

# **COMP 7003**

## **Introduction to Information and Network Security**

### *Assignment-02*

### *Testing*

Anmol Mittal

A01397754

February 2<sup>nd</sup>, 2025

Course Reference Number (CRN): 91662

<b>Tests .....</b>	<b>4</b>
Test 01 .....	4
Test 02 .....	4
Test 03 .....	4
Test 04 .....	5
Test 05 .....	5
Test 06 .....	5
Test 07 .....	6
Test 08 .....	6
Test 09 .....	7
Test 10 .....	7
Test 11 .....	8
Test 12 .....	8
Test 13 .....	9
Test 14 .....	9

Test	Expected	Actual	Screenshot
Enter invalid BPF filter	fail	fail	<a href="#">Test 1</a>
Too many arguments in the BPF Filter	fail	fail	<a href="#">Test 2</a>
Enter no BPF filter	pass	pass	<a href="#">Test 3</a>
Run the program with too many CLI arguments	fail	fail	<a href="#">Test 4</a>
Enter non-number/integer in the number of packets to capture	fail	fail	<a href="#">Test 5</a>
No number in the number of packets to capture prompt	pass	pass	<a href="#">Test 6</a>
Negative number in the number of packets to capture	pass	pass	<a href="#">Test 7</a>
Positive number in the number of packets to capture	pass	pass	<a href="#">Test 8</a>
Valid BPF filter for ARP	pass	pass	<a href="#">Test 9</a>
Valid BPF filter for TCP	pass	pass	<a href="#">Test 10</a>
Valid BPF filter for UDP	pass	pass	<a href="#">Test 11</a>
Valid BPF filter for ICMP	pass	pass	<a href="#">Test 12</a>
No interface entered	pass	pass	<a href="#">Test 13</a>
No arguments entered at all	pass	pass	<a href="#">Test 14</a>

# Tests

## Test 01

```
anmol@anmols-x1: ~/Documents/BCIT/comp-7003-a2/source
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$ sudo python3 main.py -f wrongfilter
Error: Invalid filter 'wrongfilter'. Allowed filters: tcp, icmp, udp, arp.
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$
```

## Test 02

```
anmol@anmols-x1: ~/Documents/BCIT/comp-7003-a2/source
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$ sudo python3 main.py -f tcp icmp
usage: main.py [-h] [-i INTERFACE] [-f FILTER] [-c COUNT]
main.py: error: unrecognized arguments: icmp
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$
```

## Test 03

```
anmol@anmols-x1: ~/Documents/BCIT/comp-7003-a2/source
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$ sudo python3 main.py -i any -c 1
No filter provided. Please provide a filter (tcp, icmp, arp, udp) or press Enter to capture all packets: udp
Available interfaces: ['lo', 'enp0s31f6', 'wlp2s0', 'enx84ba59172434']
Starting packet capture on wlp2s0
Starting packet capture on wlp2s0 with filter: udp

Captured Packet 1:
=====
01005e0000fb6a0265aaad89080045000710f7a000ff11c0d30a0003e0000fb14e914e9005d29f3000000
000004000000000000f5f636fd70616e696f6e2d6c696e6b045f746370056c6f63616c0000c001075f7264
6c696e6bc01c000c0001055f72616f70c01c000c0001085f616972706c6179c01c000c0001
=====
Parsing Ethernet Header
-----
Ethernet Header:
Destination MAC: 01005e0000fb | 01:00:5e:00:00:fb
Source MAC: 6a0265aaad89 | 6a:02:65:aa:ad:89
EtherType: 0800 | 2048

Parsing IPv4 Header
-----
IPv4 Header:
Version: 4 | 4
Header Length: 5 | 5
Total Length: 0071 | 113
Identification: 0f7a | 3962
Flags & Frag Offset: 0000 | 0b0000000000000000
Reserved Bit: 0
DF (Do not Fragment): 0
MF (More Fragments): 0
Fragment Offset: 0x0 | 0
Protocol: 11 | 17
Source IP: 8a000033 | 10.0.0.51
Destination IP: e0000fb | 224.0.0.251

Parsing UDP Header
-----
UDP Header:
Source Port: 14e9 | 5353
Destination Port: 14e9 | 5353
Length: 005d | 93
Checksum: 29f3 | 10739

-----UDP Payload-----
000000000004000000000000f5f636fd70616e696f6e2d6c696e6b045f746370056c6f63616c0000c00107
5f72646c696e6bc01c000c0001055f72616f70c01c000c0001085f616972706c6179c01c000c0001
=====
Packet capture completed on wlp2s0.
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$
```

Wireshark interface showing packet capture details for UDP. The packet list shows a single packet from 10.0.0.51 to 224.0.0.251. The packet details pane shows the Ethernet II, Internet Protocol Version 4, and User Datagram Protocol headers. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination
30	0.754189472	10.0.0.51	224.0.0.251
31	0.754189626	fe80::14b0:a203:8e70:279d	ff02::fb
32	0.754189654	10.0.0.51	224.0.0.251
35	7.264825491	10.0.0.51	224.0.0.251
36	7.264825597	fe80::14b0:a203:8e70:279d	ff02::fb

## Test 04

```
anmol@anmols-x1: ~/Documents/BCIT/comp-7003-a2/source

anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$ sudo python3 main.py -i any -c 1 -f tcp -r truth
usage: main.py [-h] [-i INTERFACE] [-f FILTER] [-c COUNT]
main.py: error: unrecognized arguments: -r truth
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$
```

## Test 05

```
anmol@anmols-x1: ~/Documents/BCIT/comp-7003-a2/source

anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$ sudo python3 main.py -i any -c a -f tcp
usage: main.py [-h] [-i INTERFACE] [-f FILTER] [-c COUNT]
main.py: error: argument -c/--count: invalid int value: 'a'
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$
```

## Test 06

```
anmol@anmols-x1: ~/Documents/BCIT/comp-7003-a2/source

(f default: any)
-f FILTER, --filter FILTER
    BPF filter to apply (e.g., 'tcp, udp, arp, icmp'). If not
    provided, captures all packets.
-c COUNT, --count COUNT
    Number of packets to capture (default: 1)
Available interfaces: ['lo', 'enp0s3if6', 'wlp2s0', 'enx84ba59172434']
Starting packet capture on wlp2s0
Starting packet capture on wlp2s0 with filter: icmp

Captured Packet 1:
=====
c4509c85928f3c6aa73525aa080045000054aa5400040016f6c0a000047345ee1f208009d03230b0017b0b9f
67000000005baa030000000000101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f
3031323334353637
=====
Parsing Ethernet Header
-----
Ethernet Header:
Destination MAC: c4509c85928f | c4:50:9c:85:92:8f
Source MAC: 3c6aa73525aa | 3c:6a:a7:35:25:aa
EtherType: 0800 | 2048

Parsing IPv4 Header
-----
IPv4 Header:
Version: 4 | 4
Header Length: 5 | 5
Total Length: 0054 | 84
Identification: aaa5 | 43685
Flags & Frag Offset: 4000 | 0b0100000000000000
Reserved Bit: 0
DF (Do not Fragment): 1
MF (More Fragments): 0
Fragment Offset: 0x0 | 0
Protocol: 01 | 1
Source IP: 8a000047 | 10.0.0.71
Destination IP: 345ee1f2 | 52.94.225.242

Parsing ICMP Header
-----
ICMP Header:
Type: 08 | 8
Code: 00 | 0
Checksum: 9d03 | 40195

-----ICMP Payload-----
101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f3031323334353637
=====
Packet capture completed on wlp2s0.
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$
```

Sat Feb 1 10:07 PM

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

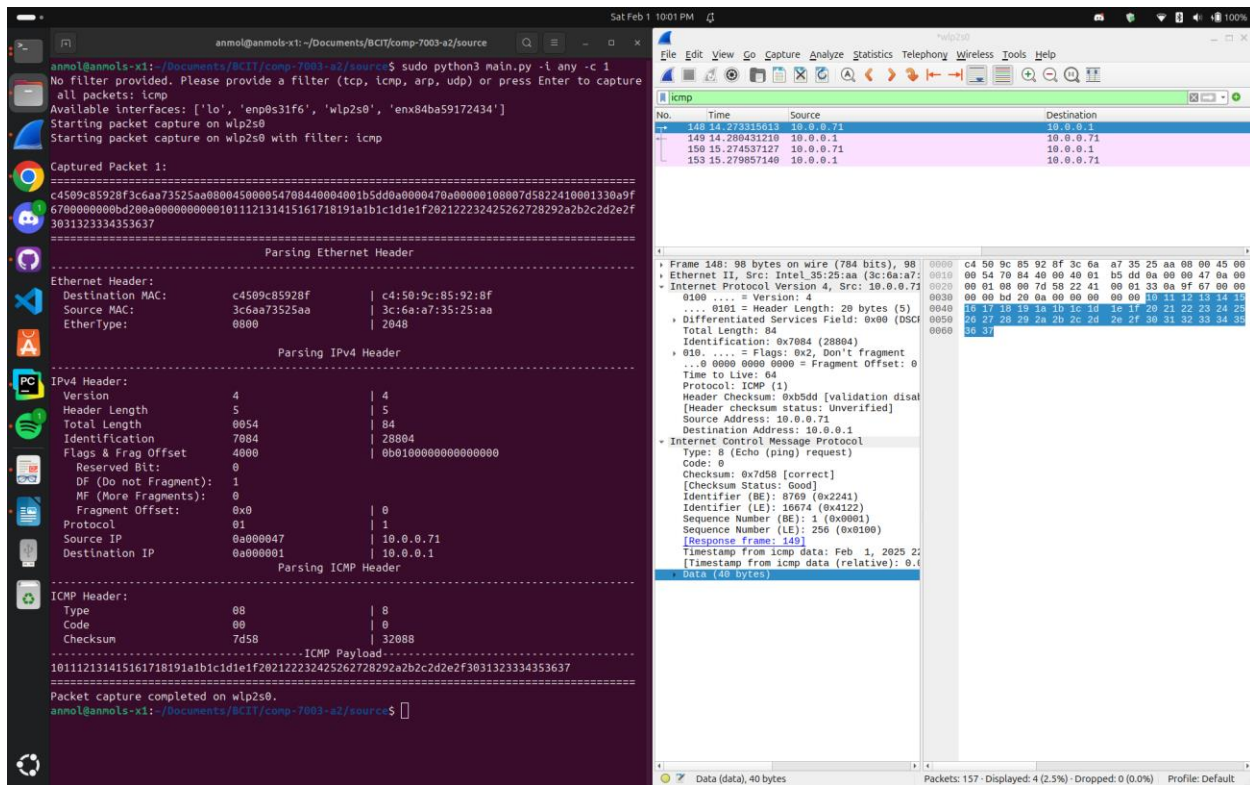
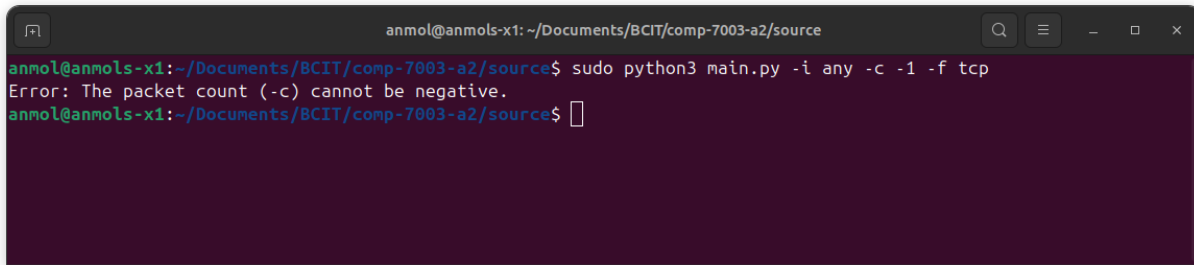
No.	Time	Source	Destination
131	22.003512970	10.0.0.71	52.94.225.242
132	22.074338375	52.94.225.242	10.0.0.71
135	23.004337983	10.0.0.71	52.94.225.242
136	23.075333772	52.94.225.242	10.0.0.71

Frame 131: 80 bytes on wire (784 bits), 80 bytes captured (784 bits) on interface wlp2s0  
Ethernet II, Src: Intel\_35:25:aa (3c:6a:a7:35:25:aa), Dst: VantivaConne\_85:92:8f (c4:50:9c:85:92:8f), Type: IPv4 (0x0800)  
Source: Intel\_35:25:aa (3c:6a:a7:35:25:aa), Destination: VantivaConne\_85:92:8f (c4:50:9c:85:92:8f), Type: IPv4 (0x0800)  
Internet Protocol Version 4, Src: 10.0.0.71, Dst: 52.94.225.242  
..... = Header Length: 20 bytes (5)  
..... 0101 = Differentiated Services Field: 0x00 (DSCP: CS0) Total Length: 84  
Identification: 0xaa5 (43685)  
Type: 0 (Echo (ping) request)  
Code: 0  
Checksum: 0x9d03 [correct]  
[Checksum Status: Good]  
Identifier (BE): 8971 (0x230b)  
Identifier (LE): 2851 (0x0b23)  
Sequence Number (BE): 1 (0x0001)  
Sequence Number (LE): 256 (0x0100)  
[Response Frame: 132]  
Timestamp from icmp data: Feb 1, 2025 2:22:23.003512970 (relative: 0.4)  
Data (48 bytes)

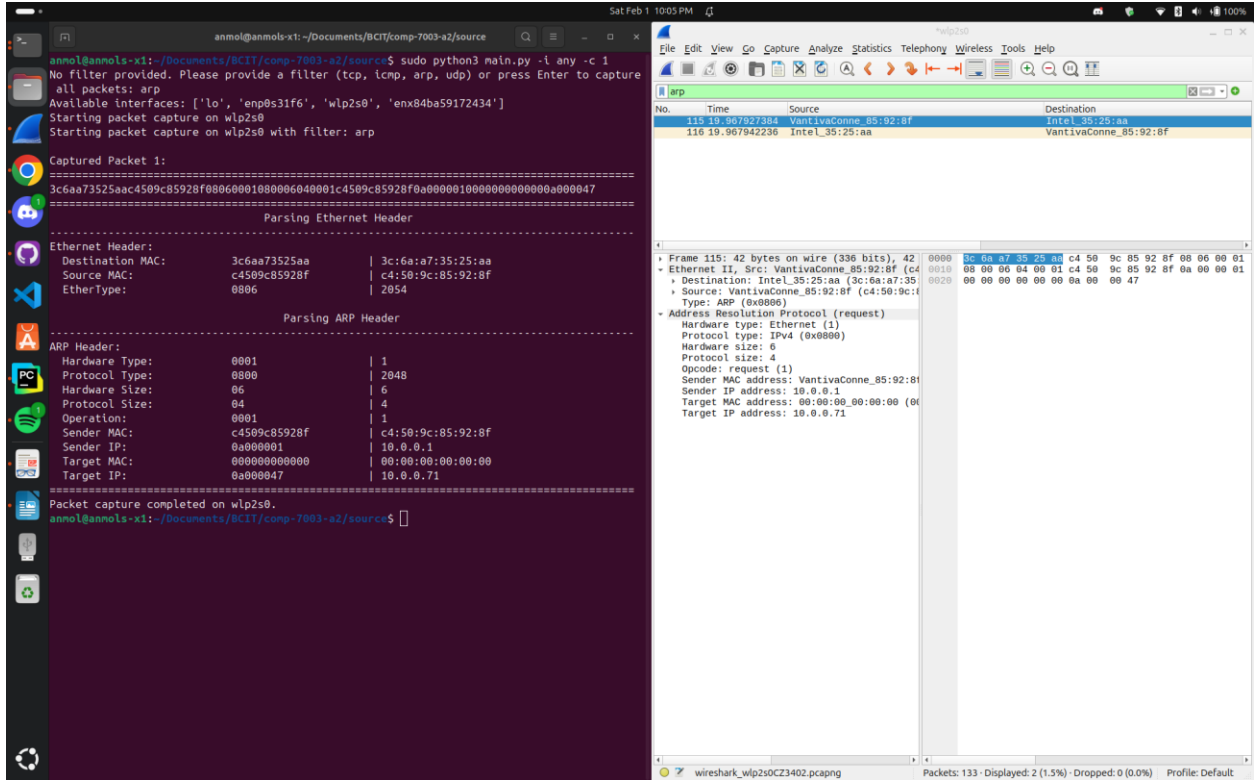
Data (data), 40 bytes

Packets: 138 - Displayed: 4 (2.9%) - Dropped: 0 (0.0%) Profile: Default

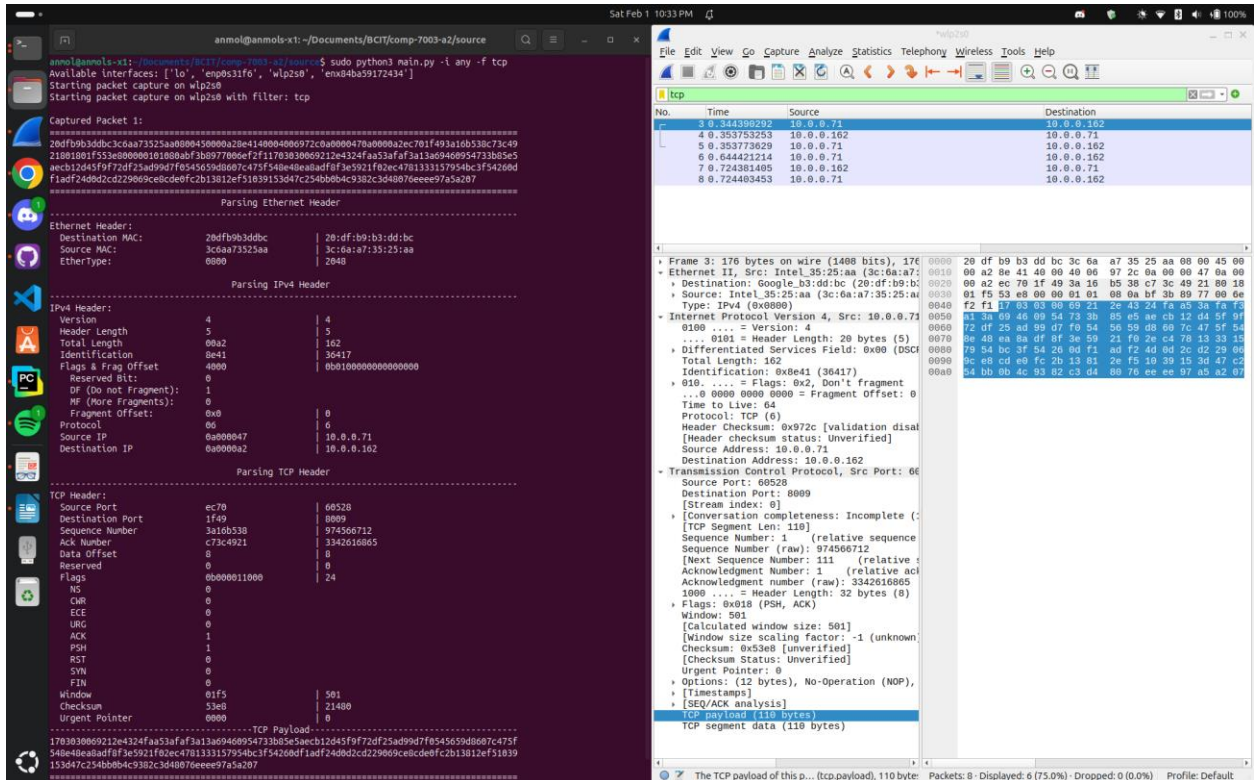
## Test 08



# Test 09

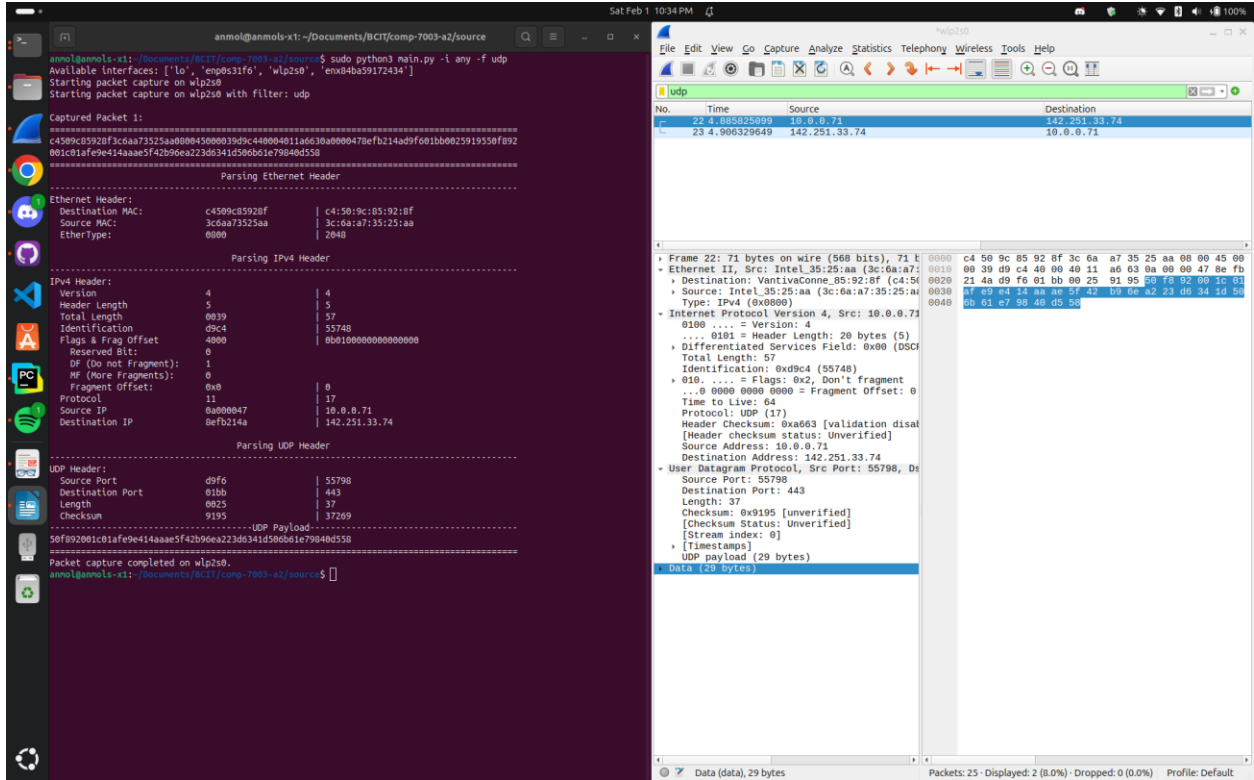


# Test 10

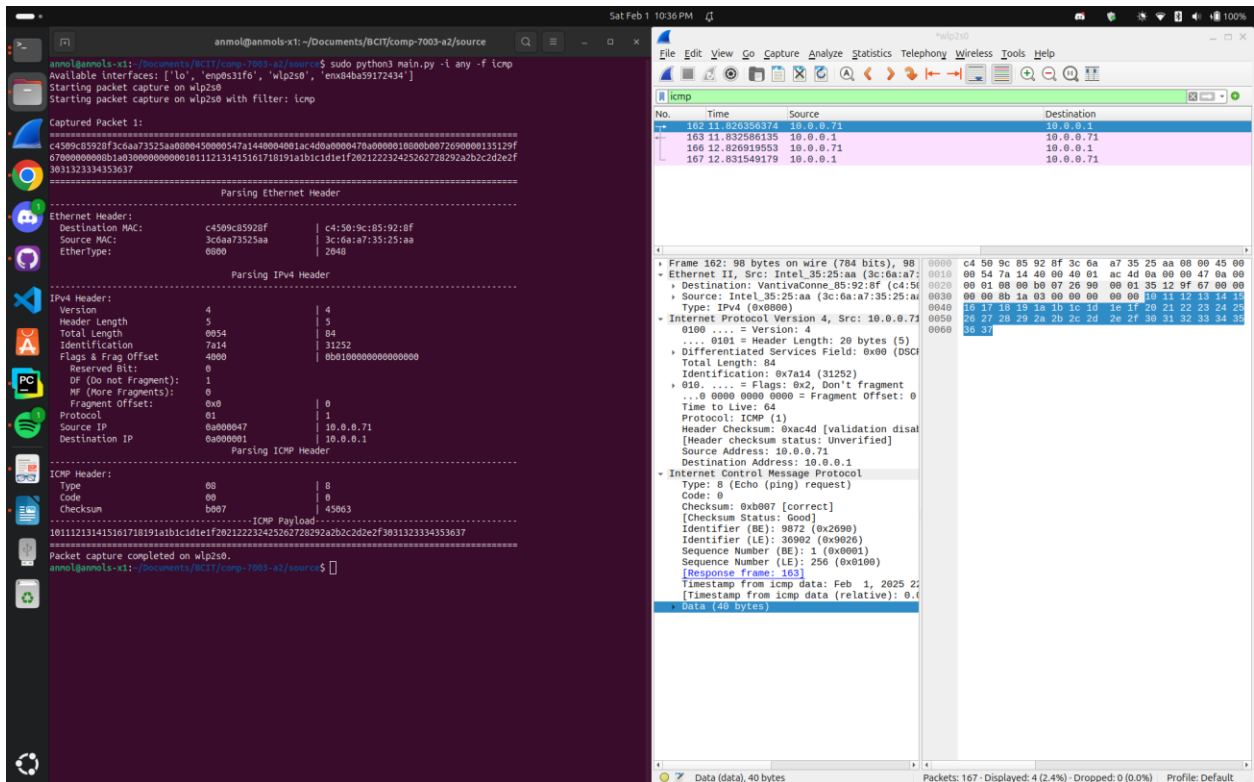




## Test 11



## Test 12





The image shows a Kali Linux desktop with two terminal windows. The left window, titled 'anmol@anmol-x1: ~/Documents/BCIT/comp-7003-a2/source', shows the execution of a packet capture on the 'arp' interface. It displays the captured packet 1, which is an ARP request from 3c6aa73525aa to c4509c8592bf. The packet details include Ethernet II header and ARP header information. The right window, titled 'anmol@anmol-x1: ~/Documents/BCIT/comp-7003-a2/sources', shows a detailed view of the ARP request packet. It displays the packet structure, including the Ethernet II header and the ARP request details. The packet is identified as 'Frame 34: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0'. The ARP request is from 3c6aa73525aa to c4509c8592bf, and it is a request for the IP address 10.0.0.1. The packet details include the Ethernet II header and the ARP request details.

The image displays two side-by-side windows from a Kali Linux desktop environment.

**Left Window (Terminal):**

- Title Bar:** anmol@anmol-x1: ~/Documents/BCTI/comp-7003-a2/source
- Command Prompt:** anmol@anmol-x1:~/Documents/BCTI/comp-7003-a2/source\$ sudo python3 main.py -h
- Output:**
  - Packet sniffer using Scapy with manual HEX parsing
  - options:
    - h, --help show this help message and exit
    - l INTERFACE, --interface INTERFACE The interface to capture packets on (e.g., eth0, wlan0, any) (default: any)
    - f FILTER, --filter FILTER BPF filter to apply (e.g., 'tcp, udp, arp, icmp'). If not provided, captures all packets.
    - c COUNT, --count COUNT Number of packets to capture (default: 1)
  - No filter provided. Please provide a filter (tcp, icmp, arp, udp) or press Enter to capture all packets: udp
  - Available interfaces: ['lo', 'enps31f0', 'wlp250', 'ens4ba9172434']
  - Starting packet capture on wlp250
  - Starting packet capture on wlp250 with filter: udp
  - Captured Packet 1:
  - Raw hex dump: ffffffff72dfbb3dbcb00b45000039000a00004011301300000a2ffffffff720f270f0023b575d0f201f00bf9af7dsef94bd1bc0fc9ce5ee0fe187eca00bf00fb6
  - Parsing Ethernet Header
    - Ethernet Header:
      - Destination MAC: ff:ff:ff:ff:ff:ff
      - Source MAC: 28:df:b9:b3:dd:bc
      - EtherType: 0800
    - Parsing IPv4 Header
      - IPv4 Header:
        - Version: 4
        - Header Length: 5
        - Total Length: 0039
        - Identification: 0000
        - Flags & Frag Offset: 4000
        - Reserved Bit: 0
        - DF (do not Fragment): 1
        - MF (More Fragments): 0
        - Fragment Offset: 0
        - Protocol: 11
        - Source IP: 0a0000a2
        - Destination IP: ffffffff
      - Parsing UDP Header
        - UDP Header:
          - Source Port: 270f
          - Destination Port: 270f
          - Length: 0025
          - Checksum: 0375
        - UDP Payload:
          - d0f2a1f00bf9af7dsef94bd1bc0fc9ce5ee0fe187eca00bf00fb6
  - Packet capture completed on wlp250.
  - anmol@anmol-x1:~/Documents/BCTI/comp-7003-a2/source\$

**Right Window (Wireshark):**

- Title Bar:** \*wlp250
- Menu Bar:** File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
- Status Bar:** Packets: 114 · Displayed: 68 (59.6%) · Promoted: 0 (0.0%) Profile: Default
- Main Area:**
  - Packets List Panel:** Shows a list of captured packets. Packet No. 109 is selected, showing Time 78.712798764, Source 142.251.211.227, and Destination 10.0.0.71.
  - Packets Detail Panel:** Displays the protocol stack for the selected packet:
    - Ethernet II, Src: Google b3:dd:bc (28:df:b9:00:00:00:00:00), Dst: Broadcast ff:ff:ff:ff:ff:ff
    - Type: IPv4 (0x0000)
      - Source: Google b3:dd:bc (20:df:b9:b3:dd:bc)
      - Type: IPv4 (0x0000)
      - Internet Protocol Version 4, Src: 10.0.0.10
        - 0100 .... = Version: 4
        - 0101 ... = Header Length: 20 bytes (5)
        - Differentiated Services Field: 0x00 (DSCP) Total Length: 57
        - Identification: 0x0000 (0)
        - 010 .... = Flags: 0x2, Don't fragment
        - ... 0000 0000 0000 = Fragment Offset: 0
        - Time to Live: 64
        - Protocol: UDP (17)
        - Header Checksum: 0x3013 [validation disabled] [Header checksum status: Unverified]
        - Source Address: 10.0.0.102
        - Destination Address: 255.255.255.255
      - User Datagram Protocol, Src Port: 9999, Dst Port: 9999
        - Length: 37
        - Checksum: 0xb575 [unverified] [Checksum status: Unverified]
        - [Stream index: 14]
        - [Timestamps]
      - UDP payload (29 bytes)
      - TP-Link Smart Home Protocol