

COMP 7003

Introduction to Information and Network Security

Assignment-02

User Guide

Anmol Mittal

A01397754

February 2nd, 2025

Course Reference Number (CRN): 91662

	2
Purpose	3
Installation	3
Building	3
Running	3
Command Line Arguments	3
main.py	3
Examples	4

Purpose

The purpose of the program is to capture and analyze network traffic at the packet level using Python and Scapy. It will filter packets by protocol (Ethernet, IPv4, ICMP, TCP, UDP, DNS, IPv6, ICMPv6), convert raw packet data into hexadecimal dumps, and parse the packet headers to extract and display key fields such as source/destination MAC and IP addresses, protocol-specific details, and port numbers. The program aims to provide a clear, structured, and human-readable output of packet information.

Installation

Navigate to <https://learn.bcit.ca>. Download COMP7003-assign02-v1.zip and Extract the contents.

Requirements

Need Scapy installed.

Building

No building required

Running

```
sudo python3 main.py -i <interface> -f <filter> -c <count>
```

Command Line Arguments

The following configuration values can be set in <file>:

main.py

Variable	Purpose
<-i> or <--interface>	Specifies the network interface to capture packets on. (Default: any)
<-f> or <--filter>	Specifies the BPF to apply. Common filters include tcp, udp, icmp, arp, ip, ip6, icmp6, and dns.
<-c> or <--count>	Specifies the number of packets to capture. (Default: 1)

Examples

main.py

```

anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$ sudo python3 main.py -i any -c 1 -f udp
Available interfaces: ['lo', 'enp0s31f6', 'wlp2s0']
Starting packet capture on wlp2s0
Starting packet capture on wlp2s0 with filter: udp

Captured Packet 1:
=====
01005e0000fb0a10c7ff82f808004500016bb5760000ff1119150a0000f0be00000fb14e914e901571ef3000084
0000000004000000004014401310143014101390142013301330143013201330144013501300143013001300130
0130013001300130013001300130013001300130013001300130013001300130013001300130013001300130
94000c0469506164056c6f63616c00033235310130013002313007696e2d61646472c050000c80010000119400
02c060013201350138013001450137014301390146013101420134014201420134013001300130014601330138
01380137013501380130014401330134013001360132c04c000c8001000011940002c060014601380134014501
30013001300130013001300130013001300130013001300130c0ab000c8001000011940002c060c00c002f800100001194
0006c00c00020008c06c002f8001000011940006c06c00020008c08d002f8001000011940006c08d00020008c0
db002f8001000011940006c0db00020008
=====
Parsing Ethernet Header
-----
Ethernet Header:
Destination MAC:      01005e0000fb      | 01:00:5e:00:00:fb
Source MAC:           0a10c7ff82f8      | 0a:10:c7:ff:82:f8
EtherType:            0800              | 2048

Parsing IPv4 Header
-----
IPv4 Header:
Version               4                  | 4
Header Length         5                  | 5
Total Length          016b               | 363
Identification        b576               | 46454
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              0a0000fb          | 10.0.0.251
Destination IP         e00000fb          | 224.0.0.251

Parsing UDP Header
-----
UDP Header:
Source Port           14e9               | 5353
Destination Port      14e9               | 5353
Length                0157               | 343
Checksum              1ef3               | 7923
-----UDP Payload-----
0000840000000000400000004014401310143014101390142013301330143013201330144013501300143013001
3001300130013001300130013001300130013001300130013001300130013001300130013001300130013001300
00001194000c0469506164056c6f63616c00033235310130013002313007696e2d61646472c050000c80010000
11940002c060013201350138013001450137014301390146013101420134014201420134013001300130014601
33013801380137013501380130014401330134013001360132c04c000c8001000011940002c060014601380134
0145013001300130013001300130013001300130013001300130c0ab000c8001000011940002c060c00c002f800100
0011940006c00c00020008c06c002f8001000011940006c06c00020008c08d002f8001000011940006c08d0002
0008c0db002f8001000011940006c0db00020008
=====
Packet capture completed on wlp2s0.
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 arp
Available interfaces: ['lo', 'enp0s31f6', 'wlp2s0']
Starting packet capture on wlp2s0
Starting packet capture on wlp2s0 with filter: arp

Captured Packet 1:
=====
3c6aa73525aac4509c85928f08060001080006040001c4509c85928f0a000001000000000000a0000047
=====

Parsing Ethernet Header
-----
Ethernet Header:
Destination MAC:      3c6aa73525aa      | 3c:6a:a7:35:25:aa
Source MAC:          c4509c85928f      | c4:50:9c:85:92:8f
EtherType:           0806              | 2054

Parsing ARP Header
-----
ARP Header:
Hardware Type:        0001              | 1
Protocol Type:        0800              | 2048
Hardware Size:        06                | 6
Protocol Size:        04                | 4
Operation:           0001              | 1
Sender MAC:          c4509c85928f      | c4:50:9c:85:92:8f
Sender IP:           0a000001          | 10.0.0.1
Target MAC:          000000000000       | 00:00:00:00:00:00
Target IP:           0a000047          | 10.0.0.71
=====
Packet capture completed on wlp2s0.
anmol@anmols-x1:~/Documents/BCIT/comp-7003-a2/source$

```