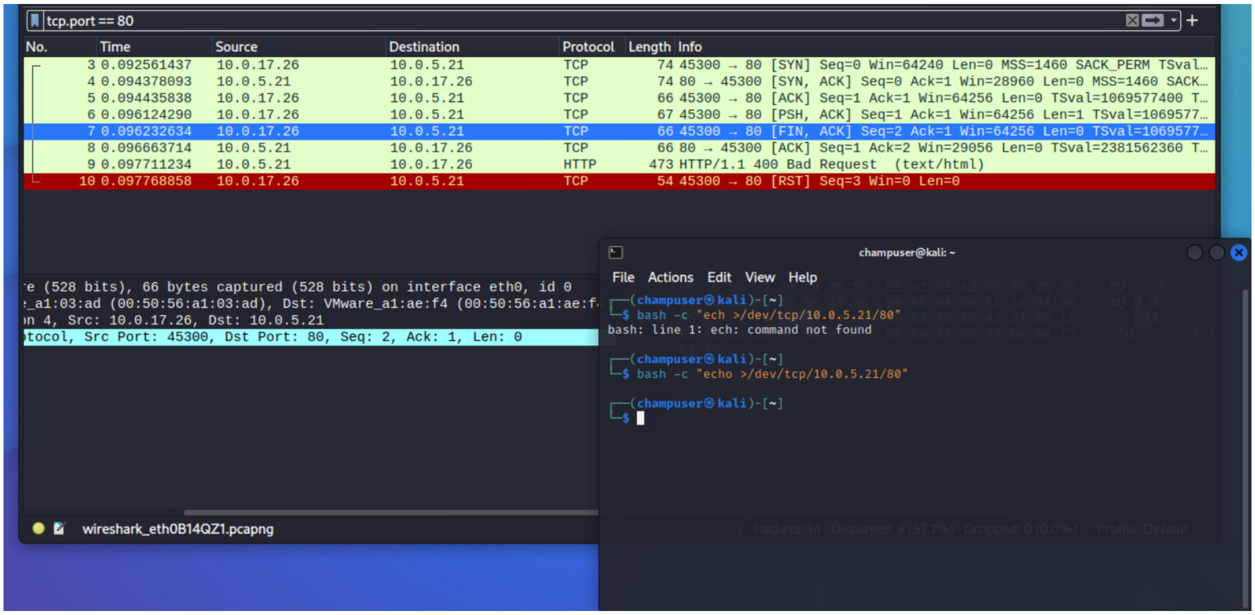
Deliverable 1. Provide a screenshot similar to the one above, make sure to take a look at the interplay of TCP flags during setup and teardown of the tcp connection.

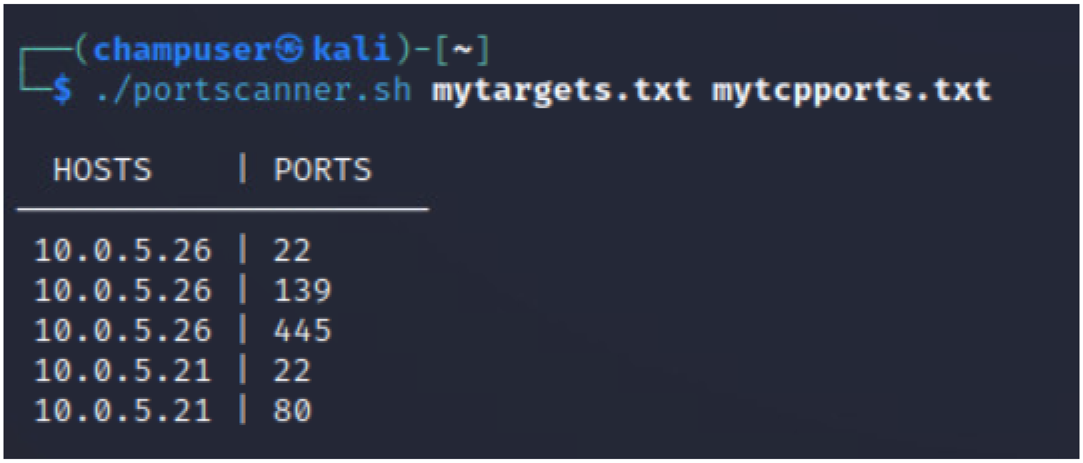

tcp.port == 80
champuser@kali: ~
File Actions Edit View Help
wireshark_eth0B14QZ1.pcapng
Packets: 14 · Displayed: 8 (57.1%) · Dropped: 0 (0.0%)
Profile: Default

```
No.  Time         Source       Destination  Protocol Length Info
  3 0.092561437  10.0.17.26   10.0.5.21    TCP       74 45300 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval…
  4 0.094378093  10.0.5.21    10.0.17.26   TCP       74 80 → 45300 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK…
  5 0.094435838  10.0.17.26   10.0.5.21    TCP       66 45300 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1069577400 T…
  6 0.096124290  10.0.17.26   10.0.5.21    TCP       67 45300 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=1 TSval=1069577…
  7 0.096232634  10.0.17.26   10.0.5.21    TCP       66 45300 → 80 [FIN, ACK] Seq=2 Ack=1 Win=64256 Len=0 TSval=1069577…
  8 0.096663714  10.0.5.21    10.0.17.26   TCP       66 80 → 45300 [ACK] Seq=1 Ack=2 Win=29056 Len=0 TSval=2381562360 T…
  9 0.097711234  10.0.5.21    10.0.17.26   HTTP     473 HTTP/1.1 400 Bad Request  (text/html)
 10 0.097768858  10.0.17.26   10.0.5.21    TCP       54 45300 → 80 [RST] Seq=3 Win=0 Len=0
```

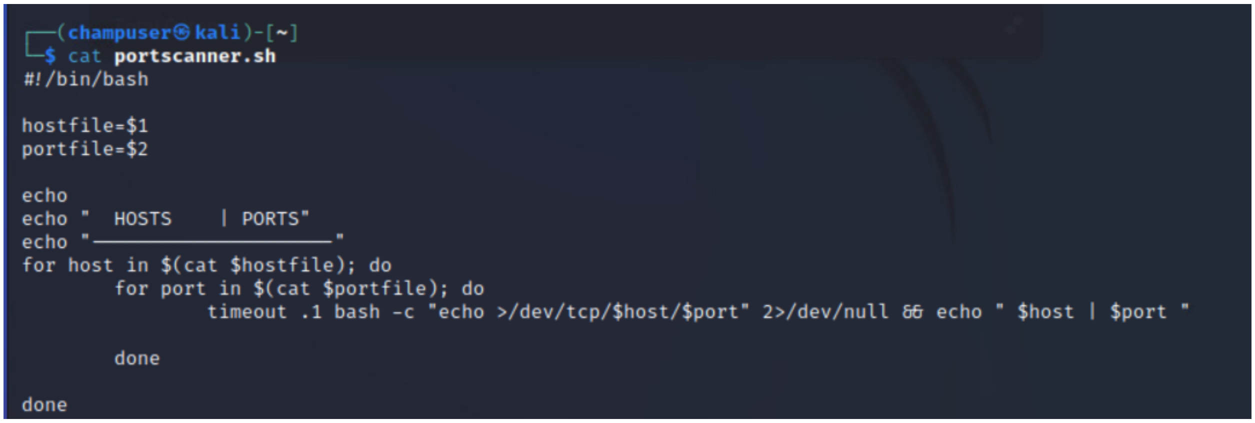
```
e (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
_a1:03:ad (00:50:56:a1:03:ad), Dst: VMware_a1:ae:f4 (00:50:56:a1:ae:f
n 4, Src: 10.0.17.26, Dst: 10.0.5.21
tocol, Src Port: 45300, Dst Port: 80, Seq: 2, Ack: 1, Len: 0
```

```
┌──(champuser㉿kali)-[~]
└─$ bash -c "ech >/dev/tcp/10.0.5.21/80"
bash: line 1: ech: command not found

┌──(champuser㉿kali)-[~]
└─$ bash -c "echo >/dev/tcp/10.0.5.21/80"

┌──(champuser㉿kali)-[~]
└─$ 
```

Deliverable 2. Execute your script (demo your enhancements as well), provide a source code listing (also upload this to your technical journal). Capture a screenshot of your program run similar to the one below. (Note, the ports may be different at the time of this lab)



```
┌──(champuser㉿kali)-[~]
└─$ ./portscanner.sh mytargets.txt mytcpports.txt

  HOSTS      | PORTS
______________________________

 10.0.5.26 | 22
 10.0.5.26 | 139
 10.0.5.26 | 445
 10.0.5.21 | 22
 10.0.5.21 | 80
```



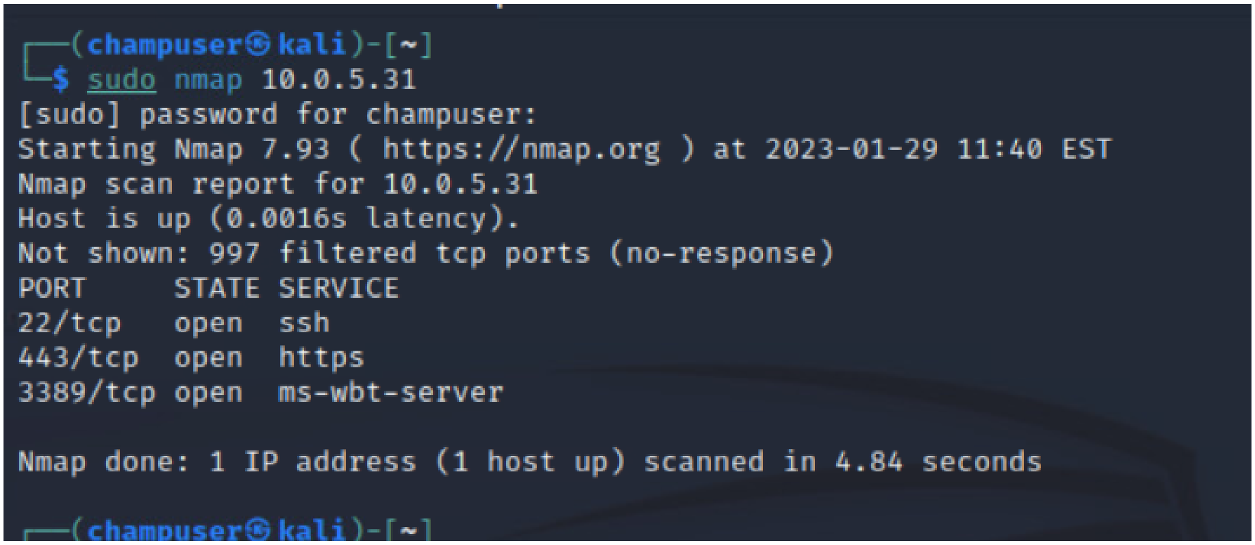
```
┌──(champuser㉿kali)-[~]
└─$ cat portscanner.sh
#!/bin/bash

hostfile=$1
portfile=$2

echo
echo "  HOSTS     | PORTS"
echo "______________________"
for host in $(cat $hostfile); do
        for port in $(cat $portfile); do
                timeout .1 bash -c "echo >/dev/tcp/$host/$port" 2>/dev/null && echo " $host | $port "

        done

done
```

Deliverable 3. So, you notice we target the file /dev/tcp/thehostip/thetcpport. Can you find this file in kali? Break out our friend google and see if you can find out what is going on. Briefly explain what you discover.

The file /dev/tcp/[host]/[port] is handled by bash exclusively, so you wont be able to see it in the kernel. Bash handels all its operations within the file /dev/tcp/[host]/[port] and writing to the file allows a TCP connection.

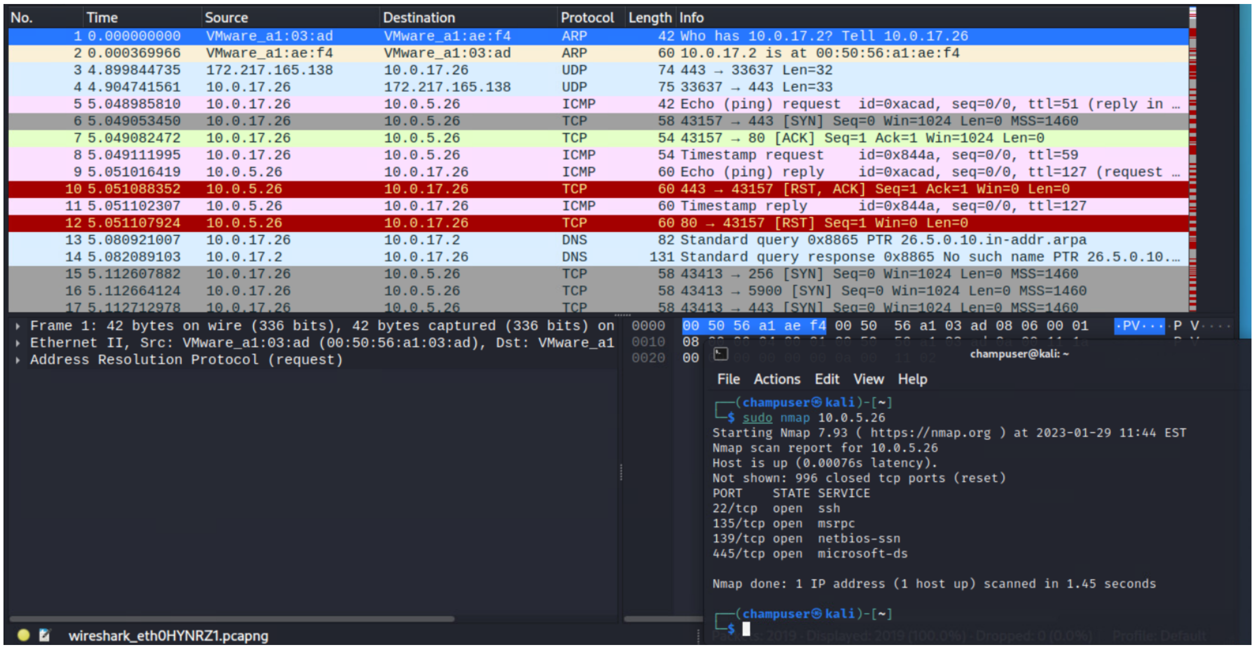
Source:
https://andreafortuna.org/2021/03/06/some-useful-tips-about-dev-tcp/#references

Deliverable 4. Provide a screenshot showing your nmap output



```
┌──(champuser㉿kali)-[~]
└─$ sudo nmap 10.0.5.31
[sudo] password for champuser:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-29 11:40 EST
Nmap scan report for 10.0.5.31
Host is up (0.0016s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
443/tcp  open  https
3389/tcp open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 4.84 seconds

┌──(champuser㉿kali)-[~]
```

Deliverable 5, find another open port, create the appropriate display filter and submit a screenshot similar to the example (but with another port).



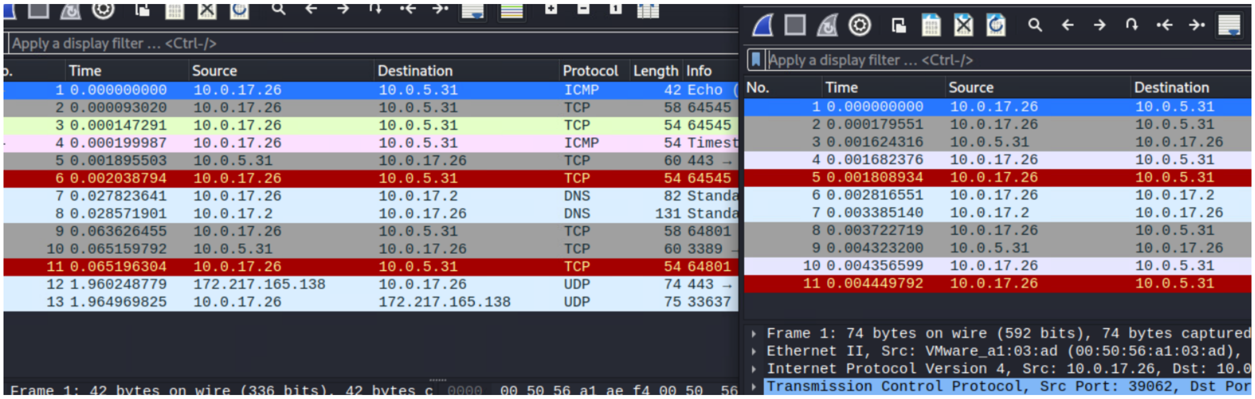
```
No. Time         Source            Destination       Protocol Length Info
  1 0.000000000  VMware_a1:03:ad   VMware_a1:ae:f4   ARP       42 Who has 10.0.17.2? Tell 10.0.17.26
  2 0.000369966  VMware_a1:ae:f4   VMware_a1:03:ad   ARP       60 10.0.17.2 is at 00:50:56:a1:ae:f4
  3 4.899844735  172.217.165.138   10.0.17.26        UDP       74 443 → 33637 Len=32
  4 4.904741561  10.0.17.26        172.217.165.138   UDP       75 33637 → 443 Len=33
  5 5.048985810  10.0.17.26        10.0.5.26         ICMP      42 Echo (ping) request  id=0xacad, seq=0/0, ttl=51 (reply in …
  6 5.049053450  10.0.17.26        10.0.5.26         TCP       58 43157 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  7 5.049082472  10.0.17.26        10.0.5.26         TCP       54 43157 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
  8 5.049111995  10.0.17.26        10.0.5.26         ICMP      54 Timestamp request    id=0x844a, seq=0/0, ttl=59
  9 5.051016419  10.0.5.26         10.0.17.26        ICMP      60 Echo (ping) reply    id=0xacad, seq=0/0, ttl=127 (request …
 10 5.051088352  10.0.5.26         10.0.17.26        TCP       60 443 → 43157 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 11 5.051102307  10.0.5.26         10.0.17.26        ICMP      60 Timestamp reply      id=0x844a, seq=0/0, ttl=127
 12 5.051107924  10.0.5.26         10.0.17.26        TCP       60 80 → 43157 [RST] Seq=1 Win=0 Len=0
 13 5.080921007  10.0.17.26        10.0.17.2         DNS       82 Standard query 0x8865 PTR 26.5.0.10.in-addr.arpa
 14 5.082089103  10.0.17.2         10.0.17.26        DNS      131 Standard query response 0x8865 No such name PTR 26.5.0.10.…
 15 5.112607882  10.0.17.26        10.0.5.26         TCP       58 43413 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 16 5.112664124  10.0.17.26        10.0.5.26         TCP       58 43413 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 17 5.112712978  10.0.17.26        10.0.5.26         TCP       58 43413 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

```
› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on
› Ethernet II, Src: VMware_a1:03:ad (00:50:56:a1:03:ad), Dst: VMware_a1
› Address Resolution Protocol (request)
```

```
0000  00 50 56 a1 ae f4 00 50  56 a1 03 ad 08 06 00 01   ·PV···P V····
0010  08
0020  00
```

```
┌──(champuser㉿kali)-[~]
└─$ sudo nmap 10.0.5.26
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-29 11:44 EST
Nmap scan report for 10.0.5.26
Host is up (0.00076s latency).
Not shown: 996 closed tcp ports (reset)
PORT    STATE SERVICE
22/tcp  open  ssh
135/tcp open  msrpc
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 1.45 seconds

┌──(champuser㉿kali)-[~]
└─$ 
```

Deliverable 6. Describe the difference in the two wireshark captures



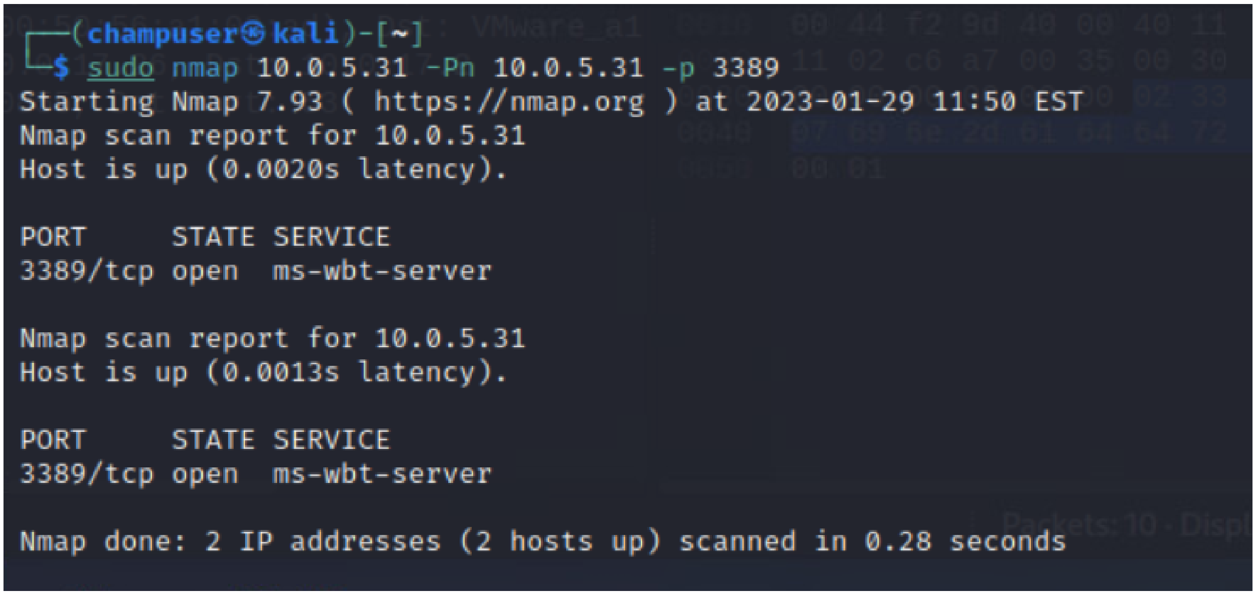
Apply a display filter … <Ctrl-/>
Apply a display filter … <Ctrl-/>

```
o. Time         Source            Destination       Protocol Length Info
  1 0.000000000  10.0.17.26        10.0.5.31         ICMP      42 Echo (
  2 0.000093020  10.0.17.26        10.0.5.31         TCP       58 64545
  3 0.000147291  10.0.17.26        10.0.5.31         TCP       54 64545
  4 0.000199987  10.0.17.26        10.0.5.31         ICMP      54 Timest
  5 0.001895503  10.0.5.31         10.0.17.26        TCP       60 443 →
  6 0.002038794  10.0.17.26        10.0.5.31         TCP       54 64545
  7 0.027823641  10.0.17.26        10.0.17.2         DNS       82 Standa
  8 0.028571901  10.0.17.2         10.0.17.26        DNS      131 Standa
  9 0.063626455  10.0.17.26        10.0.5.31         TCP       58 64801
 10 0.065159792  10.0.5.31         10.0.17.26        TCP       60 3389 
 11 0.065196304  10.0.17.26        10.0.5.31         TCP       54 64801
 12 1.960248779  172.217.165.138   10.0.17.26        UDP       74 443 →
 13 1.964969825  10.0.17.26        172.217.165.138   UDP       75 33637

Frame 1: 42 bytes on wire (336 bits), 42 bytes c  0000  00 50 56 a1 ae f4 00 50  56
```

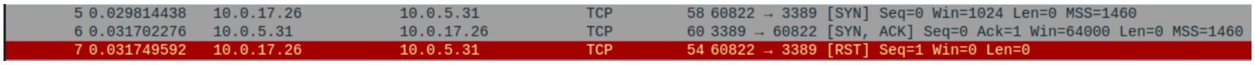
```
No. Time         Source        Destination
  1 0.000000000  10.0.17.26    10.0.5.31
  2 0.000179551  10.0.17.26    10.0.5.31
  3 0.001624316  10.0.5.31     10.0.17.26
  4 0.001682376  10.0.17.26    10.0.5.31
  5 0.001808934  10.0.17.26    10.0.5.31
  6 0.002816551  10.0.17.26    10.0.17.2
  7 0.003385140  10.0.17.2     10.0.17.26
  8 0.003722719  10.0.17.26    10.0.5.31
  9 0.004323200  10.0.5.31     10.0.17.26
 10 0.004356599  10.0.17.26    10.0.5.31
 11 0.004449792  10.0.17.26    10.0.5.31

› Frame 1: 74 bytes on wire (592 bits), 74 bytes captured
› Ethernet II, Src: VMware_a1:03:ad (00:50:56:a1:03:ad),
› Internet Protocol Version 4, Src: 10.0.17.26, Dst: 10.0
› Transmission Control Protocol, Src Port: 39062, Dst Por
```

The difference between the sudo and non-sudo nmap scan was the TCP handshake at the beginning of the sudo one.

Deliverable 7. Add the -Pn flag and provide a wireshark display. With no display filter, you should have a total of 3 packets and evidence of a simple SYN scan similar to the one below.



```
┌──(champuser㉿kali)-[~]
└─$ sudo nmap 10.0.5.31 -Pn 10.0.5.31 -p 3389
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-29 11:50 EST
Nmap scan report for 10.0.5.31
Host is up (0.0020s latency).

PORT     STATE SERVICE
3389/tcp open  ms-wbt-server

Nmap scan report for 10.0.5.31
Host is up (0.0013s latency).

PORT     STATE SERVICE
3389/tcp open  ms-wbt-server

Nmap done: 2 IP addresses (2 hosts up) scanned in 0.28 seconds
```



```
5 0.029814438   10.0.17.26      10.0.5.31       TCP     58 60822 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
6 0.031702276   10.0.5.31       10.0.17.26      TCP     60 3389 → 60822 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460
7 0.031749592   10.0.17.26      10.0.5.31       TCP     54 60822 → 3389 [RST] Seq=1 Win=0 Len=0
```

Deliverable 8. Provide links to any source code written in accomplishing this lab's objectives (remember, you can collaborate with your teammates on this). If you were asked to write a script (more than a line), make sure this is an actual file uploaded to the source part of github as opposed to a wiki entry (though you can certainly link to this file in your wiki).

https://github.com/dthomsen116/SEC-335/wiki/Lab-2.1---Port-Scanning-1/