

OLD DOMINION UNIVERSITY

CYSE 301 CYBERSECURITY TECHNIQUES AND OPERATIONS

---

## Assignment #2 Traffic Tracing and Sniffing

---

Jo Brown

01196622

# Task A

1. How many packets are displayed? How many total?

|     |              |               |               |     |                                                                   |
|-----|--------------|---------------|---------------|-----|-------------------------------------------------------------------|
| 118 | 57.741937000 | 192.168.217.3 | 192.168.217.2 | DNS | 95 Standard query 0x496f A content-signature-2.cdn.mozilla.net    |
| 119 | 57.741956000 | 192.168.217.3 | 192.168.217.2 | DNS | 95 Standard query 0x6f85 AAAA content-signature-2.cdn.mozilla.net |
| 120 | 57.743428000 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x496f Refused                         |
| 121 | 57.743427500 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x6f85 Refused                         |
| 122 | 58.106327300 | 192.168.217.3 | 192.168.217.2 | DNS | 88 Standard query 0xa2e5 A contile.services.mozilla.com           |
| 123 | 58.106805800 | 192.168.217.3 | 192.168.217.2 | DNS | 88 Standard query 0x41c2 AAAA contile.services.mozilla.com        |
| 124 | 58.108113300 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0xa2e5 Refused                         |
| 125 | 58.108120700 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x41c2 Refused                         |
| 126 | 58.422198000 | 192.168.217.3 | 192.168.217.2 | DNS | 87 Standard query 0x6e40 AAAA safebrowsing.googleapis.com         |
| 127 | 58.422218700 | 192.168.217.3 | 192.168.217.2 | DNS | 87 Standard query 0x6e40 AAAA safebrowsing.googleapis.com         |
| 128 | 58.428400000 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x7691 Refused                         |
| 129 | 58.428407400 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x6e40 Refused                         |
| 130 | 58.653228000 | 192.168.217.3 | 192.168.217.2 | DNS | 79 Standard query 0x489a A spocs.getpocket.com                    |
| 131 | 58.653252100 | 192.168.217.3 | 192.168.217.2 | DNS | 79 Standard query 0x76b8 AAAA spocs.getpocket.com                 |
| 132 | 58.655511400 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x489a Refused                         |
| 133 | 58.655519800 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x76b8 Refused                         |
| 134 | 59.980820500 | 192.168.217.3 | 192.168.217.2 | DNS | 85 Standard query 0x6ab3 A push.services.mozilla.com              |
| 135 | 59.980847800 | 192.168.217.3 | 192.168.217.2 | DNS | 85 Standard query 0xa0d1 AAAA push.services.mozilla.com           |
| 136 | 59.980160700 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0x6ab3 Refused                         |
| 137 | 59.980168200 | 192.168.217.2 | 192.168.217.3 | DNS | 54 Standard query response 0xa0d1 Refused                         |

A total of 137 packets were captured, all packets being displayed

2. Filter with ICMP, repeat 1

|    |              |               |               |      |                                                                     |
|----|--------------|---------------|---------------|------|---------------------------------------------------------------------|
| 40 | 18.025899900 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=19/4864, ttl=64 (reply in 41) |
| 41 | 18.027299800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=19/4864, ttl=63 (request in 40) |
| 42 | 19.028017700 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=20/5120, ttl=64 (reply in 43) |
| 43 | 19.029935800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=20/5120, ttl=63 (request in 42) |
| 44 | 20.029052100 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=21/5376, ttl=64 (reply in 45) |
| 45 | 20.031708800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=21/5376, ttl=63 (request in 44) |
| 46 | 21.031146100 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=22/5632, ttl=64 (reply in 47) |
| 47 | 21.034591800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=22/5632, ttl=63 (request in 46) |
| 48 | 22.033143300 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=23/5888, ttl=64 (reply in 49) |
| 49 | 22.035251500 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=23/5888, ttl=63 (request in 48) |
| 50 | 23.033825500 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=24/6144, ttl=64 (reply in 51) |
| 51 | 23.035981800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=24/6144, ttl=63 (request in 50) |
| 52 | 24.035438900 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=25/6400, ttl=64 (reply in 53) |
| 53 | 24.037994500 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=25/6400, ttl=63 (request in 52) |
| 54 | 25.037360900 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=26/6656, ttl=64 (reply in 55) |
| 55 | 25.039580100 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=26/6656, ttl=63 (request in 54) |
| 56 | 26.039085100 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=27/6912, ttl=64 (reply in 57) |
| 57 | 26.040721900 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=27/6912, ttl=63 (request in 56) |
| 58 | 27.041123200 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=28/7168, ttl=64 (reply in 59) |
| 59 | 27.043995900 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=28/7168, ttl=63 (request in 58) |

Filtering to only ICMP packets, out of 137 captured packets, only 59 are displayed

3. Select an Echo (reply) message from the list. What are the source and destination IPs of this packet? What are the sequence number and the size of the data? What is the response time?

|    |              |               |               |      |                                                                     |
|----|--------------|---------------|---------------|------|---------------------------------------------------------------------|
| 40 | 18.025899900 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=19/4864, ttl=64 (reply in 41) |
| 41 | 18.027299800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=19/4864, ttl=63 (request in 40) |
| 42 | 19.028017700 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=20/5120, ttl=64 (reply in 43) |
| 43 | 19.029935800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=20/5120, ttl=63 (request in 42) |
| 44 | 20.029052100 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=21/5376, ttl=64 (reply in 45) |
| 45 | 20.031708800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=21/5376, ttl=63 (request in 44) |
| 46 | 21.031146100 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=22/5632, ttl=64 (reply in 47) |
| 47 | 21.034591800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=22/5632, ttl=63 (request in 46) |
| 48 | 22.033143300 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=23/5888, ttl=64 (reply in 49) |
| 49 | 22.035251500 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=23/5888, ttl=63 (request in 48) |
| 50 | 23.033825500 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=24/6144, ttl=64 (reply in 51) |
| 51 | 23.035981800 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=24/6144, ttl=63 (request in 50) |
| 52 | 24.035438900 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=25/6400, ttl=64 (reply in 53) |
| 53 | 24.037994500 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=25/6400, ttl=63 (request in 52) |
| 54 | 25.037360900 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=26/6656, ttl=64 (reply in 55) |
| 55 | 25.039580100 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=26/6656, ttl=63 (request in 54) |
| 56 | 26.039085100 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=27/6912, ttl=64 (reply in 57) |
| 57 | 26.040721900 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=27/6912, ttl=63 (request in 56) |
| 58 | 27.041123200 | 192.168.217.3 | 192.168.10.18 | ICMP | 98 Echo (ping) request id=0x0eab, seq=28/7168, ttl=64 (reply in 59) |
| 59 | 27.043995900 | 192.168.10.18 | 192.168.217.3 | ICMP | 98 Echo (ping) reply id=0x0eab, seq=28/7168, ttl=63 (request in 58) |

Frame 59: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0

Ethernet II, Src: Microsoft 40:57:38 (00:15:5d:40:57:38), Dst: Microsoft 40:57:27 (00:15:5d:40:57:27)

Internet Protocol Version 4, Src: 192.168.10.18, Dst: 192.168.217.3

Internet Control Message Protocol

0000 00 15 5d 40 57 27 00 15 5d 40 57 38 00 00 45 00 ...]0x15:5d:40:57:38 [0x00:15:5d:40:57:27

0010 00 54 cd b7 00 00 3f 01 49 8b c0 a8 0a 12 c0 a8 ...T...?I...

0020 d9 03 00 00 3d 27 6e ab 00 1c 2f f7 b6 67 00 00 ......=n.../g...

0030 00 00 a1 df 0d 00 00 00 00 00 10 11 12 13 14 15 ........I\*%\$%

0040 10 17 10 10 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ........I\*%\$%

0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &()\*+,-./012345

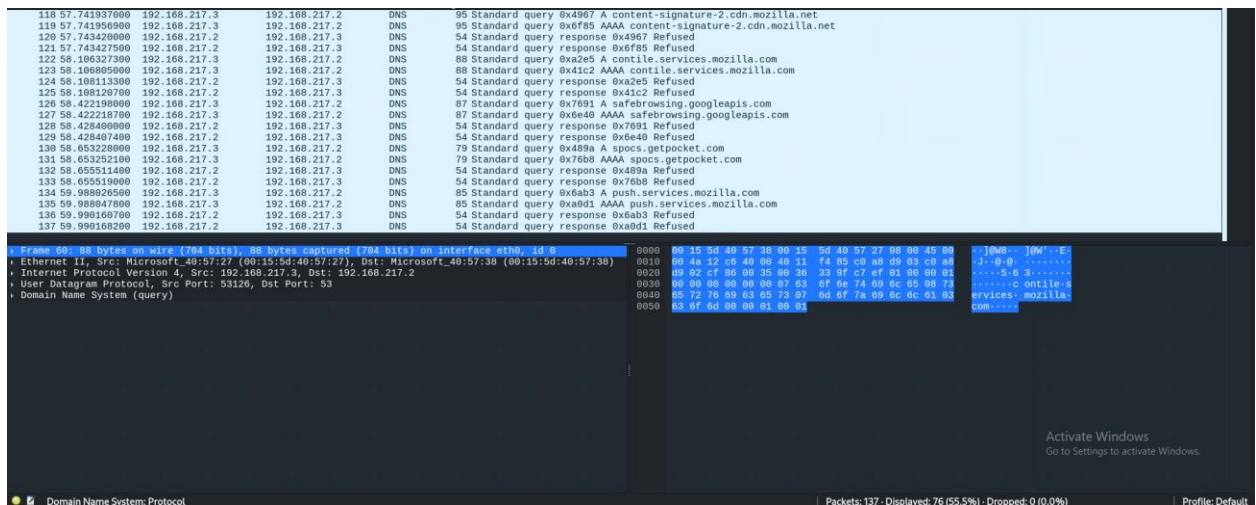
0060 36 37 67

Source: 192.168.10.18 Destination: 192.168.217.31

Sequence No: 28/7168 Size: 98 bytes

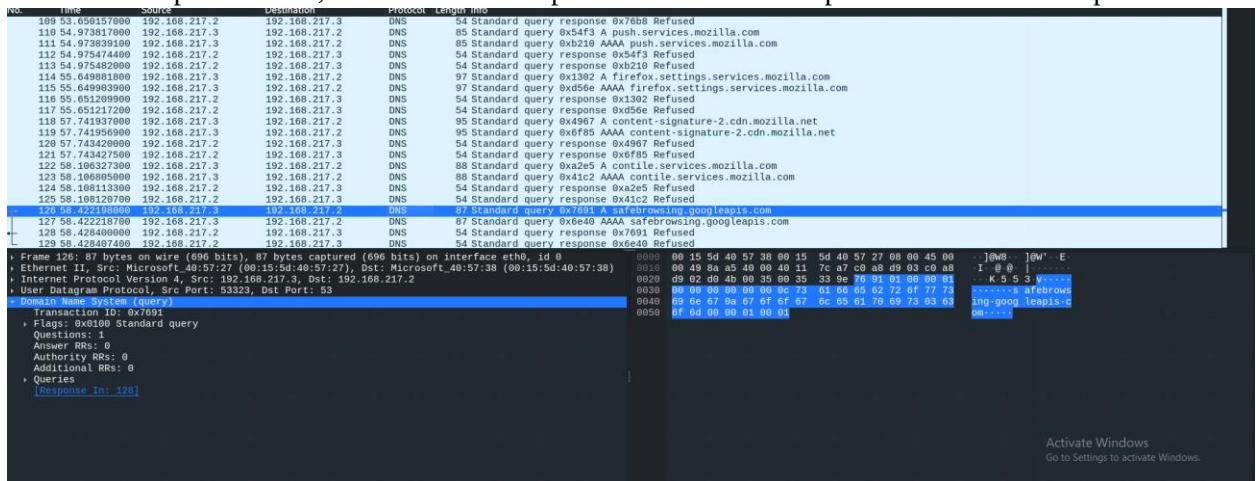
Response time: 58ms

4. Apply “DNS” as a display filter in Wireshark. How many packets are displayed?



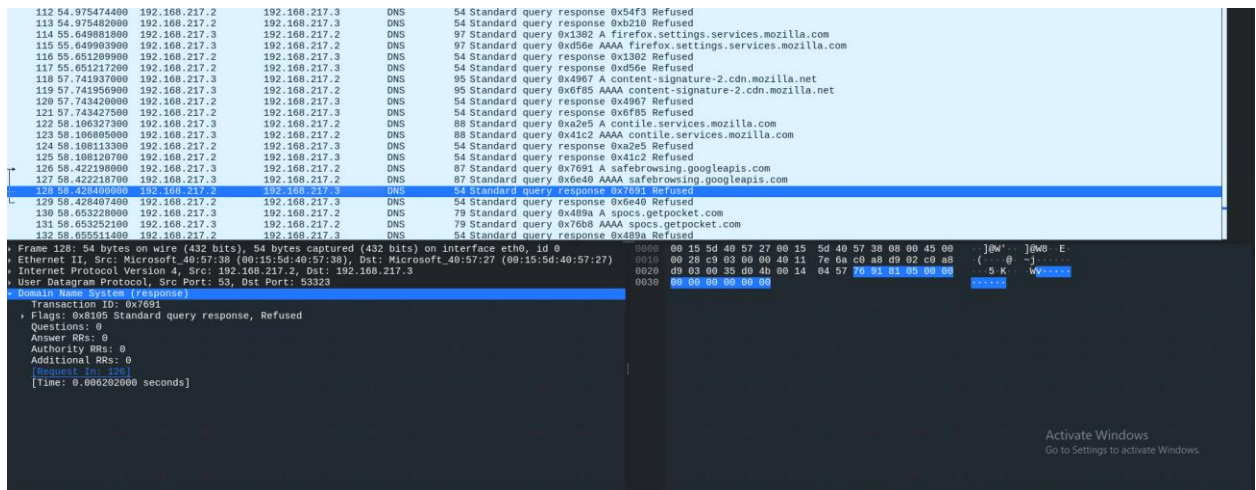
Of the 137 packets, 74 are displayed.

- Find a DNS query packet. What is the domain name this host is trying to resolve? What is the source IP and port number, destination IP and port number? Please express in the format: IP:port.



Safebrowsing.googleapis.com  
Source – 192.168.217.3:53323  
Destination – 192.168.217.2:53

- Find the corresponding DNS response to the query you selected at the previous step, and what is the source IP and port number, destination IP and port number? What is the message replied from the DNS server?



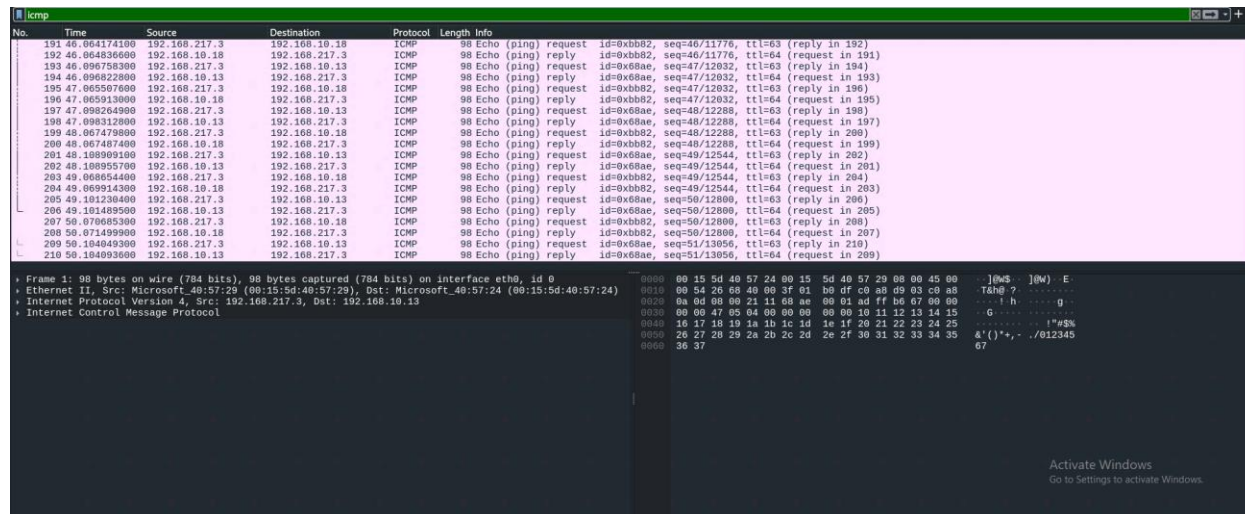
Source – 192.168.217.2:53

Destination - 192.168.217.3:53323

## Task B

### 1. Sniff ICMP traffic

- Apply proper display or capture filter in Wireshark on Internal Kali VM to show active ICMP traffic.



This is the filtered list of all active ICMP traffic.

- Apply a proper display or capture filter on the internal Kali VM that ONLY displays the ICMP request that originated from the external Kali VM and goes to the Ubuntu 64-bit VM.

| icmp.type == 8 && ip.src == 192.168.217.3 && ip.dst == 192.168.10.18 |              |               |               |          |        |                                                                   |
|----------------------------------------------------------------------|--------------|---------------|---------------|----------|--------|-------------------------------------------------------------------|
| No.                                                                  | Time         | Source        | Destination   | Protocol | Length | Info                                                              |
| 242                                                                  | 58.093156500 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=58/14848, ttl=63 (reply in 243) |
| 246                                                                  | 59.089887200 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=59/15104, ttl=63 (reply in 247) |
| 250                                                                  | 60.091553000 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=60/15360, ttl=63 (reply in 251) |
| 254                                                                  | 61.095986800 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=61/15616, ttl=63 (reply in 255) |
| 258                                                                  | 62.099270800 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=62/15872, ttl=63 (reply in 259) |
| 262                                                                  | 63.097260800 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=63/16128, ttl=63 (reply in 263) |
| 266                                                                  | 64.099762400 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=64/16384, ttl=63 (reply in 267) |
| 270                                                                  | 65.101140800 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=65/16640, ttl=63 (reply in 271) |
| 274                                                                  | 66.102083900 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=66/16896, ttl=63 (reply in 275) |
| 278                                                                  | 67.103205300 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=67/17152, ttl=63 (reply in 279) |
| 282                                                                  | 68.106206800 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=68/17408, ttl=63 (reply in 283) |
| 286                                                                  | 69.107846400 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=69/17664, ttl=63 (reply in 287) |
| 290                                                                  | 70.109435600 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=70/17920, ttl=63 (reply in 291) |
| 294                                                                  | 71.110520700 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=71/18176, ttl=63 (reply in 295) |
| 298                                                                  | 72.121927400 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=72/18432, ttl=63 (reply in 299) |
| 302                                                                  | 73.120650800 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=73/18688, ttl=63 (reply in 303) |
| 306                                                                  | 74.122920300 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=74/18944, ttl=63 (reply in 307) |
| 310                                                                  | 75.120999300 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=75/19200, ttl=63 (reply in 311) |
| 314                                                                  | 76.123584800 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=76/19456, ttl=63 (reply in 315) |
| 318                                                                  | 77.124691200 | 192.168.217.3 | 192.168.10.18 | ICMP     | 98     | Echo (ping) request id=0xb82, seq=77/19712, ttl=63 (reply in 319) |

Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0  
 Ethernet II, Src: Microsoft\_40:57:20 (00:15:5d:40:57:20), Dst: Microsoft\_40:57:32 (00:15:5d:40:57:32)  
 Internet Protocol Version 4, Src: 192.168.217.3, Dst: 192.168.10.18  
 Internet Control Message Protocol

This is the super-filtered list only showing traffic from the attacker to the target.

## 2. Sniff FTP traffic

- Ubuntu VM is also serving as an FTP server inside the LAN network. Now, you need to use External Kali to access this FTP server by using the command: ftp [ip\_addr of ubuntu VM]. The username for the FTP server is student, and the password is password. You can follow the steps below to access the FTP server.

```

root@kali: ~
File Actions Edit View Help

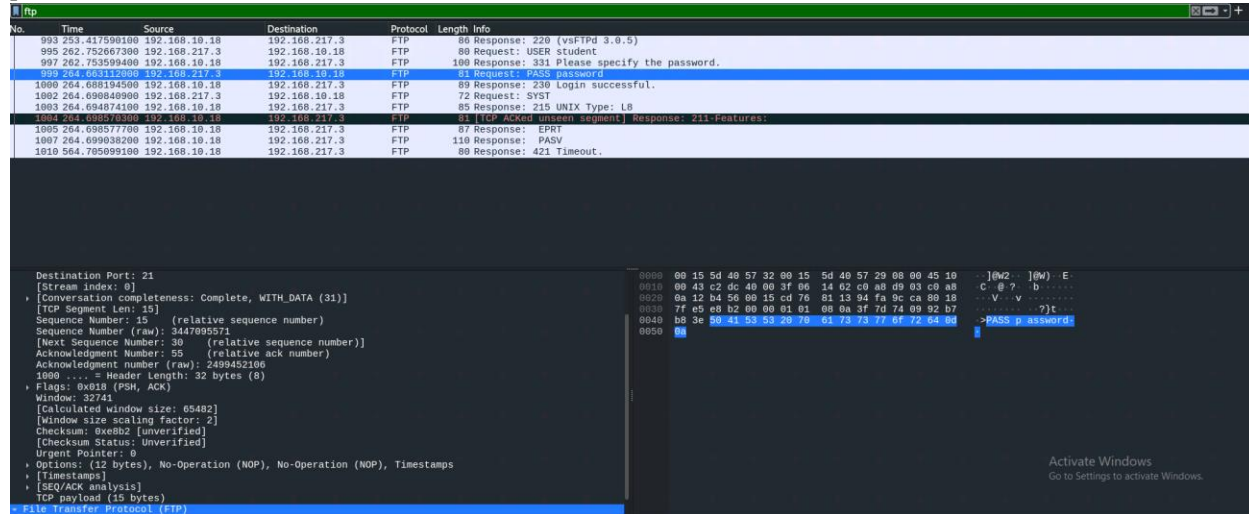
64 bytes from 192.168.10.18: icmp_seq=227 ttl=63 time=8.68 ms
64 bytes from 192.168.10.18: icmp_seq=228 ttl=63 time=2.60 ms
64 bytes from 192.168.10.18: icmp_seq=229 ttl=63 time=3.76 ms
64 bytes from 192.168.10.18: icmp_seq=230 ttl=63 time=3.31 ms
64 bytes from 192.168.10.18: icmp_seq=231 ttl=63 time=11.3 ms
64 bytes from 192.168.10.18: icmp_seq=232 ttl=63 time=16.7 ms
64 bytes from 192.168.10.18: icmp_seq=233 ttl=63 time=2.27 ms
64 bytes from 192.168.10.18: icmp_seq=234 ttl=63 time=2.27 ms
64 bytes from 192.168.10.18: icmp_seq=235 ttl=63 time=11.1 ms
64 bytes from 192.168.10.18: icmp_seq=236 ttl=63 time=3.09 ms
64 bytes from 192.168.10.18: icmp_seq=237 ttl=63 time=20.7 ms
^C
— 192.168.10.18 ping statistics —
237 packets transmitted, 237 received, 0% packet loss, time 236457ms
rtt min/avg/max/mdev = 1.783/6.423/27.033/5.397 ms

(root@kali)-[~]
# ftp 192.168.10.18
Connected to 192.168.10.18.
220 (vsFTPd 3.0.5)
Name (192.168.10.18:root): student
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

```

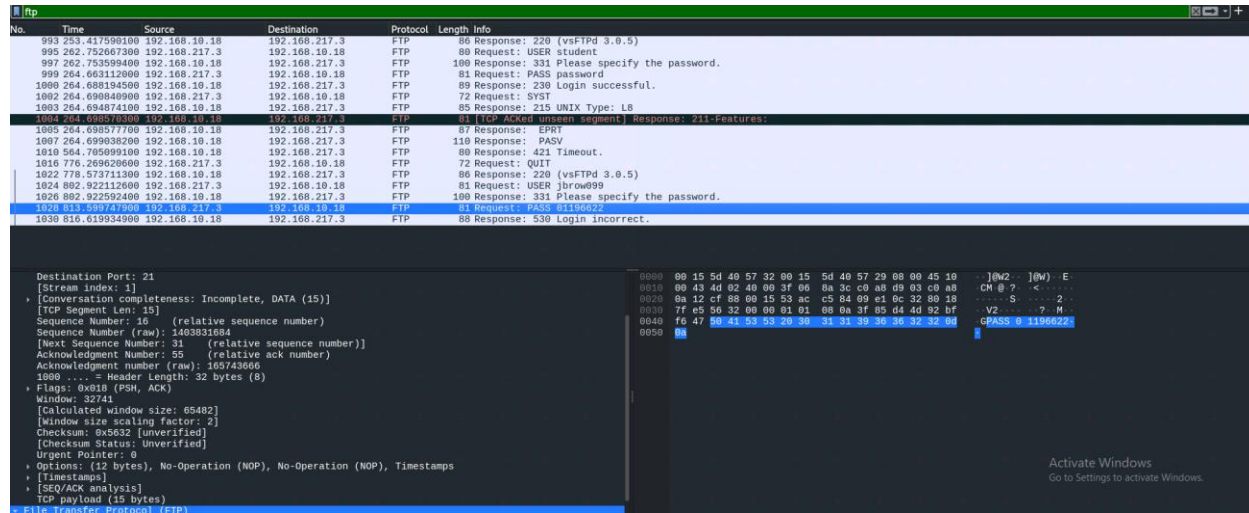
I'm In.

- b. Unfortunately, Internal Kali, the attacker, is also sniffing into the communication. Therefore, all of your communication is exposed to the attacker. Now, you need to find out the password used by External Kali to access the FTP server from the intercepted traffic on Internal Kali. You need to take a screenshot and explain how you found the password.



This screenshot shows how I found the password used by the attacker. I first filtered for any FTP traffic, and then looked for the contents of the “info” section to find where the unbuntu VM requested the password. The next communication would be the password.

- c. After you successfully find the username & password from the FTP traffic, repeat the previous step (2.a), and use your MIDAS ID as the username and UIN as the password to re-access the FTP server from External Kali. Although External Kali may not access the FTP server, you need to intercept the packets containing these “secrets” from the attacker VM, which is Internal Kali.



I used the same filtering method, only my MIDAS and UIN were the username and password respectively.