**Practical 4**

**Objectives:** Use nmap and other port scanning software

Try ping sweeps

Use hping to craft packets

**Exercise Use nmap and inspect the packets sent**

**Description :**

You will be running nmap against your web-server2 VM. Your web-server2 will have several services running, but not the Telnet server. The firewall on the web-server2 will allow clients to connect to a number of ports, including the Telnet port 23.

In web-server2

1. Check that the firewall configuration file /etc/firewalld/zones/public.xml contains the following line to allow incoming connections to Telnet port 23.

<service name= "telnet"/>

If the above line is not in the file, add it in. Run “systemctl restart firewalld” to restart the Firewall.

1. Edit the file /etc/xinetd.d/telnet. Change the following line to disable telnet

disable = yes

1. Restart the xinetd service if it is not already running.

systemctl restart xinetd

In Kali Linux

1. Look at the options available for nmap.

nmap -h

What is the option for running a SYN Scan? (Ans : -sS)

1. Start a Wireshark capture.
2. Use nmap to run a SYN scan against your web-server2.

Change to the IP of your web-server2

sudo nmap –sS 192.168.10.100

1. Stop Wireshark when nmap has finished running.

Table

Description automatically generated

Namp basically sends like tons of SYN packets to every possible port at random. Those port that reply with (ack) will mean that it is open

1. Look at the results of nmap. Port 21 and other ports are reported as “opened”, while Port 23 is reported as “closed”.
2. In Wireshark, apply the following filter to see only the SYN packets.



1. Scroll through the filtered packets to see the ports targeted. Look for any SYN packet going to port 21 (ftp).

You can either search manually or you can apply a filter



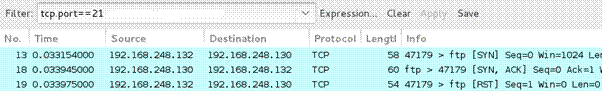
1. To see all the packets going to or coming from port 21, try the following filter :



How did your web-server2 respond to the SYN packet sent to the opened port 21? What kind of packet did it send back?

How did nmap respond to the packet sent by the web-server2 VM?

(Answer : example of a packet capture below

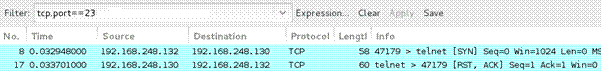


Packer 13 is the SYN packet sent to port 21. In Packet 18, the web-server2 responded with a SYN/ACK packet. In Packet 19, nmap responded with a RST packet.)

1. Change the filter to see all the packets going to or coming from port 23.

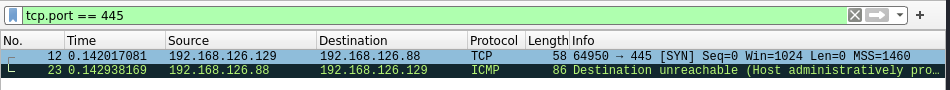
How did your web-server2 respond to the SYN packet sent to the closed port 23? What kind of packet did it send back?

(Answer : example of a packet capture below



In Packet 17, the web-server2 responded with a RST packet. This RST packet also has the ACK flag set.)

1. Change the filter to see all the packets going to or coming from port 445.



How did your web-server2 respond to the SYN packet sent to port 445 which is blocked by the firewall? Did it send any packet back?

(Ans : It did not send back any packet. Sometimes the firewall blocking the port may send back an ICMP destination unreachable packet)

Use the table below to record whether a SYN/ACK packet or a RST packet was returned or no packet was returned at all.

|  |  |
| --- | --- |
|  | Response from web-server2 |
| Port 21 (service running and not blocked by firewall) | SYN, ACK |
| Port 23 (service not running and port not blocked by firewall) | RST, ACK |
| Port 445 (service not running and port blocked by firewall) | No response |

**Exercise FIN Scan**

**Description :**

You will now use Nmap to send FIN packets to a range of ports on a range of IP addresses.

Pick a range of IP addresses that include your web-server2.

For example, if your web-server2 IP is 192.168.10.100, you can pick the range 192.168.10.97 to 192.168.10.102

In Kali Linux

1. Start a Wireshark capture.
2. Use nmap to run a FIN scan against your range of IP addresses. This time, to reduce network traffic, only scan ports 21,23,25,53, 80 and 110. For example, if your target network range is 192.168.10.97 – 192.168.10.102, you will run :

sudo nmap –sF –p 21,23,25,53,80,110 192.168.10.97-102

1. Stop the Wireshark capture when nmap has finished running.
2. Using the Wireshark capture, find out how the different systems in your range of IP addresses responded to the FIN packet on the different ports. Use the table below to record whether each system replied with a SYN/ACK or a RST packet or did not reply at all. Sometimes the firewall blocking the ports may send back an ICMP destination unreachable packet.

|  |  |  |
| --- | --- | --- |
|  | web-server2 |  |
| Port 21 | ICMP Destination unreachable |  |
| Port 23 | ICMP Destination unreachable |  |
| Port 25 | ICMP Destination unreachable |  |
| Port 53 | ICMP Destination unreachable |  |
| Port 80 | ICMP Destination unreachable |  |
| Port 110 | ICMP Destination unreachable |  |

**Exercise ACK Scan**

**Description :**

ACK scans can be used to detect the presence of stateful firewalls.

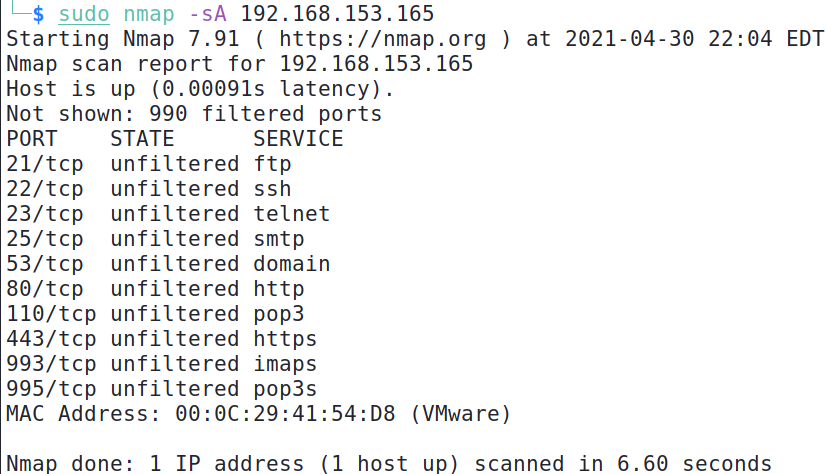
In Kali Linux

1. Start a Wireshark capture.
2. Run an ACK scan against your web-server2 VM. Currently the firewall on web-server2 is blocking some ports while allowing other ports.

Change to the IP of your web-server2

sudo nmap –sA 192.168.10.100

1. Stop Wireshark when nmap has finished running.
2. An ACK scan can be used to find if there a firewall blocking the ports. In the screenshot below, the ten ports listed as “unfiltered” are not blocked by the firewall. Another 990 ports have also been scanned and are listed as “filtered” (blocked by firewall).



990 ports found to be blocked by firewall (filtered)

These ports are not blocked by firewall (unfiltered)

1. Using the Wireshark capture, find out how your web-server2 VM responded to the ACK packets. Use the table below to record whether a RST packet was returned or no packets were returned at all. If a RST packet is received, the firewall allowed the ACK packet to that port to go through.

Sometimes the firewall blocking the ports may send back an ICMP destination unreachable packet.

|  |  |  |
| --- | --- | --- |
|  | Your web-server2 VM | Is the port blocked by firewall? |
| Port 21 | RST | No |
| Port 22 | RST | No |
| Port 23 | RST | No |
| Port 25 | RST | No |
| Port 445 | ACK | Yes |
| Poer 3389 | ACK | Yes |

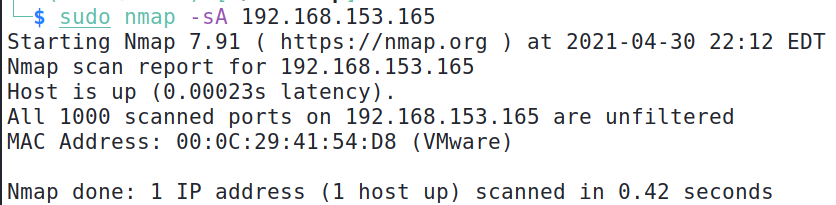
In web-server2

1. Run the following command to turn the firewall off.

systemctl stop firewalld

In Kali Linux

1. Start the Wireshark capture again. Repeat the ACK scan.
2. Stop Wireshark when nmap has finished running.
3. Using the Wireshark capture, find out how your web-server2 responded to the ACK packets. This time, with no firewall enabled, a RST packet should be returned to every ACK packet sent by nmap.



Nmap reports that all 1000 scanned ports are not blocked by firewall (unfiltered)

In web-server2

1. Run the following command to turn the firewall on again.

systemctl start firewalld

**Exercise UDP Scans**

**Description :**

UDP ports can also be opened on a system, and attackers can also scan for these opened UDP ports. Note that UDP port scanning can take much longer than TCP port scanning. This is because many opened UDP ports only responds to UDP packets containing specific data. When Nmap send generic UDP packets in the UDP scan, it may not receive replies from most of the UDP ports, and spends a while waiting for the time out.

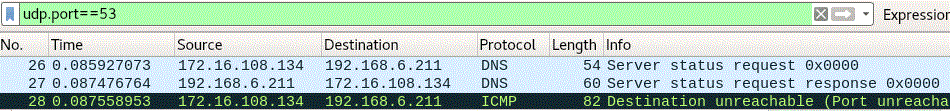
In Kali Linux

1. Start a Wireshark capture.
2. Run a UDP scan on ports 21,53, 80 and 161 against your web-server2.

Change to the IP of your web-server2

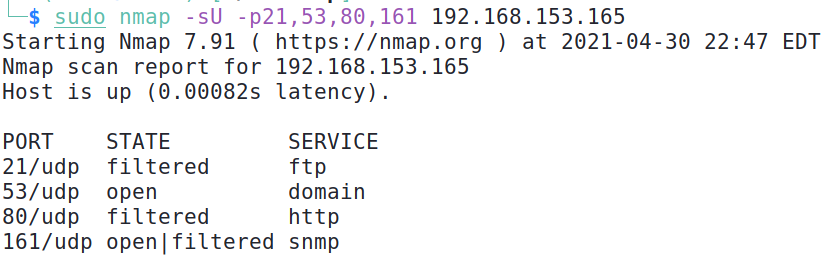
sudo nmap -sU –p 21,53,80,161 192.168.10.100

1. Stop Wireshark when nmap has finished running.
2. Which UDP ports are opened? What services are normally running on these ports?



(UDP Port 53 replied to the UDP scan)

1. In the screenshot below, UDP port 53 is reported to be opened. Nmap is not sure if UDP port 161 is opened or blocked by firewall, as it did not receive any reply from UDP port 161.



**Exercise Other nmap scans**

1. In Kali Linux, try other nmap options. For example, the -sV option to determine the version number of the service.

Change to the IP of your web-server2

sudo nmap -sV 192.168.10.100

1. Try the -sV option with UDP port scanning to increase the chances of finding out if the UDP ports are opened, plus the version number of the UDP service.

sudo nmap -sV –sU –p21,53,80,161 192.168.10.100

Change to the IP of your web-server2

1. If you do a UDP scan without specifying any ports, Nmap will scan 1000 most-frequently ports, which will take a long time

.

Change to the IP of your web-server2

sudo nmap -sU 192.168.10.100

Press Control-C to stop the UDP scan.

1. You can add the -v or -vv option (verbose or more verbose) to see what Nmap is doing during the scan.

.

Change to the IP of your web-server2

sudo nmap -sU –vv 192.168.10.100

Press Control-C to stop the UDP scan.

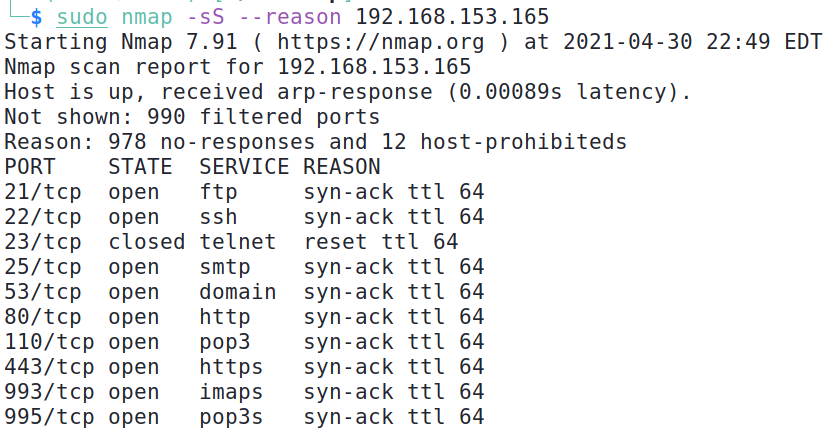
1. You can add the --reason option to see what kind of packet the scanned port returned (or did not return) for Nmap to make its deduction if the port is open, closed or filtered.

.

Change to the IP of your web-server2

sudo nmap -sS --reason 192.168.10.100

The following screenshot shows an example of using the --reason option.



990 ports are listed as filtered because 978 ports did not respond, and another 12 ports replied with an ICMP Destination Unreachable “Host Prohibited” packet

These ports are listed as open or closed because Nmap received these packets

1. Try the -O option to guess the operating system.

Change to the IP of your web-server2

sudo nmap -O 192.168.10.100

You can also try the nmap -O option against your Win10 VM or your Host PC. Note : it is only a best-guess of the operating system.

1. View the file /usr/share/nmap/nmap-services to see the list of common ports, their service names and frequencies.
2. Try the –top-ports option to scan the top 10 most frequently ports according to the nmap-services file.

Change to the IP of your web-server2

sudo nmap --top-ports 10 192.168.10.100

**Exercise Ping Sweeps and other methods to discover hosts in network**

**Description :**

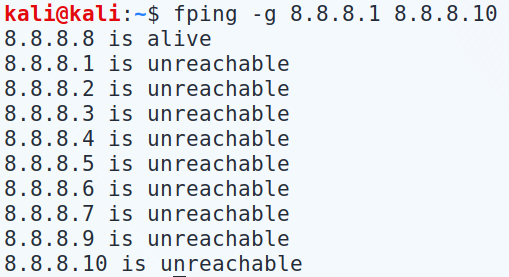
Fping can be used to ping a group of IP addresses. However, it may not always work as firewalls may block the ping packets.

Netdiscover and Nmap can be used to send ARP requests to discover alive hosts in the current network.

In Kali Linux

1. Use fping command to ping the range of addresses from 8.8.8.1 to 8.8.8.10. Which system is up?

fping -g 8.8.8.1 8.8.8.10



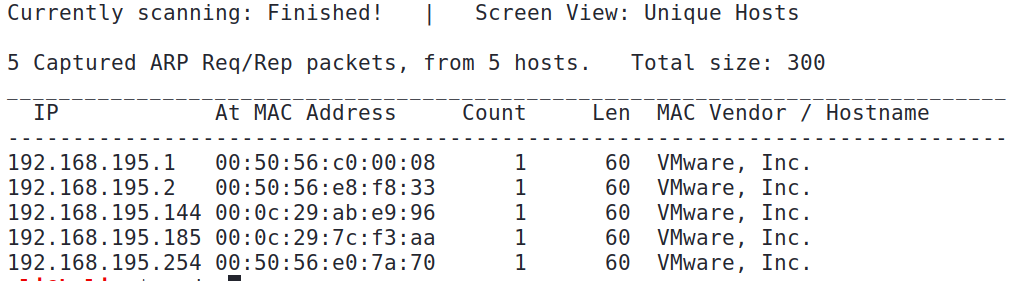
You may see results like this. 8.8.8.8 is the IP address of Google’s public DNS Server

1. Use Netdiscover to send ARP broadcast requests to the subnet of your Kali. Which system is up?

Replace with the subnet of your Kali.

For example, if your Kali IP is 192.168.23.4 with netmask 255.255.255.0 then replace with 192.168.23.0/24

sudo netdiscover –i eth0 –r 172.16.108.0/24



In this example, two IP addresses are discovered : 192.168.195.144 and 192.168.195.185.

192.168.195.1 is the Host PC

192.168.195.2 is the VMware Gateway and DNS Server

192.168.195.254 is the VMware DHCP Server

Note : Netdiscover may take a while to display the results.

1. Use Nmap to do host discovery by sending ARP broadcast requests to the local network. Which system is up?

Replace with the subnet of your Kali.

For example, if your Kali IP is 192.168.23.4 with netmask 255.255.255.0 then replace with 192.168.23.0/24

sudo nmap –sn 172.16.108.0/24

**Exercise Crafting packets**

You will use hping to craft ICMP packets with a spoofed source IP.

In Kali Linux

1. Start a Wireshark capture.
2. Type “icmp” in the Filter textbox to see only ICMP packets.
3. Use hping to send 2 ICMP packets to your Win10 VM or web-server2 VM.

Change to the IP of your Win10 or web-server2 VM

sudo hping3 –c 2 –-icmp 192.168.10.100

1. While Wireshark is running, look at the ICMP packets captured. Note that the ICMP echo request packets have the source IP of your Kali.
2. Use hping to send 2 ICMP packets to your Win10 VM or web-server2 VM with a spoofed IP

sudo hping3 –c 2 –a 192.168.20.20 –-icmp 192.168.10.100

Change to the IP of your Win10 or web-server2 VM

This is the fake IP

1. Look at the ICMP packets captured. This time the ICMP echo request packets have the fake source IP.

(Important : Do not scan other people’s systems or networks without their permission!)

Nah fam lol

*End of Practical*