dropdown menu. A secondary dropdown menu is open under 'Firewall', showing options: Aliases, NAT, Rules (which is highlighted with a red box), Schedules, Traffic Shaper, and Virtual IPs.

9. Make sure the **WAN** item is selected and click the **Add to Top** button.

The screenshot shows the 'Firewall / Rules / WAN' page. The 'WAN' tab is selected, indicated by a red box. A message at the top says: 'The changes have been applied successfully. The firewall rules are now reloading in the background. Monitor the filter reload progress.' Below the message, there is a table titled 'Rules (Drag to Change Order)'. The table lists several firewall rules, including ones for port forwarding. At the bottom of the table, there is a row of buttons: 'Add' (highlighted with a red box), 'Delete', 'Save', and 'Separator'.

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	0/0 B	*	RFC 1918 networks	*	*	*	*	*		Block private networks	
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	172.16.1.10	15000	*	none			
<input type="checkbox"/>	✓ 0/348 B	IPv4 TCP	*	*	172.16.1.10	443 (HTTPS)	*	none			
<input type="checkbox"/>	✓ 0/2 KIB	IPv4 TCP	*	*	172.16.1.10	80 (HTTP)	*	none			
<input type="checkbox"/>	✓ 0/0 B	IPv4 TCP	*	*	172.16.1.10	22 (SSH)	*	none			



You will see in the list of firewall rules the four TCP ports that are being forwarded to the *UbuntuSRV* computer.

10. In the *Edit Firewall Rule* section, use the list arrow to change the *Protocol* to **Any**.

Edit Firewall Rule

Action	Pass
Choose what to do with packets that match the criteria specified below. Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.	
Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
Interface	WAN
Choose the interface from which packets must come to match this rule.	
Address	IPv4
Family	Select the Internet Protocol version this rule applies to.
Protocol	Any
Choose which IP protocol this rule should match.	

11. Scroll down to the lower half of the window.

12. In the *Extra Options* section, change the description to **Allow all traffic from the External network to the LAN**. Then, click the **Save** button at the bottom of the window.

Extra Options

Log	<input type="checkbox"/> Log packets that are handled by this rule Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
Description	Allow all traffic from the External network to the LAN
A description may be entered here for administrative reference. A maximum of 52 characters will be used in the ruleset and displayed in the firewall log.	
Advanced Options	Display Advanced
Save	

13. Note the new rule has been added to the list. Click the **Apply Changes** button.

States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
0/0 B	*	RFC 1918 networks	*	*	*	*	*	*	Block private networks	
0/0 B	IPv4*	*	*	*	*	*	none		Allow all traffic from the External network to the LAN	
0/300 B	IPv4 TCP	*	*	172.16.1.10	15000	*	none			
3/471 KIB	IPv4 TCP	*	*	172.16.1.10	443 (HTTPS)	*	none			
13/487 KIB	IPv4 TCP	*	*	172.16.1.10	80 (HTTP)	*	none			
0/43 KIB	IPv4 TCP	*	*	172.16.1.10	22 (SSH)	*	none			

14. Change focus back to the **Kali** computer.

15. Let's scan the whole **192.168.0.0/24** network again using *nmap* to see which ports are open on the hosts in the LAN.

```
nmap 192.168.0.0/24
```

```
(sysadmin㉿kali)-[~]
$ nmap 192.168.0.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-16 11:56 EDT
Nmap scan report for 192.168.0.1
Host is up (0.00057s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http

Nmap scan report for 192.168.0.60
Host is up (0.00023s latency).
All 1000 scanned ports on 192.168.0.60 are closed

Nmap done: 256 IP addresses (2 hosts up) scanned in 11.79 seconds
```



Notice that the *WinOS* computer, which is set to the IP address **192.168.0.50** is missing from the scan. That's because the *Windows Defender Firewall* is blocking traffic. In the previous step, we only allowed traffic from the *MintOS* computer at **192.168.0.60**.

16. Let's take a look at the DMZ. In the terminal window, enter the following command:

```
nmap 172.16.1.0/28
```

```
(sysadmin㉿kali)-[~]
$ nmap 172.16.1.0/28
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-16 12:06 EDT
Nmap scan report for 172.16.1.1
Host is up (0.00048s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http

Nmap scan report for 172.16.1.10
Host is up (0.00024s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https

Nmap done: 16 IP addresses (2 hosts up) scanned in 6.10 seconds
```

Once again, notice which ports are open on every host in the DMZ (including the *pfSense* firewall). Every open port presents an opportunity for hackers to break into your system. Without the firewall between the exterior and interior networks we would be dangerously exposed.

17. Leave the *Kali* machine open and continue to the next task.

2.4 Exporting Scan Logs from nmap

Now that you have run *nmap* on several hosts and networks, let's export the results of the scans for further analysis and documentation. *nmap* provides five different output formats for saving the scans. They are:

- **Interactive:** The default
- **Normal:** Similar to Interactive, but without the extra runtime information and is used for more in-depth analysis
- **XML:** Can be converted into an HTML page and can be imported into databases
- **Greppable:** Used to output the target host's scan on a single line
- **ScRipT Kidd|3 oUTpuT:** A humorous way of displaying the output

The two most commonly used for documentation and analysis are **Normal** and **XML**.



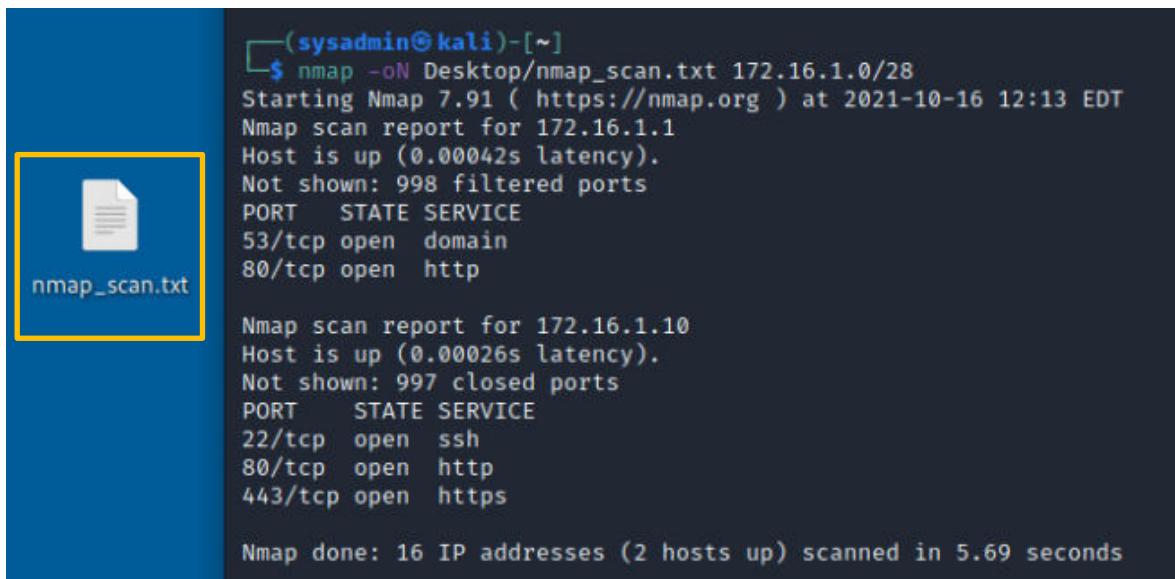
From the nmap.org website:

"Any security tool is only as useful as the output it generates. Complex tests and algorithms are of little value if they aren't presented in an organized and comprehensible fashion. Given the number of ways nmap is used by people and other software, no single format can please everyone. So nmap offers several formats, including the interactive mode for humans to read directly and XML for easy parsing by software."

<https://nmap.org/book/man-output.html>

1. Open a terminal session if there is not one already open.
2. Type the following command to generate the output to a text file on the desktop:

```
nmap -oN Desktop/nmap_scan.txt 172.16.1.0/28
```

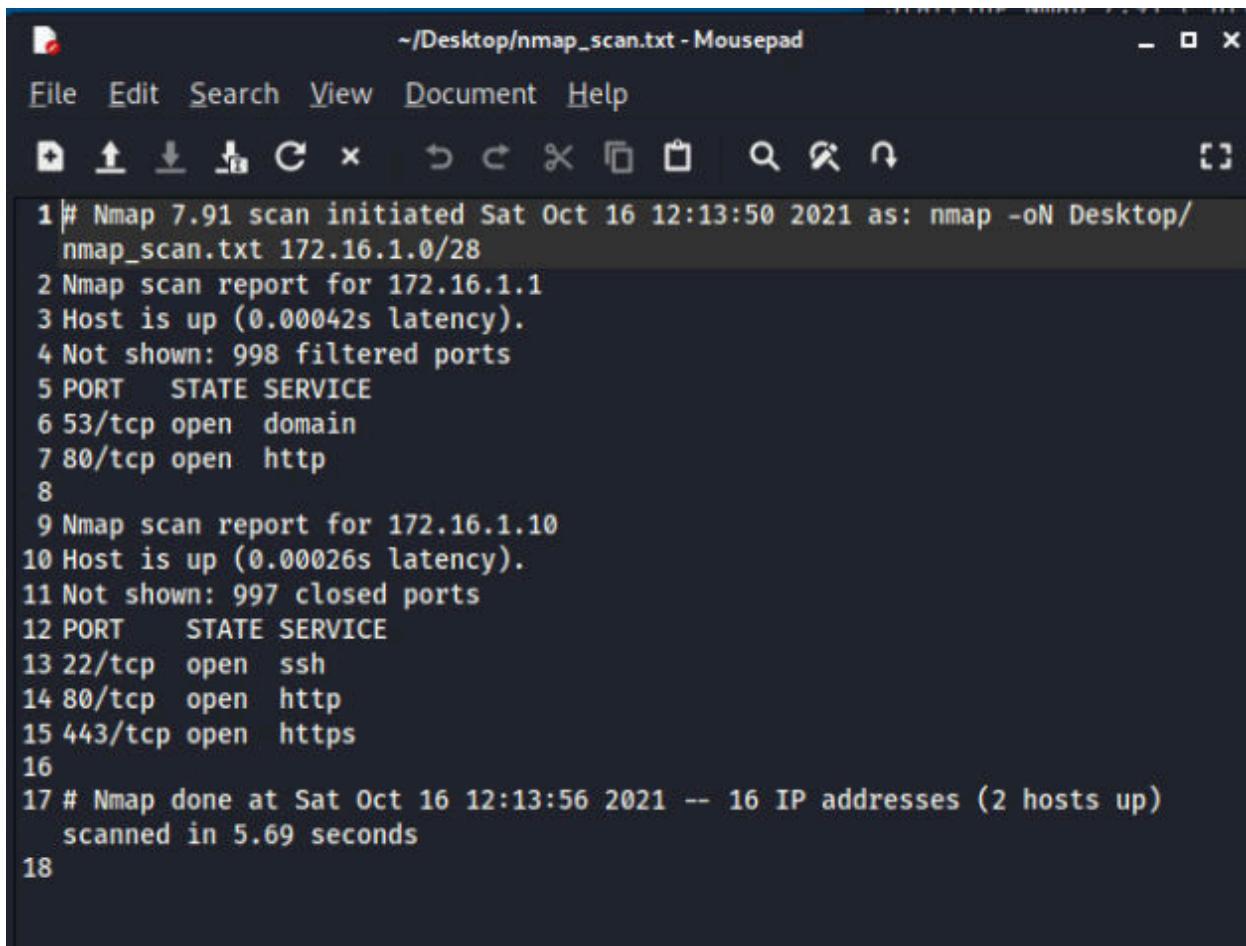


```
(sysadmin㉿kali)-[~]
$ nmap -oN Desktop/nmap_scan.txt 172.16.1.0/28
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-16 12:13 EDT
Nmap scan report for 172.16.1.1
Host is up (0.00042s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http

Nmap scan report for 172.16.1.10
Host is up (0.00026s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https

Nmap done: 16 IP addresses (2 hosts up) scanned in 5.69 seconds
```

3. On the desktop, you should see a file named *nmap_scan.txt*. This file contains the **Normal** output from the *nmap* scan. Double-click on the **nmap_scan.txt** file, and it will open in a text editor.



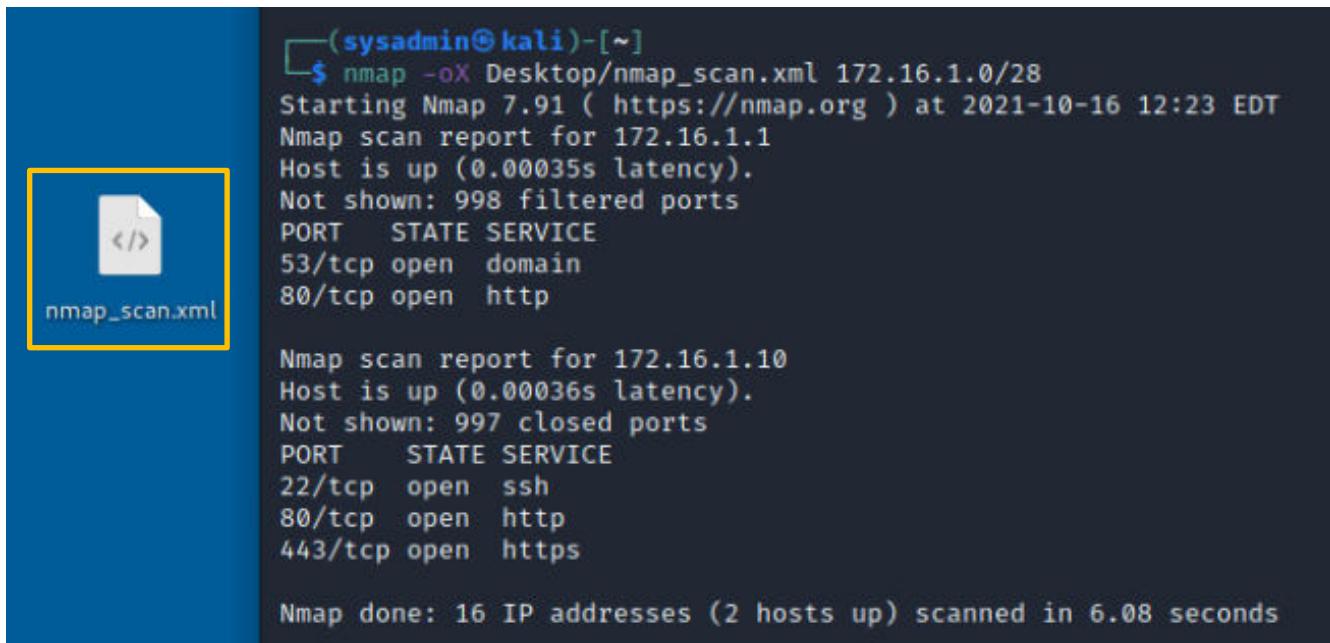
The screenshot shows a text editor window titled '~/Desktop/nmap_scan.txt - Mousepad'. The window has a dark theme with white text. The content of the file is a text-based Nmap scan report. The report includes details about two hosts: 172.16.1.0/28 and 172.16.1.10. It lists open ports (e.g., 53/tcp, 80/tcp, 443/tcp) and services (e.g., domain, http, https, ssh). The scan summary at the end indicates 16 IP addresses were scanned in 5.69 seconds, with 2 hosts up.

```
1 # Nmap 7.91 scan initiated Sat Oct 16 12:13:50 2021 as: nmap -oN Desktop/
nmap_scan.txt 172.16.1.0/28
2 Nmap scan report for 172.16.1.1
3 Host is up (0.00042s latency).
4 Not shown: 998 filtered ports
5 PORT      STATE SERVICE
6 53/tcp    open  domain
7 80/tcp    open  http
8
9 Nmap scan report for 172.16.1.10
10 Host is up (0.00026s latency).
11 Not shown: 997 closed ports
12 PORT      STATE SERVICE
13 22/tcp    open  ssh
14 80/tcp    open  http
15 443/tcp   open  https
16
17 # Nmap done at Sat Oct 16 12:13:56 2021 -- 16 IP addresses (2 hosts up)
scanned in 5.69 seconds
18
```

4. Close the text editor window by clicking the X in the upper-right corner.

5. Exporting the scan to an XML file, then converting the file to HTML will produce much more detailed output in a format that can be used for more extensive analysis and documentation. Type the following command to generate the output to an XML file on the desktop. On the desktop, you should see a file named *nmap_scan.xml*. This file contains the output from the *nmap* scan in XML format.

```
nmap -oX Desktop/nmap_scan.xml 172.16.1.0/28
```



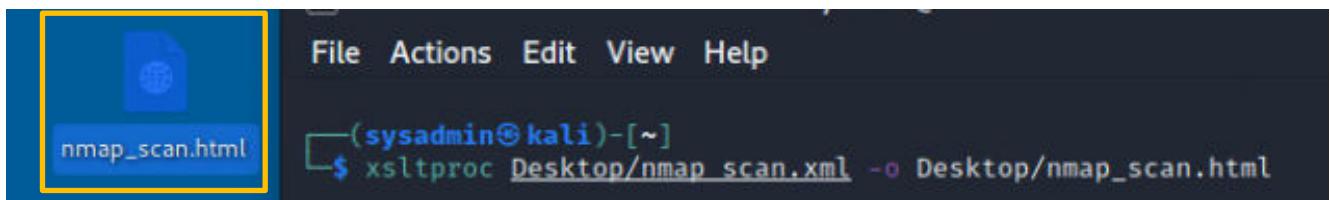
```
(sysadmin㉿kali)-[~]
└─$ nmap -oX Desktop/nmap_scan.xml 172.16.1.0/28
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-16 12:23 EDT
Nmap scan report for 172.16.1.1
Host is up (0.00035s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http

Nmap scan report for 172.16.1.10
Host is up (0.00036s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https

Nmap done: 16 IP addresses (2 hosts up) scanned in 6.08 seconds
```

6. To convert the XML to HTML, use the **xsltproc** command. Type the following command. On the desktop, you should see a file named *nmap_scan.html*. This file contains the converted XML file.

```
xsltproc Desktop/nmap_scan.xml -o Desktop/nmap_scan.html
```



```
File Actions Edit View Help
└─$ xsltproc Desktop/nmap_scan.xml -o Desktop/nmap_scan.html
```

7. Double-click on the **nmap_scan.html** file and it will open in a web browser window.

Nmap Scan Report - Scanned at Sat Oct 16 12:23:34 2021 - Mozilla Firefox

Nmap Scan Report - Scanned at Sat Oct 16 12:23:34 2021

Scan Summary | 172.16.1.1 | 172.16.1.10

Scan Summary

Nmap 7.91 was initiated at Sat Oct 16 12:23:34 2021 with these arguments:
`nmap -oX Desktop/nmap_scan.xml 172.16.1.0/28`

Verbosity: 0; Debug level 0

Nmap done at Sat Oct 16 12:23:40 2021; 16 IP addresses (2 hosts up) scanned in 6.08 seconds

172.16.1.1

Address

- 172.16.1.1 (ipv4)

Ports

The 998 ports scanned but not shown below are in state: **filtered**

- 998 ports replied with: **no-responses**

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
53	tcp open	domain	syn-ack			
80	tcp open	http	syn-ack			

[Go to top](#)
[Toggle Closed Ports](#)

Misc Metrics (click to expand)

172.16.1.10

Address

- 172.16.1.10 (ipv4)

Ports

The 997 ports scanned but not shown below are in state: **closed**

- 997 ports replied with: **conn-refused**

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
22	tcp open	ssh	syn-ack			
80	tcp open	http	syn-ack			
443	tcp open	https	syn-ack			

[Go to top](#)
[Toggle Closed Ports](#)
[Toggle Filtered Ports](#)

Misc Metrics (click to expand)

8. Close the *Firefox* browser window and the terminal window.
9. Leave the *Kali* machine open and continue to the next task.

3 Performing Host Scans with Legion

Legion is a very powerful GUI tool that you use to script different port scans and output the results for analysis and documentation.

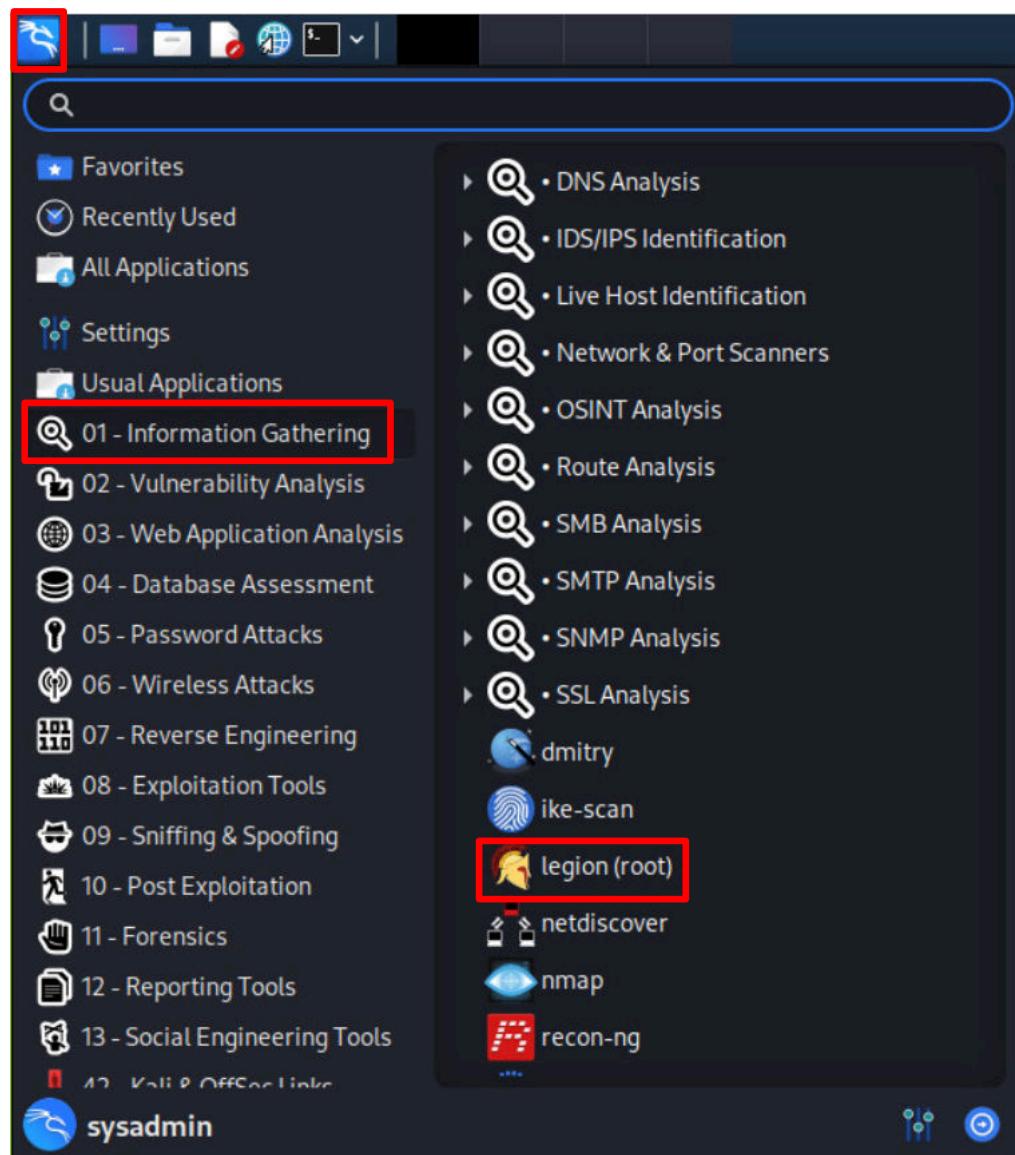


From the *Kali Linux Tutorial Site*:

"Legion, a fork of SECFORCE's Sparta, is an open source, easy-to-use, super-extensible and semi-automated network penetration testing framework that aids in discovery, reconnaissance and exploitation of information systems."

<https://kalilinuxtutorials.com/legion-penetration-testing/>

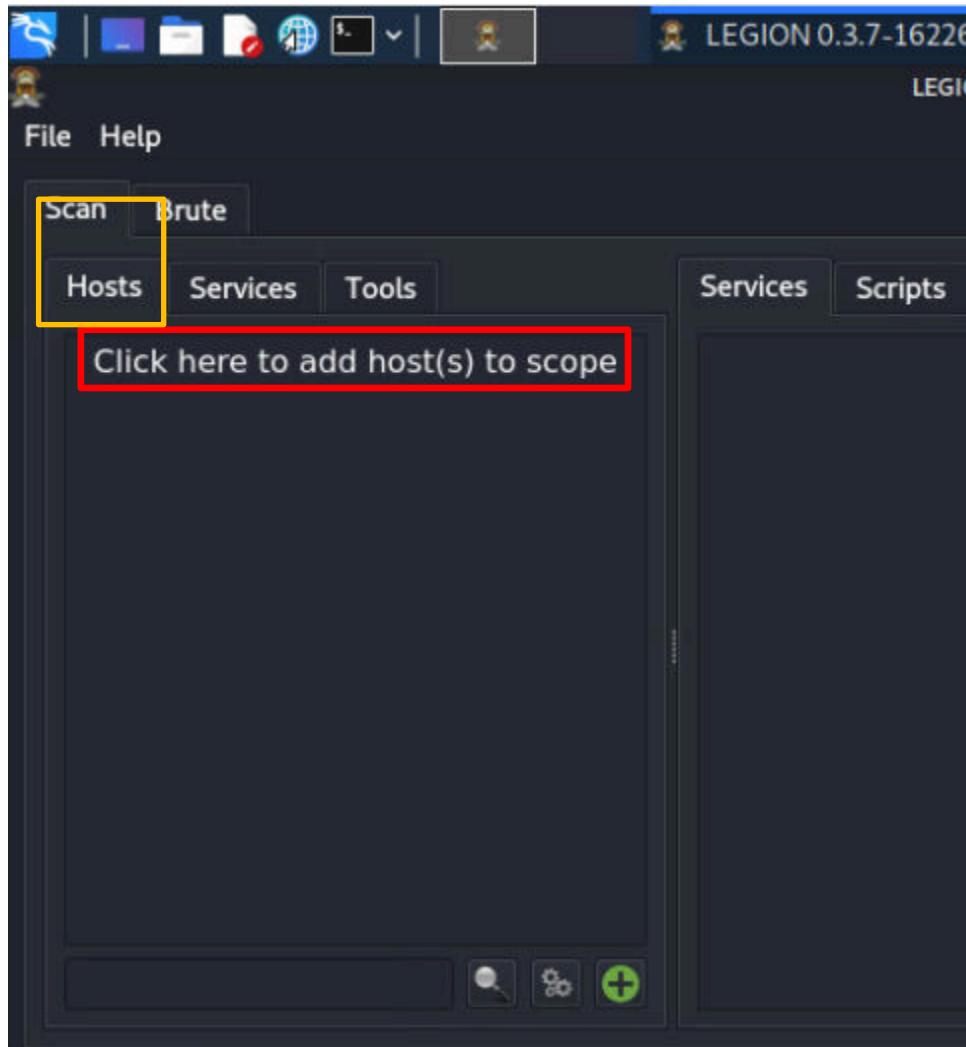
1. Click on the **Start** button in the upper-left corner of the window, click **01 Information Gathering** from the menu and select **legion (root)** from the submenu. When asked for the password for the password, use: NDGlabpass123!



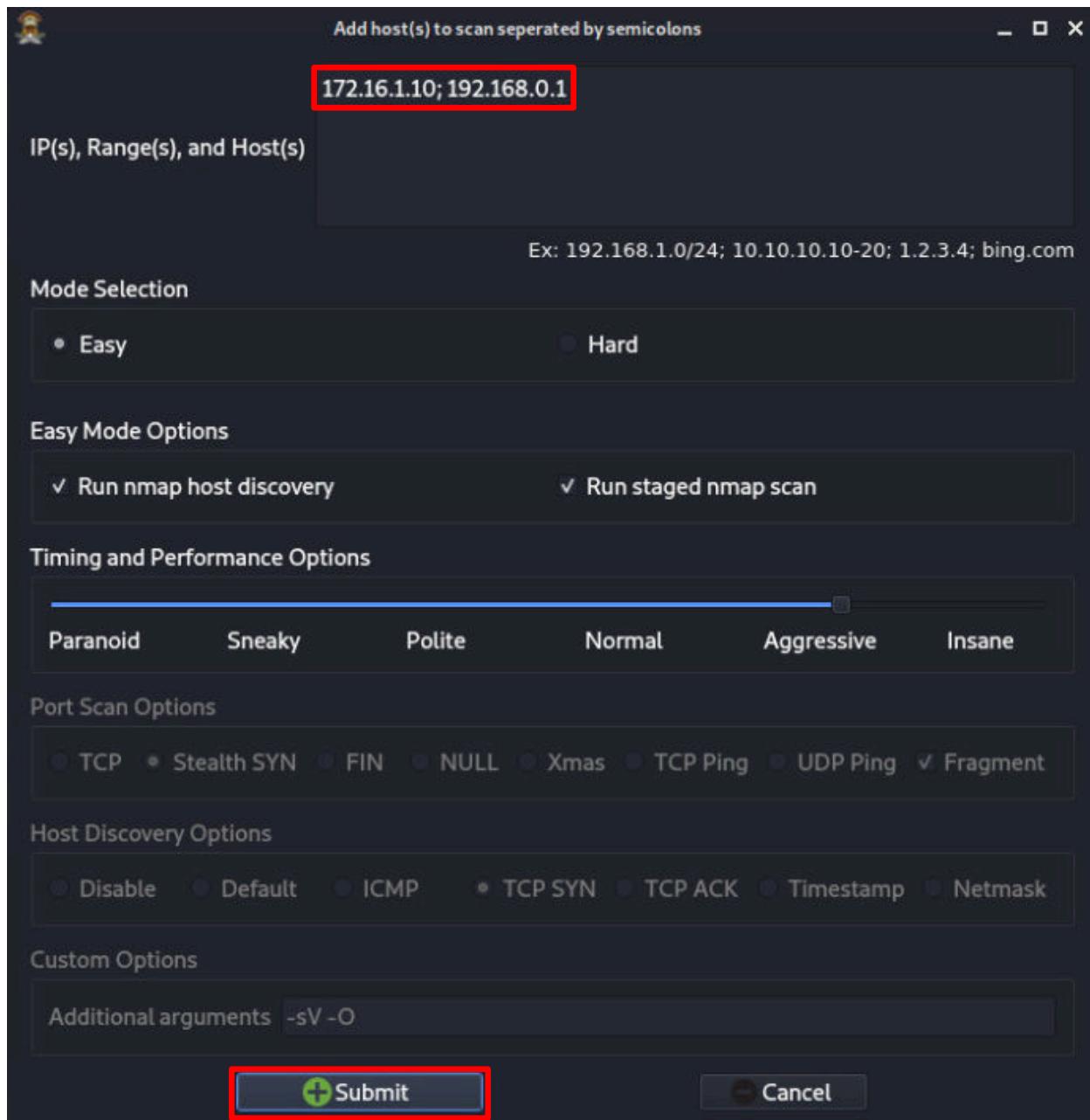


The Legion window may open and be too large for the desktop space. If this happens, right-click on the LEGION entry in the taskbar and select **Maximize**.

- With the **Scan** tab and **Hosts** tab selected, click on the text that says:
Click here to add host(s) to scope in the upper-left box.



3. In the *Add Host(s) to Scan* window, in the box for *IP(s), Range(s) and Host(s)*, type **172.16.1.10; 192.168.0.1**. Leave all the other settings at their default values and press the **Submit** button.



4. Once the scan is complete, click on the **172.16.1.10** host (which is *UbuntuSRV*) and under the *Services* tab you will see what ports are open, in this case, **443**, which is *HTTPS*, **80**, which is *HTTP* and **22**, which is *SSH*.



For each host, there are 6 *nmap* stages and it can take as long as 20 minutes to run all of the scans.

LEGION 0.3.7-1622656779 - untitled - /usr/share/legion/

Port	Protocol	State	Name	Version
22	tcp	open	ssh	OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
80	tcp	open	http	Apache httpd 2.4.41
443	tcp	open	http	Apache httpd 2.4.41 ((Ubuntu))

- Click on **192.168.0.1** host (which is the *pfSense* firewall), and under the *Services* tab, you will see what ports are open, in this case, **53**, which is **DNS**, and **80**, which is **HTTP**.

LEGION 0.3.7-1622656779 - untitled - /usr/share/legion/

Port	Protocol	State	Name	Version
53	tcp	open	domain	(generic dns response: REFUSED)
80	tcp	open	http	nginx

- The lab is now complete; you may end the reservation.