CSEC-140 Introduction to Cybersecurity: Assignment (Week 9)

Instructions

1. Follow the instructions in MyCourses to duplicate this file in your shared ***INDIVIDUAL*** folder. Do not duplicate this file in your group folder!
2. Navigate to the assignment instructions document https://drive.google.com/file/d/1oJZqnTZVvLM0YSObSyHzUXqf1EnhtPel/view?usp=sharing
3. Complete each question as instructed in the document and paste screenshots below.
4. Respond to the questions below.

NOTE 1: Include all **SCREENSHOTS** requested in the assignment in your report. All **screenshots** must only depict useful information. Please no **Excessive whitespace or large screenshots.** Highlight all **useful information** and refer to them in your answers below.

IMPORTANT NOTE 2: When asked to provide the **captured traffic** in Wireshark; you need to take a **screenshot** of the PACKET LISTING window only showing all information from all columns clearly.

Activity 1 (SYN/Connect/FIN(Windows) Scans)

SYN

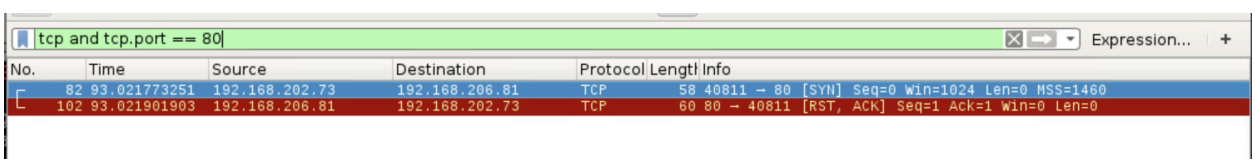
- (6 Points) A description of the scan technique, how it works, and when to use it (using your own words).

The SYN scan technique is used to see if ports are open, closed, or filtered. It works by sending a SYN packet to the target and waits for the response. You use it to see the state of a communications port.

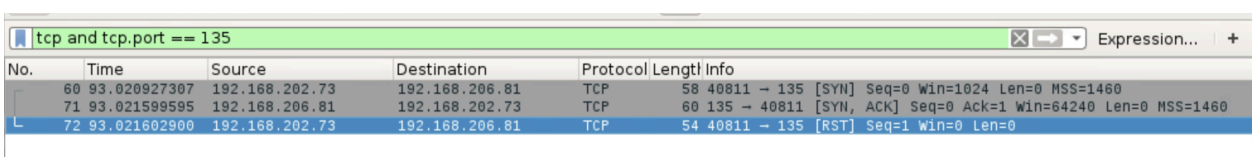
- (2 Points) The result of the scan shown on the terminal when running the nmap command

When running the command in the terminal it shows me the amount of closed ports, it tells me about a port's state and service.

- (4 Points) Captured traffic on Wireshark while scanning a closed port



- (4 Points) Captured traffic on Wireshark while scanning an open port



- (4 Points) Do the Wireshark captures match what you expected from this scan? Elaborate.

I was not expecting anything from the scan really. I can see the red line during the scan for the closed port though, so it may be a vulnerability.

Connect

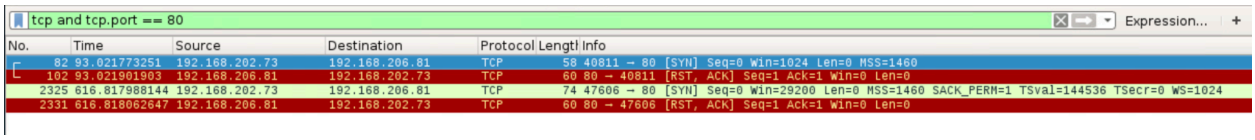
- (6 Points) A description of the scan technique, how it works, and when to use it (using your own words).

The connect scan technique also determines whether the ports are open, closed or filtered. It uses a handshake method which sends the SYN packet, waits for response, and depending on if its open or closed it will send either a TCP SYN-ACK packet or TCP RST packet.

- (2 Points) The result of the scan shown on the terminal when running the nmap command

It shows me the number of closed ports, the open ports and what their service is.

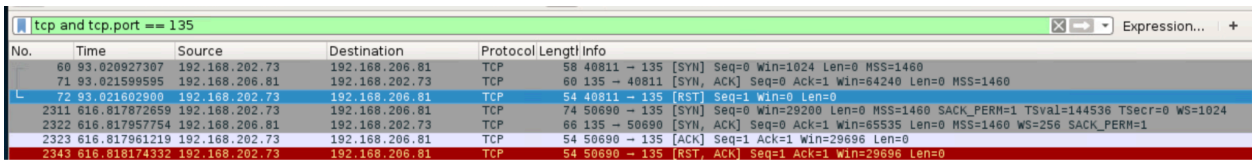
- (4 Points) Captured traffic on Wireshark while scanning a closed port



tcp and tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
82	93.021773251	192.168.202.73	192.168.206.81	TCP	58	40811 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
102	93.021901903	192.168.206.81	192.168.202.73	TCP	60	80 → 40811 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2325	616.817988144	192.168.202.73	192.168.206.81	TCP	74	47606 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=144536 TSecr=0 WS=1024
2331	616.818062647	192.168.206.81	192.168.202.73	TCP	60	80 → 47606 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

- (4 Points) Captured traffic on Wireshark while scanning an open port



tcp and tcp.port == 135

No.	Time	Source	Destination	Protocol	Length	Info
60	93.020927307	192.168.202.73	192.168.206.81	TCP	58	40811 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
71	93.021599595	192.168.206.81	192.168.202.73	TCP	60	135 → 40811 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
72	93.021602900	192.168.202.73	192.168.206.81	TCP	54	40811 → 135 [RST] Seq=1 Win=0 Len=0
2311	616.817872659	192.168.202.73	192.168.206.81	TCP	74	50690 → 135 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=144536 TSecr=0 WS=1024
2322	616.817957754	192.168.206.81	192.168.202.73	TCP	66	135 → 50690 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
2323	616.817961219	192.168.202.73	192.168.206.81	TCP	54	50690 → 135 [ACK] Seq=1 Ack=1 Win=29696 Len=0
2343	616.818174332	192.168.202.73	192.168.206.81	TCP	54	50690 → 135 [RST, ACK] Seq=1 Ack=1 Win=29696 Len=0

- (4 Points) Do the Wireshark captures match what you expected from this scan? Elaborate.

I did not expect there to be that many gray ports for the traffic on the open port. I'm not sure the meaning of the gray though. There is also way more red in the closed port. I wonder why each scan has a different number of red sources.

FIN(Windows)

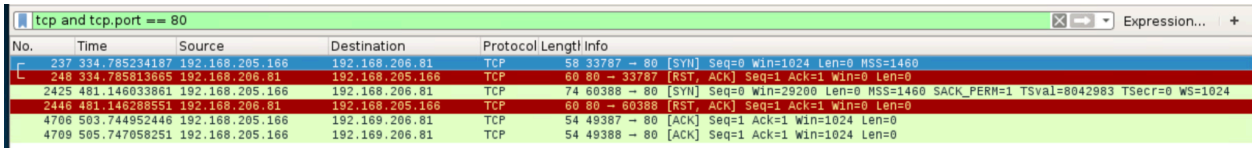
- (6 Points) A description of the scan technique, how it works, and when to use it (using your own words).

This scan technique is stealth scan which sends tcp packets with the FIN bit set to see if a port is closed. You would use it to see if the port is closed.

- (2 Points) The result of the scan shown on the terminal when running the nmap command

When the scan is over it shows how many ip addresses it did and the amount of time the scan took. It also gave me a note saying that the host seems down. It also shows me the starting nmap.

- (4 Points) Captured traffic on Wireshark while scanning a closed port



tcp and tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
237	334.785234187	192.168.205.166	192.168.206.81	TCP	58	33787 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
248	334.785813665	192.168.206.81	192.168.205.166	TCP	60	80 → 33787 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
2425	481.146033861	192.168.205.166	192.168.206.81	TCP	74	60388 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=8042983 TSecr=0 WS=1024
2446	481.146288551	192.168.206.81	192.168.205.166	TCP	60	80 → 60388 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4706	503.744952446	192.168.205.166	192.169.206.81	TCP	54	49387 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
4709	505.747058251	192.168.205.166	192.169.206.81	TCP	54	49388 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0

- (4 Points) Captured traffic on Wireshark while scanning an open port

tcp and tcp.port == 135

No.	Time	Source	Destination	Protocol	Length	Info
255	334.785869236	192.168.205.166	192.168.206.81	TCP	58	33787 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
276	334.786248308	192.168.206.81	192.168.205.166	TCP	60	135 → 33787 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
277	334.786252020	192.168.205.166	192.168.206.81	TCP	54	33787 → 135 [RST] Seq=1 Win=0 Len=0
2424	481.146019228	192.168.205.166	192.168.206.81	TCP	74	55580 → 135 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=8042983 TSecr=0 WS=1024
2444	481.146280652	192.168.206.81	192.168.205.166	TCP	66	135 → 55580 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
2445	481.146285365	192.168.205.166	192.168.206.81	TCP	54	55580 → 135 [ACK] Seq=1 Ack=1 Win=29696 Len=0
2450	481.146333266	192.168.205.166	192.168.206.81	TCP	54	55580 → 135 [RST, ACK] Seq=1 Ack=1 Win=29696 Len=0

- (4 Points) Do the Wireshark captures match what you expected from this scan? Elaborate.

To be honest, I didn't expect much from the scan, I did not know what it would show. But i can see some red lines and the ip address endings aren't the same for those ones. Maybe those are the weaker ports.

Activity 2 (<u>FIN(Linux) Scan</u>)

- (3 Points) A brief description of the difference between the scanning results of running a FIN scan against the Ubuntu VM vs running a FIN scan against the Windows target VM from **activity 1**. How are they different and why?

They are different because the FIN scan from activity 1 results had a note saying that the port seemed down. In this activity the FIN scan said open|filtered. It doesn't show the service name of any of the ports either.

- (3 Points) Why does nmap show 'open|filtered' instead of merely 'open' in the result of this scan?

It shows it because this scan probably shows the scanning results more in detail. The other nmap probably doesn't show because it's not as descriptive.

- (2 Points) The result of the scan shown on the terminal when running the nmap command

```
root@CSEC:~# nmap -sF 192.168.200.91

Starting Nmap 7.40 ( https://nmap.org ) at 2024-11-12 22:06 EST
Nmap scan report for 192.168.200.91
Host is up (0.0023s latency).
All 1000 scanned ports on 192.168.200.91 are open|filtered
MAC Address: 00:50:56:B0:69:73 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 21.54 seconds
root@CSEC:~#
```

- (4 Points) Captured traffic on Wireshark while scanning a closed port

tcp and tcp.port == 21

No.	Time	Source	Destination	Protocol	Length	Info
60	155.635262659	192.168.202.73	192.168.200.91	TCP	54	44456 → 21 [FIN] Seq=1 Win=1024 Len=0
61	156.736383142	192.168.202.73	192.168.200.91	TCP	54	44457 → 21 [FIN] Seq=1 Win=1024 Len=0

- (4 Points) Captured traffic on Wireshark while scanning an open/filtered port

tcp and tcp.port == 22

No.	Time	Source	Destination	Protocol	Length	Info
73	156.836608720	192.168.202.73	192.168.200.91	TCP	54	44456 → 22 [FIN] Seq=1 Win=1024 Len=0
87	156.936778540	192.168.202.73	192.168.200.91	TCP	54	44457 → 22 [FIN] Seq=1 Win=1024 Len=0

- (4 Points) Do the Wireshark captures match what you expected from this scan? Elaborate.

It did meet my expectations, I knew there wouldn't be any red sources because it said all the ports were open|filtered.

Activity 3 (<u>ACK Scan</u>)

- (6 Points) A description of the ACK scan technique, how it works, and when to use this type of scan (using your own words). The ACK scan technique uses ACK flagged packets to see if the port is filtered. It is used when trying to check if the firewall is actually protecting the host.

- (2 Points) The result of the scan shown on the terminal when running the nmap command

```
root@CSEC:~# nmap -sA 192.168.200.91
Starting Nmap 7.40 ( https://nmap.org ) at 2024-11-12 23:08 EST
Nmap scan report for 192.168.200.91
Host is up (0.0019s latency).
Not shown: 999 filtered ports
PORT    STATE      SERVICE
22/tcp unfiltered ssh
MAC Address: 00:50:56:B0:69:73 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.24 seconds
root@CSEC:~#
```

- (4 Points) Captured traffic on Wireshark while scanning a **filtered** port

tcp and tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
84	84.194942235	192.168.202.73	192.168.200.91	TCP	54	48095 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
113	84.295098083	192.168.202.73	192.168.200.91	TCP	54	48096 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0

- (4 Points) Captured traffic on Wireshark while scanning an **unfiltered** port

tcp and tcp.port == 22

No.	Time	Source	Destination	Protocol	Length	Info
76	84.192067519	192.168.202.73	192.168.200.91	TCP	54	48095 → 22 [ACK] Seq=1 Ack=1 Win=1024 Len=0
82	84.192730204	192.168.200.91	192.168.202.73	TCP	60	22 → 48095 [RST] Seq=1 Win=0 Len=0
441	85.494420194	192.168.202.73	192.168.200.91	TCP	54	48106 → 22 [ACK] Seq=1 Ack=1 Win=1024 Len=0
443	85.495059456	192.168.200.91	192.168.202.73	TCP	60	22 → 48106 [RST] Seq=1 Win=0 Len=0
1418	86.796946354	192.168.202.73	192.168.200.91	TCP	54	48107 → 22 [ACK] Seq=1 Ack=1 Win=1024 Len=0
1419	86.797545842	192.168.200.91	192.168.202.73	TCP	60	22 → 48107 [RST] Seq=1 Win=0 Len=0

- (4 Points) Do the Wireshark captures match what you expected from this scan? Elaborate.

I did not know what to expect but I can see that the unfiltered scan shows the ports with weak firewalls.