

## **Packet Sniffer Report**

### **Brian Tokumoto**

#### **Intro:**

- This report presents the validation of the pktsniffer.py program, a network packet analyzer that reads packets from a .pcap file and filters them based on specific criteria. This report seeks to compare the output of pktsniffer.py with the corresponding results from Wireshark, ensuring the program correctly extracts and displays network packet

#### **Capture Packets No Filters:**

For common-line arguments: `python pktsniffer.py -r network_traffic.pcap -c 10`

Which limits the number of packets analyzed to 10. What is returned from pktsniffer.py is:

```
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\NtworksHW1> python pktsniffer.py -r network_traffic.pcap -c 10
>>
```

Analyzing 10 packets from network\_traffic.pcap...

Packet Captured:

Ethernet: fe:00:1d:52:57:d7 -> 01:00:5e:00:00:fb | Type: 0x800  
IP: 192.168.1.111 -> 224.0.0.251 | TTL: 1 | Protocol: 2

Packet Captured:

Ethernet: fe:00:1d:52:57:d7 -> 01:00:5e:00:00:fb | Type: 0x800  
IP: 192.168.1.111 -> 224.0.0.251 | TTL: 255 | Protocol: 17  
UDP: 5353 -> 5353

Packet Captured:

Ethernet: fe:00:1d:52:57:d7 -> 33:33:00:00:00:fb | Type: 0x86dd  
UDP: 5353 -> 5353

Packet Captured:

Ethernet: 80:69:1a:07:72:da -> 3c:f0:11:05:38:ae | Type: 0x800  
IP: 140.82.114.25 -> 192.168.1.15 | TTL: 45 | Protocol: 6  
TCP: 443 -> 53789 | Flags: PA

Packet Captured:

Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800  
IP: 192.168.1.15 -> 140.82.114.25 | TTL: 128 | Protocol: 6  
TCP: 53789 -> 443 | Flags: PA

Packet Captured:

Ethernet: 80:69:1a:07:72:da -> 3c:f0:11:05:38:ae | Type: 0x800  
IP: 140.82.114.25 -> 192.168.1.15 | TTL: 45 | Protocol: 6  
TCP: 443 -> 53789 | Flags: A

Packet Captured:

Ethernet: 42:6e:09:07:b5:04 -> 33:33:00:00:00:fb | Type: 0x86dd  
UDP: 5353 -> 5353

Packet Captured:

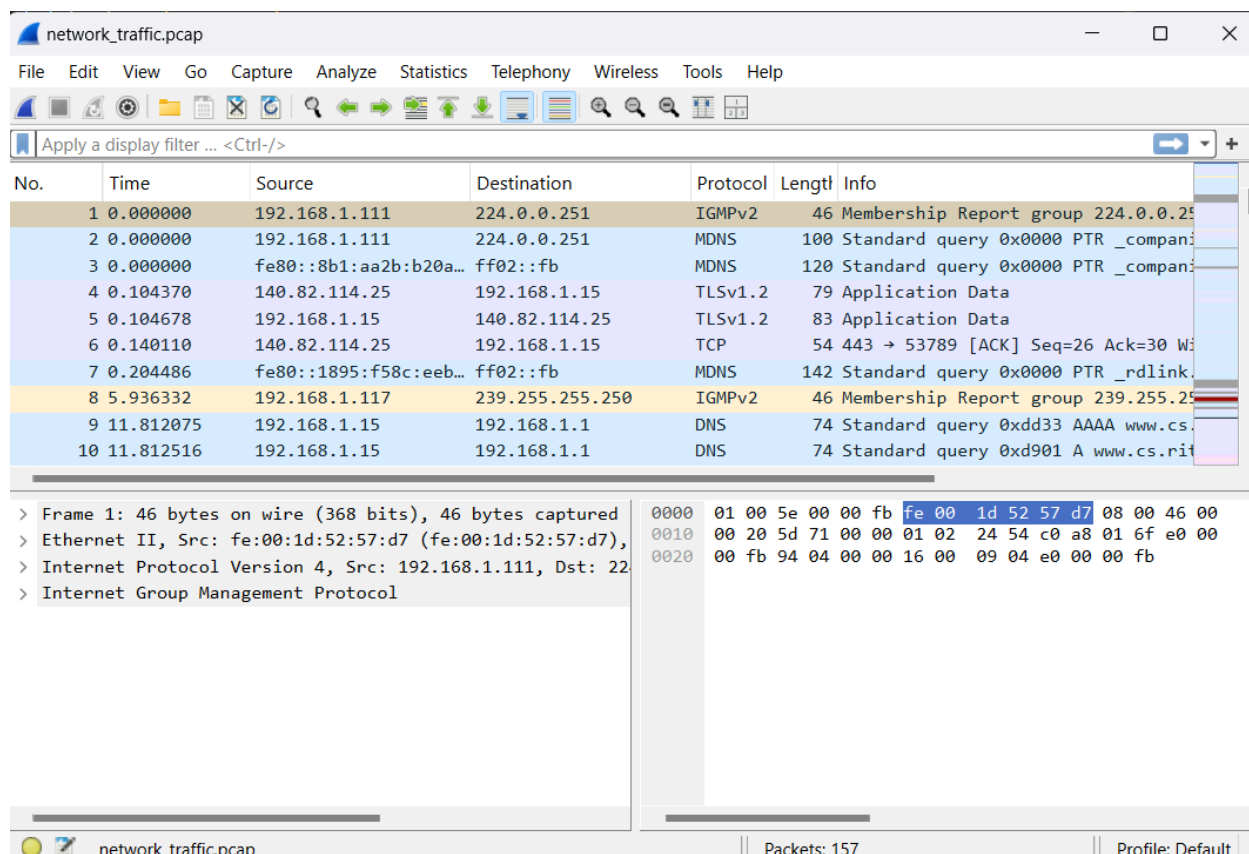
Ethernet: a4:08:01:26:e9:5b -> 01:00:5e:7f:ff:fa | Type: 0x800  
IP: 192.168.1.117 -> 239.255.255.250 | TTL: 1 | Protocol: 2

Packet Captured:

Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800  
IP: 192.168.1.15 -> 192.168.1.1 | TTL: 128 | Protocol: 17  
UDP: 61549 -> 53

Packet Captured:

Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800  
IP: 192.168.1.15 -> 192.168.1.1 | TTL: 128 | Protocol: 17  
UDP: 50863 -> 53  
PS C:\Users\hiron\OneDrive\Desktop\UNI\_HW\NtworksHW1> □



- The pktsniffer.py output correctly displays the first 10 packets.
- Screenshots from Wireshark and pktsniffer.py match when it comes to the first 10 packets.

## Filtering Commands:

### Filter by host (ip address):

```
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\NetworksHW1> python pktsniffer.py -r network_traffic.pcap --host 192.168.1.1 -c 5
```

Analyzing 5 packets from network\_traffic.pcap...

Packet Captured:

Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800  
IP: 192.168.1.15 -> 192.168.1.1 | TTL: 128 | Protocol: 17  
UDP: 61549 -> 53

Packet Captured:

Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800  
IP: 192.168.1.15 -> 192.168.1.1 | TTL: 128 | Protocol: 17  
UDP: 50863 -> 53

Packet Captured:

Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800  
IP: 192.168.1.15 -> 192.168.1.1 | TTL: 128 | Protocol: 17  
UDP: 55314 -> 53

Packet Captured:

Ethernet: 80:69:1a:07:72:da -> 3c:f0:11:05:38:ae | Type: 0x800  
IP: 192.168.1.1 -> 192.168.1.15 | TTL: 64 | Protocol: 17  
UDP: 53 -> 50863

Packet Captured:

Ethernet: 80:69:1a:07:72:da -> 3c:f0:11:05:38:ae | Type: 0x800  
IP: 192.168.1.1 -> 192.168.1.15 | TTL: 64 | Protocol: 17  
UDP: 53 -> 55314

```
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\NetworksHW1>
```

No.	Time	Source	Destination	Protocol	Length	Info
16	11.949146	192.168.1.1	192.168.1.15	DNS	140	Standard query response 0xdd33 AAAA
15	11.929427	2605:9480:10c:22e0:f...	2605:9480:10c:22e0:8...	DNS	94	Standard query 0x7d47 HTTPS www.cs.r
14	11.928816	192.168.1.15	192.168.1.1	DNS	74	Standard query 0x9500 AAAA www.cs.ri
13	11.928317	192.168.1.1	192.168.1.15	DNS	140	Standard query response 0xe397 HTTPS
12	11.882529	192.168.1.1	192.168.1.15	DNS	111	Standard query response 0xd901 A www
11	11.812784	192.168.1.15	192.168.1.1	DNS	74	Standard query 0xe397 HTTPS www.cs.r
10	11.812516	192.168.1.15	192.168.1.1	DNS	74	Standard query 0xd901 A www.cs.rit.e

> Frame 10: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)	0000	80 69 1a 07 72 da 3c f0 11 05 3
> Ethernet II, Src: Intel_05:38:ae (3c:f0:11:05:38:ae), Dst: BelkinIntern_07:72:da (80:6	0010	00 3c a2 b1 00 00 80 11 00 00 c
> Internet Protocol Version 4, Src: 192.168.1.15, Dst: 192.168.1.1	0020	01 01 c6 af 00 35 00 28 83 9a d
> User Datagram Protocol, Src Port: 50863, Dst Port: 53	0030	00 00 00 00 00 00 03 77 77 77 0
Source Port: 50863	0040	74 03 65 64 75 00 00 01 00 01
Destination Port: 53		
Length: 40		
Checksum: 0x839a [unverified]		
[Checksum Status: Unverified]		
[Stream index: 4]		
[Stream Packet Number: 1]		
> [Timestamps]		
UDP payload (32 bytes)		
> Domain Name System (query)		

- The pktsniffer.py output correctly displays packets where 192.168.1.1 is either the source or destination.
- Screenshots from Wireshark and pktsniffer.py match in terms of IP addresses, protocol, and UDP port 53.

**Filter by Port:**

```
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\NetworksHW1> python pktsniffer.py -r network_traffic.pcap --port 61345
Analyzing 2 packets from network_traffic.pcap...
```

```
Packet Captured:
Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x86dd
UDP: 61345 -> 53
```

```
Packet Captured:
Ethernet: 80:69:1a:07:72:da -> 3c:f0:11:05:38:ae | Type: 0x86dd
UDP: 53 -> 61345
```

No.	Time	Source	Destination	Protocol	Length	Info
156	17.101906	fe80::8269:1aff:fe07...	2605:9480:10c:22e0:f...	ICMPv6	86	Neighbor Solicitation for 2605:9480:10c:22e0:f...
154	16.514414	2605:9480:10c:22e0:8...	2605:9480:10c:22e0:f...	ICMPv6	78	Neighbor Advertisement 2605:9480:10c:22e0:8...
153	16.505662	2605:9480:10c:22e0:f...	2605:9480:10c:22e0:8...	ICMPv6	86	Neighbor Solicitation for 2605:9480:10c:22e0:f...
132	15.822165	2605:9480:10c:22e0:8...	2605:9480:10c:22e0:f...	DNS	189	Standard query response 0x9a3e No su
131	15.822165	2605:9480:10c:22e0:8...	2605:9480:10c:22e0:f...	DNS	189	Standard query response 0x38c3 No su
127	15.770540	2605:9480:10c:22e0:f...	2605:9480:10c:22e0:8...	DNS	107	Standard query 0x9a3e AAAA wpad.greenl
126	15.770217	2605:9480:10c:22e0:f...	2605:9480:10c:22e0:8...	DNS	107	Standard query 0x38c3 A wpad.greenli
80	12.268228	192.168.1.1	192.168.1.15	DNS	176	Standard query response 0x1caa HTTPS
79	12.268228	192.168.1.1	192.168.1.15	DNS	102	Standard query response 0x2d07 A cdn
78	12.268228	192.168.1.1	192.168.1.15	DNS	176	Standard query response 0x5383 AAAA

> Frame 131: 189 bytes on wire (1512 bits), 189 bytes captured on interface 0

> Ethernet II, Src: BelkinIntern\_07:72:da (80:69:1a:07:72:da), Dst: 3c:f0:11:05:38:ae

> Internet Protocol Version 6, Src: 2605:9480:10c:22e0:8269:1aff:fe07:1105, Dst: fe80::8269:1aff:fe07:1105

> User Datagram Protocol, Src Port: 53, Dst Port: 61345

Source Port: 53

Destination Port: 61345

Length: 135

Checksum: 0x47e6 [unverified]

[Checksum Status: Unverified]

[Stream index: 17]

[Stream Packet Number: 2]

```

0000  3c f0 11 05 38 ae 80 69 1a 07 72 da 86 dd 60 09  <...8...
0010  7e 96 00 87 11 40 26 05 94 80 01 0c 22 e0 82 69  ~....@
0020  1a ff fe 07 72 da 26 05 94 80 01 0c 22 e0 fc b9  ....r.
0030  ae 50 ac de b4 0d 00 35 ef a1 00 87 47 e6 38 c3  .P....
0040  81 83 00 01 00 00 00 01 00 00 04 77 70 61 64 12  ....
0050  67 72 65 65 6e 6c 69 67 68 74 6e 65 74 77 6f 72  greenl
0060  6b 73 03 63 6f 6d 00 00 01 00 01 c0 11 00 06 00  ks.com
0070  01 00 00 03 01 00 46 07 6e 73 2d 31 31 33 39 09  ....
0080  61 77 73 64 6e 73 2d 31 34 03 6f 72 67 00 11 61  awsdns
0090  77 73 64 6e 73 2d 68 6f 73 74 6d 61 73 74 65 72  wsdns-
00a0  06 61 6d 61 7a 6f 6e c0 24 00 00 00 01 00 00 1c  -amazo
00b0  20 00 00 03 84 00 12 75 00 00 01 51 80  ....

```

User Datagram Protocol (udp), 8 bytes      Packets: 157      Profile: Default

- The program accurately captures packets where port 61345 is involved.
- Wireshark shows DNS requests and responses using port 61345, matching pktsniffer.py output.

## Filter by icmp:

```
PS C:\Users\hiron\OneDrive\Desktop\UNI_HW\NetworksHW1> python pktsniffer.py -r network_traffic.pcap --icmp
Analyzing 0 packets from network_traffic.pcap...
```

No.	Time	Source	Destination	Protocol	Length	Info
127	15.770540	2605:9480:10c:22e0:f...	2605:9480:10c:22e0:8...	DNS	107	Standard query 0x9a3e AAAA wpad.gree
131	15.822165	2605:9480:10c:22e0:8...	2605:9480:10c:22e0:f...	DNS	189	Standard query response 0x38c3 No su
132	15.822165	2605:9480:10c:22e0:8...	2605:9480:10c:22e0:f...	DNS	189	Standard query response 0x9a3e No su
153	16.505662	2605:9480:10c:22e0:f...	2605:9480:10c:22e0:8...	ICMPv6	86	Neighbor Solicitation for 2605:9480:
154	16.514414	2605:9480:10c:22e0:8...	2605:9480:10c:22e0:f...	ICMPv6	78	Neighbor Advertisement 2605:9480:10c
156	17.101906	fe80::8269:1aff:fe07...	2605:9480:10c:22e0:f...	ICMPv6	86	Neighbor Solicitation for 2605:9480:
157	17.101988	2605:9480:10c:22e0:f...	fe80::8269:1aff:fe07...	ICMPv6	86	Neighbor Advertisement 2605:9480:10c
1	0.000000	192.168.1.111	224.0.0.251	IGMPv2	46	Membership Report group 224.0.0.251
8	5.936332	192.168.1.117	239.255.255.250	IGMPv2	46	Membership Report group 239.255.255.
36	12.082310	192.168.1.1	239.255.255.250	IGMPv2	46	Membership Query, specific for group
155	17.003340	192.168.1.15	239.255.255.250	IGMPv2	46	Membership Report group 239.255.255.
2	0.000000	192.168.1.111	224.0.0.251	MDNS	100	Standard query 0x0000 PTR _companion

- No ICMP packets were found in the .pcap file, Wireshark has ICMPv6 but not ICMP and `` returned no results, confirming correctness.

## Filtered by NET:

```
python pktsniffer.py -r network_traffic.pcap --net 192.168.1 -c 5

Analyzing 5 packets from network_traffic.pcap...

Packet Captured:
Ethernet: fe:00:1d:52:57:d7 -> 01:00:5e:00:00:fb | Type: 0x800
IP: 192.168.1.111 -> 224.0.0.251 | TTL: 1 | Protocol: 2

Packet Captured:
Ethernet: fe:00:1d:52:57:d7 -> 01:00:5e:00:00:fb | Type: 0x800
IP: 192.168.1.111 -> 224.0.0.251 | TTL: 255 | Protocol: 17
UDP: 5353 -> 5353

Packet Captured:
Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800
IP: 192.168.1.15 -> 140.82.114.25 | TTL: 128 | Protocol: 6
TCP: 53789 -> 443 | Flags: PA

Packet Captured:
Ethernet: a4:08:01:26:e9:5b -> 01:00:5e:7f:ff:fa | Type: 0x800
IP: 192.168.1.117 -> 239.255.255.250 | TTL: 1 | Protocol: 2

Packet Captured:
Ethernet: 3c:f0:11:05:38:ae -> 80:69:1a:07:72:da | Type: 0x800
IP: 192.168.1.15 -> 192.168.1.1 | TTL: 128 | Protocol: 17
UDP: 61549 -> 53
```

140	15.900259	192.168.1.15	52.182.143.213	TCP	54	53842 -> 443 [ACK] Seq=2721 Ack=6506
145	16.003826	52.182.143.213	192.168.1.15	TCP	54	443 -> 53842 [ACK] Seq=6506 Ack=4709
148	16.007261	192.168.1.15	52.182.143.213	TCP	54	53842 -> 443 [ACK] Seq=4709 Ack=6671
151	16.084715	192.168.1.15	52.182.143.213	TCP	54	53842 -> 443 [ACK] Seq=4740 Ack=6983
152	16.095338	52.182.143.213	192.168.1.15	TCP	54	443 -> 53842 [ACK] Seq=6983 Ack=4740
4	0.104370	140.82.114.25	192.168.1.15	TLSv1.2	79	Application Data
5	0.104678	192.168.1.15	140.82.114.25	TLSv1.2	83	Application Data
24	12.006424	192.168.1.15	129.21.34.17	TLSv1.3	2161	Client Hello (SNI=www.cs.rit.edu)
25	12.006944	192.168.1.15	129.21.34.17	TLSv1.3	2193	Client Hello (SNI=www.cs.rit.edu)
28	12.042678	129.21.34.17	192.168.1.15	TLSv1.3	314	Server Hello, Change Cipher Spec, Ap
29	12.043472	192.168.1.15	129.21.34.17	TLSv1.3	134	Change Cipher Spec, Application Data
31	12.051249	129.21.34.17	192.168.1.15	TLSv1.3	314	Server Hello, Change Cipher Spec, Ap

- The output correctly shows packets involving the 192.168.1.x subnet.
- Matches observed in Wireshark confirm accurate packet selection

**Conclusion:**

- The pktsniffer.py program successfully replicates packet filtering behavior from Wireshark and all tests demonstrate that the extracted packets match those observed in Wireshark.