

Mikias Berhanu

2021280115

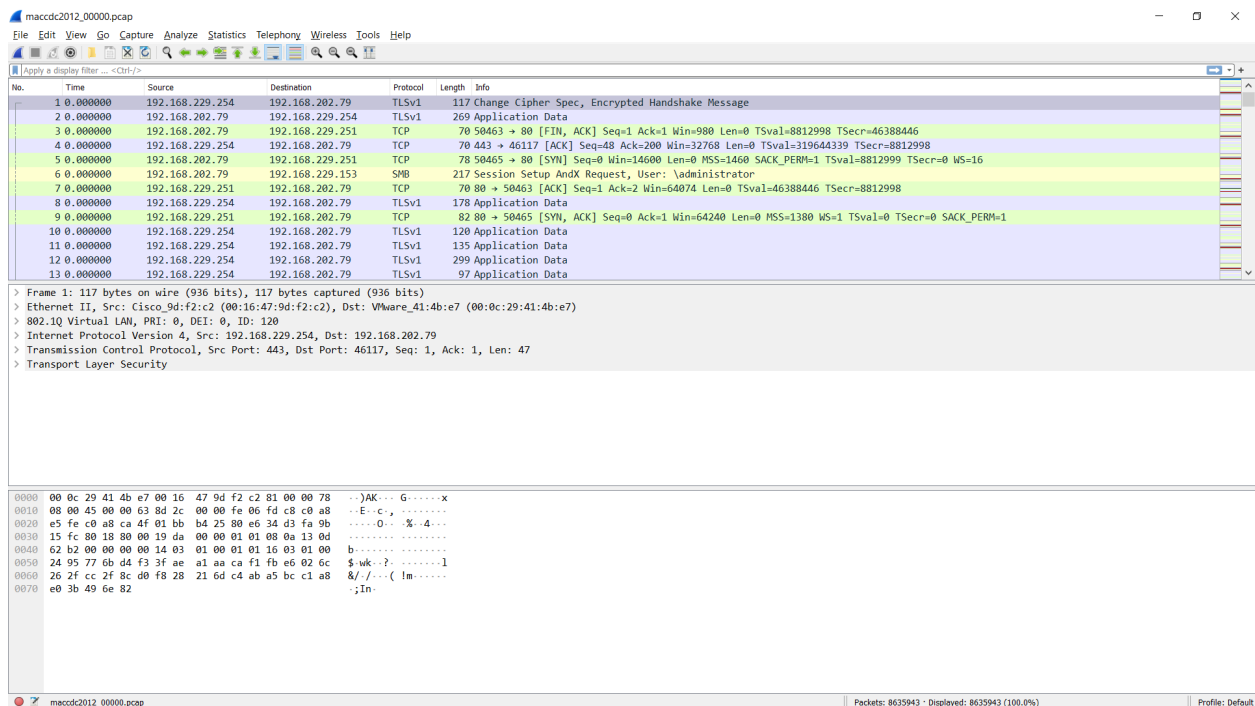
Assignment Submission VI

Analysing Network Traffic Using Wireshark and Python

Wireshark is a free and open source tool which is used for network troubleshooting testing and network traffic analysis. The first name was Ethereal which was later renamed to wireshark.

For this assignment I have used pyshark, a python module which allows us to parse packets using wireshark dissectors . [github](#)

I download a Pcap file from this website for this assignment [here](#) and I have also linked the work for this assignment [here](#)



Some simple filter techniques on wireshark

- 1) If we want to filter out data using ip address source we can use `ip.src== <ip address>`

ip.v4-smtp.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 192.168.20.70

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.20.70	74.125.131.27	TCP	74	54557 → 25 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=177816630 TSecr=0 WS=128
3	0.029301	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=177816637 TSecr=61180008
5	0.189851	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=1 Ack=52 Win=29312 Len=0 TSval=177816677 TSecr=61180169
6	3.781970	192.168.20.70	74.125.131.27	SMTP	72	C: ehlo
9	3.814535	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=7 Ack=219 Win=30336 Len=0 TSval=177817583 TSecr=61183793
10	8.699111	192.168.20.70	74.125.131.27	SMTP	72	C: quit
12	8.729514	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=13 Ack=275 Win=30336 Len=0 TSval=177818812 TSecr=61188708
14	8.729783	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [FIN, ACK] Seq=13 Ack=276 Win=30336 Len=0 TSval=177818812 TSecr=61188708

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

> Ethernet II, Src: VMware_b3:a:a0 (00:50:56:bb:3a:a0), Dst: HewlettP_5e:4d:26 (00:1f:29:5e:4d:26)

> Internet Protocol Version 4, Src: 192.168.20.70, Dst: 74.125.131.27

> Transmission Control Protocol, Src Port: 54557, Dst Port: 25, Seq: 0, Len: 0

```

0000  00 1f 29 5e 4d 26 00 50 56 bb 3a a0 08 00 45 10  --)M&P V:...E:
0010  00 3c 83 1b 40 00 40 06 15 0a c0 a8 14 46 4a 7d  -<.@.@. ....FJ]
0020  83 1b d5 1d 00 19 0b 7f c7 2d 00 00 00 00 a0 02  ....K. ....
0030  72 10 a2 b5 00 00 02 04 05 b4 04 02 08 0a 0a 99  P.....
0040  44 36 00 00 00 01 03 07 06 00 00 00 00 00 00 00  D6.....
  
```

Packets: 15 · Displayed: 8 (53.3%)

Profile: Default

2) If we want to filter out data using ip subnet address we can use *ip.addr=<ip address/subnet>*

ip4-smtp.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 192.168.20.1/24

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.20.70	74.125.131.27	TCP	74	54557 → 25 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=177816630 TSecr=0 WS=128
2	0.0029195	74.125.131.27	192.168.20.70	TCP	74	25 → 54557 [SYN, ACK] Seq=0 Ack=1 Win=42540 Len=0 MSS=1460 SACK_PERM=1 TSval=611800008 TSecr=177816630 WS=128
3	0.029301	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=177816637 TSecr=611800008
4	0.189770	74.125.131.27	192.168.20.70	SMTP	117	S: 220 mx.google.com ESMTP q8si1038396vcq.58 - gsmt
5	0.189851	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=1 Ack=52 Win=29312 Len=0 TSval=177816677 TSecr=61180169
6	3.781970	192.168.20.70	74.125.131.27	SMTP	72	C: ehlo
7	3.811138	74.125.131.27	192.168.20.70	TCP	66	25 → 54557 [ACK] Seq=52 Ack=7 Win=42624 Len=0 TSval=61183790 TSecr=177817575
8	3.814478	74.125.131.27	192.168.20.70	SMTP	233	S: 250-mx.google.com at your service, [108.39.81.51] SIZE 35882577 8BITIME STARTTLS ENHANCEDSTATUSCODES PIPELINING ...
9	3.814535	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=7 Ack=219 Win=30336 Len=0 TSval=177817583 TSecr=61183793
10	8.699111	192.168.20.70	74.125.131.27	SMTP	72	C: quit
11	8.729437	74.125.131.27	192.168.20.70	SMTP	122	S: 221 2.0.0 closing connection q8si1038396vcq.58 - gsmt
12	8.729514	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=13 Ack=275 Win=30336 Len=0 TSval=177818012 TSecr=61180708
13	8.729537	74.125.131.27	192.168.20.70	TCP	66	25 → 54557 [FIN, ACK] Seq=275 Ack=13 Win=42624 Len=0 TSval=61180708 TSecr=177818005

> Frame 11: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
> Ethernet II, Src: HewlettP_Se:4d:26 (00:1f:29:5e:4d:26), Dst: VMware_bb:3a:a0 (00:50:56:bb:3a:a0)
> Internet Protocol Version 4, Src: 74.125.131.27, Dst: 192.168.20.70
> Transmission Control Protocol, Src Port: 25, Dst Port: 54557, Seq: 219, Ack: 13, Len: 56
> Simple Mail Transfer Protocol

0000 00 50 56 bb 3a a0 00 1f 29 5e 4d 26 08 00 45 00 -PV:....)MR:E
0010 00 6c 51 5d 00 00 31 06 95 a8 4a 7d 83 1b c0 a8 -[Q]-1-...J)....
0020 14 46 00 19 d5 1d 99 56 ae 5b 6b 7f c7 3a 08 18 -F:....V [k:....
0030 01 4d 13 5f 00 00 01 01 08 0a 03 a5 aa 64 0a 99 -M:.....d-..
0040 4c b5 32 32 31 20 32 2e 30 2e 30 20 63 6c 6f 73 L 221 2. 0.0 clos
0050 69 6e 67 20 63 6f 6e 65 63 74 69 6e 20 71 ing connection q
0060 38 73 69 31 30 33 38 33 39 36 76 63 71 2e 35 38 8si10383 96vcq.58
0070 20 2d 20 67 73 6d 74 70 0d 0a - gsmt -

ip4-smtp.cap Packets: 15 · Displayed: 15 (100.0%) Profile: Default

3) Filter out using port numbers `tcp.port == <port number>`

ip4-smtp.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 25

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.20.70	74.125.131.27	TCP	74	54557 → 25 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=177816630 TSecr=0 WS=128
2	0.0029195	74.125.131.27	192.168.20.70	TCP	74	25 → 54557 [SYN, ACK] Seq=0 Ack=1 Win=42540 Len=0 MSS=1460 SACK_PERM=1 TSval=611800008 TSecr=177816630 WS=128
3	0.029301	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=177816637 TSecr=611800008
4	0.189770	74.125.131.27	192.168.20.70	SMTP	117	S: 220 mx.google.com ESMTP q8si1038396vcq.58 - gsmt
5	0.189851	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=1 Ack=52 Win=29312 Len=0 TSval=177816677 TSecr=61180169
6	3.781970	192.168.20.70	74.125.131.27	SMTP	72	C: ehlo
7	3.811138	74.125.131.27	192.168.20.70	TCP	66	25 → 54557 [ACK] Seq=52 Ack=7 Win=42624 Len=0 TSval=61183790 TSecr=177817575
8	3.814478	74.125.131.27	192.168.20.70	SMTP	233	S: 250-mx.google.com at your service, [108.39.81.51] SIZE 35882577 8BITIME STARTTLS ENHANCEDSTATUSCODES PIPELINING ...
9	3.814535	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=7 Ack=219 Win=30336 Len=0 TSval=177817583 TSecr=61183793
10	8.699111	192.168.20.70	74.125.131.27	SMTP	72	C: quit
11	8.729437	74.125.131.27	192.168.20.70	SMTP	122	S: 221 2.0.0 closing connection q8si1038396vcq.58 - gsmt
12	8.729514	192.168.20.70	74.125.131.27	TCP	66	54557 → 25 [ACK] Seq=13 Ack=275 Win=30336 Len=0 TSval=177818012 TSecr=61180708
13	8.729537	74.125.131.27	192.168.20.70	TCP	66	25 → 54557 [FIN, ACK] Seq=275 Ack=13 Win=42624 Len=0 TSval=61180708 TSecr=177818005

> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> Ethernet II, Src: VMware_bb:3a:a0 (00:50:56:bb:3a:a0), Dst: HewlettP_Se:4d:26 (00:1f:29:5e:4d:26)
> Internet Protocol Version 4, Src: 192.168.20.70, Dst: 74.125.131.27
> Transmission Control Protocol, Src Port: 54557, Dst Port: 25, Seq: 0, Len: 0

0000 00 1f 29 5e 4d 26 00 50 56 bb 3a a0 08 00 45 10 -)MR.P V:....E
0010 00 3c 83 1b 40 00 00 06 15 0a c0 a8 14 46 4a 7d -<-@.@:....FJ
0020 83 1b d5 1d 00 19 6b 7f c7 2d 00 00 00 00 a0 02 -....k:.....
0030 72 10 a2 b5 00 00 02 04 05 b4 04 02 08 0a 0a 99 m:.....
0040 44 36 00 00 00 01 03 03 07 D6:.....

ip4-smtp.cap Packets: 15 · Displayed: 15 (100.0%) Profile: Default

4) Get SMTP records just search using keyword `SMTP`

No.	Time	Source	Destination	Protocol	Length	Info
4	0.189770	74.125.131.27	192.168.20.70	SMTP	117	S: 220 mx.google.com ESMTP q8si1038396vcq.58 - gsmt
6	3.781970	192.168.20.70	74.125.131.27	SMTP	72	C: ehlo
8	3.814478	74.125.131.27	192.168.20.70	SMTP	233	S: 250-mx.google.com at your service, [108.39.81.51] SIZE 35882577 8BITMIME STARTTLS ENHANCEDSTATUSCODES PIPELINING CHUN...
10	8.699111	192.168.20.70	74.125.131.27	SMTP	72	C: quit
11	8.729437	74.125.131.27	192.168.20.70	SMTP	122	S: 221 2.0.0 closing connection q8si1038396vcq.58 - gsmt

> Frame 11: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)
 > Ethernet II, Src: HewlettP_5e:4d:26 (00:1f:29:5e:4d:26), Dst: VMware_bb:3a:a0 (00:50:56:bb:3a:a0)
 > Internet Protocol Version 4, Src: 74.125.131.27, Dst: 192.168.20.70
 > Transmission Control Protocol, Src Port: 25, Dst Port: 54557, Seq: 219, Ack: 13, Len: 56
 > Simple Mail Transfer Protocol

0000 00 50 56 bb 3a a0 00 1f 29 5e 4d 26 08 00 45 00 -PV:....YMR:E
 0010 00 6c 51 5d 00 00 31 06 95 a8 4a 7d 83 1b c0 a8 -lQ]-1- -J}....
 0020 14 46 00 19 d5 1d 99 56 ae 5b 6b 7f c7 3a 00 18 -F:---V-[k:---
 0030 01 4d 13 5f 00 00 01 01 08 0a 03 a5 aa 64 0a 99 -M:-----d-..
 0040 4c b5 32 32 31 20 32 2e 30 2e 30 20 63 6c 6f 73 L-221 2. 0.0 clos
 0050 69 6e 67 20 63 6f 6e 6e 65 63 74 69 6f 6e 20 71 ing conn action q
 0060 38 73 69 31 30 33 38 33 39 36 76 63 71 2e 35 38 8si10383 96vcq.58
 0070 20 2d 20 67 73 6d 74 70 0d 0a - gsmt -

Using pyshark for packet analysis

Note: Most of the values returned by pyshark are string values

```
import pyshark

pcap_file = pyshark.FileCapture("nf9-juniper-vmx.pcapng.cap")

# get a single packet
packet = pcap_file[0]

# get packet source ip
print(packet['ip'].src)
# >> 192.168.17.114
print(packet['ip'].dst)
# >> 192.168.16.36

# get packet layers
```

```

print(packet.layers)
# >> [ < ETH Layer > , < IP Layer > , < UDP Layer > , < DATA Layer > ]

# get Ip field names
print(packet.ip.field_names)
# >> ['version', 'hdr_len', 'dsfield', 'dsfield_dscp', 'dsfield_ecn',
'len', 'id', 'flags', 'flags_rb', 'flags_df',
#      'flags_mf', 'frag_offset', 'ttl', 'proto', 'checksum',
'checksum_status', 'src', 'addr', 'src_host', 'host', 'dst', 'dst_host']

# get Ip version
print(packet.ip.version)
# > 4

# print out the packet content
print(packet.pretty_print())
"""
Layer ETH:
    Destination: 0a:7c:7c:ec:d6:97
    Address: 0a:7c:7c:ec:d6:97
        .... ..1. .... = LG bit: Locally administered
address (this is NOT the factory default)
        .... ...0 .... = IG bit: Individual address
(unicast)
    Source: 00:05:86:71:82:00
        .... ..0. .... = LG bit: Globally unique address
(factory default)
        .... ...0 .... = IG bit: Individual address
(unicast)
    Type: IPv4 (0x0800)
    Address: 00:05:86:71:82:00
Layer IP:
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable
Transport (0)
    Total Length: 84
    Identification: 0x0003 (3)

```

```
Flags: 0x00
0... .... = Reserved bit: Not set
.0... .... = Don't fragment: Not set
..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 250
Protocol: UDP (17)
Header Checksum: 0x1daf [validation disabled]
Header checksum status: Unverified
Source Address: 192.168.17.114
Destination Address: 192.168.16.36
```

Layer UDP:

```
Source Port: 47043
Destination Port: 9995
Length: 16384 (bogus, payload length 64)
Expert Info (Error/Malformed): Bad length value 16384 > IP payload
```

length

```
Bad length value 16384 > IP payload length
Severity level: Error
Group: Malformed
Checksum: 0x0000 [zero-value ignored]
Checksum Status: Not present
Stream index: 0
Timestamps
Time since first frame: 0.000000000 seconds
Time since previous frame: 0.000000000 seconds
UDP payload (56 bytes)
```

DATANone

"""

```
# get host name
print(packet.http.host)
```
