ASSIGNMENT-3

Group Members:

Minam Faisal (21i-1901)

Momenah Saif (21i-1909)

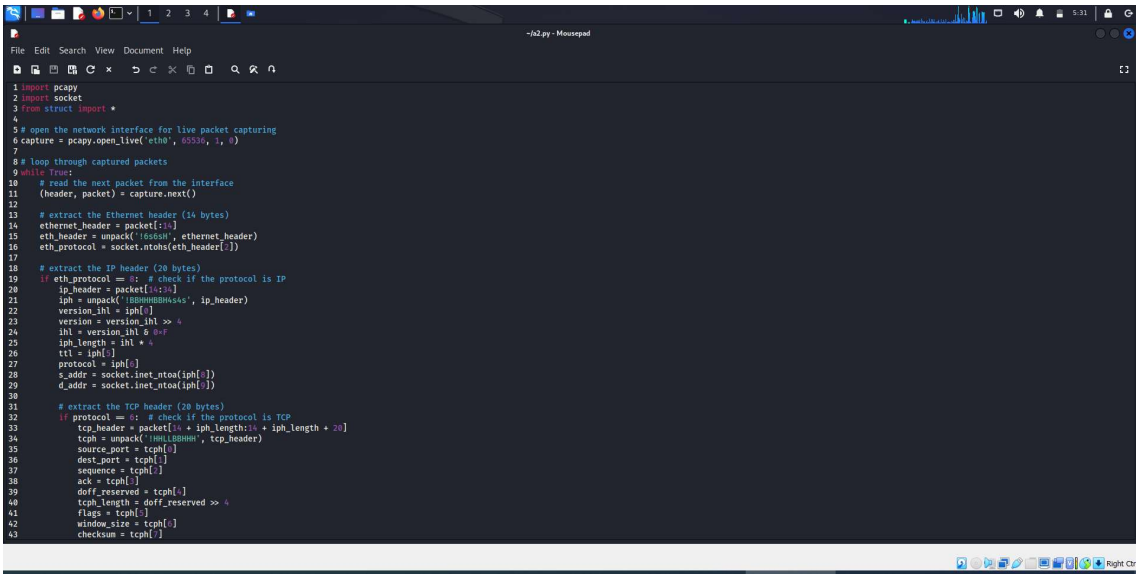
Section:

BS (CYS) “T”

Subject:

Computer Networks

Screenshots of Code



```
        tcph = unpack('!HHLLBBHHH', tcp_header)
        source_port = tcph[0]
        dest_port = tcph[1]
        sequence = tcph[2]
        ack = tcph[3]
        doff_reserved = tcph[4]
        tcph_length = doff_reserved >> 4
        flags = tcph[5]
        window_size = tcph[6]
        checksum = tcph[7]
        urgent_pointer = tcph[8]

        # print packet details
        print('Source IP: {}  , Destination IP: {},   Protocol: {},   Source Port: {},   Destination Port: {},   Flags: {}'.format(
            s_addr,  d_addr,  protocol,  source_port,  dest_port,  flags))
```

Explanation of Code:

Operating System:

We use Kali Linux to run our code.

Language:

Python

Import pcapy

Import socket

From struct import

- To begin we bring in three essential modules: pcapy, socket, and struct.
- Pcapy is a nifty extension module for Python that allows developers to access packets from network devices. Meanwhile socket is a standard module that provides easy access to the BSD socket interface.
- Finally struct is a handy tool for converting between Python values and C structs represented as Python bytes objects.

 # open the network interface for live packet capturing

Capture = pcapy.open_live ('eth0', 65536, 1, 0)

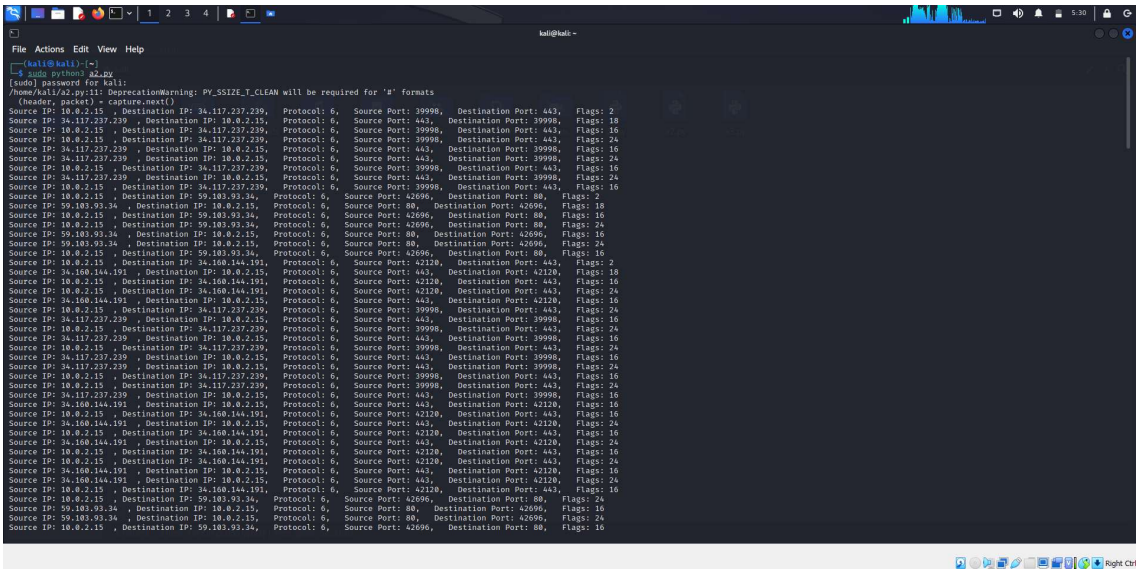
- For successful live packet capturing operations, it is imperative that system administrators first open up 'eth0' -the designated network interface- with an appropriate line of code that allows for optimal performance during data retrieval.
- This line should comprise specific parameters including one which specifies a maximum packet size limit of 65536 bytes (as per second parameter).
- Moreover, users seeking unrestricted access during capture sessions should modify third argument settings accordingly: if set at '1', eth0 enters promiscuous mode automatically detecting all packets irrespective of destination whilst being placed at '0' restricts access appropriately(useful for tight security).
- Finally, users can also include customized timeout values for added control by providing a desired duration measured in milliseconds as the fourth and final argument.

```
# loop through captured packets

While True:

   # read the next packet from the
interface

   (Header, packet) = capture.next ()
```

- The program employs a loop to seize packets from the interface until its discontinuation.
- To obtain information about the next captured packet, capture.next () furnishes a tuple consisting of both header and packet data.
- The packet's characteristics such as length and timestamp are contained within the header in form of metadata that exists as a dictionary.

Output

- For output, we wrote the command shown below in the screenshot.
- Then opened Firefox, open the Google Classroom.
- After that packets started showing as you can see in the screenshot below.



```
┌──(kali㉿kali)-[~]
└─$ sudo python3 a2.py
[sudo] password for kali:
/home/kali/a2.py:11: DeprecationWarning: PY_SSIZE_T_CLEAN will be required for '#' formats
  (header, packet) = capture.next()
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 2
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 18
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 24
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 16
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 16
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 59.103.93.34,   Protocol: 6,   Source Port: 42696,   Destination Port: 80,   Flags: 2
Source IP: 59.103.93.34  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 80,   Destination Port: 42696,   Flags: 18
Source IP: 10.0.2.15  , Destination IP: 59.103.93.34,   Protocol: 6,   Source Port: 42696,   Destination Port: 80,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 59.103.93.34,   Protocol: 6,   Source Port: 42696,   Destination Port: 80,   Flags: 24
Source IP: 59.103.93.34  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 80,   Destination Port: 42696,   Flags: 16
Source IP: 59.103.93.34  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 80,   Destination Port: 42696,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 59.103.93.34,   Protocol: 6,   Source Port: 42696,   Destination Port: 80,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 2
Source IP: 34.160.144.191  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 42120,   Flags: 18
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 24
Source IP: 34.160.144.191  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 42120,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 24
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 24
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 24
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 16
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.117.237.239,   Protocol: 6,   Source Port: 39998,   Destination Port: 443,   Flags: 24
Source IP: 34.117.237.239  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 39998,   Flags: 16
Source IP: 34.160.144.191  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 42120,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 16
Source IP: 34.160.144.191  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 42120,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 16
Source IP: 34.160.144.191  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 42120,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 24
Source IP: 34.160.144.191  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 42120,   Flags: 16
Source IP: 34.160.144.191  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 443,   Destination Port: 42120,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 34.160.144.191,   Protocol: 6,   Source Port: 42120,   Destination Port: 443,   Flags: 16
Source IP: 10.0.2.15  , Destination IP: 59.103.93.34,   Protocol: 6,   Source Port: 42696,   Destination Port: 80,   Flags: 24
Source IP: 59.103.93.34  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 80,   Destination Port: 42696,   Flags: 16
Source IP: 59.103.93.34  , Destination IP: 10.0.2.15,   Protocol: 6,   Source Port: 80,   Destination Port: 42696,   Flags: 24
Source IP: 10.0.2.15  , Destination IP: 59.103.93.34,   Protocol: 6,   Source Port: 42696,   Destination Port: 80,   Flags: 16
```

Reference of Code:

CHATGPT

Constrains

- Any code that we took from the internet we couldn’t run it.
- We tried to use visual studio and downloaded pcapy library file and tried to run a c++ code but we were not able to run it.
- Next we tried to use visual studio code and downloaded many library files to run the code but we failed.
- We were no able to run any code in c++ language on any software.
- We had only studied c++ till now so we couldn’t use any other language.
- At last we had to use python language as the code was only running in that language.
- We took the python language code ran it and tried our best to understand the code from internet.

THE END