College of Engineering,Trivandrum

Experiment 15

Network Programming Lab

Author:
Alan Anto

Registration Number :
TVE16CS09

April 24, 2019

Contents

1 Packet capturing and filtering application 2
1.1 Aim 2
1.2 Theory 2
1.2.1 Packet Capturing 2
1.2.2 Algorithm 2
1.2.3 Program 3
2 Output 5
3 Result 5

1 Packet capturing and filtering application

1.1 Aim

Develop a packet capturing and filtering application using raw sockets .

1.2 Theory

1.2.1 Packet Capturing

Packet Capture is a networking term for intercepting a data packet that is crossing a specific point in a data network. Once a packet is captured in real-time, it is stored for a period of time so that it can be analyzed, and then either archived or discarded. Packets are captured and examined to help diagnose and solve network problems such as:

1.Identifying security threats

2.Troubleshooting undesirable network behaviors

3.Identifying network congestion

4.Identifying data/packet loss

5.Forensic network analysis

Once a packet is captured, it is stored temporarily so that it can be analyzed. The packet is inspected to help diagnose and solve network problems and determine whether network security policies are being followed.

1.2.2 Algorithm

1.START

2.CREATE SOCKET

3.RECEIVE all packets

4.UNPACK the packets

5.FORMAT the packets

6.PRINT the filtered data

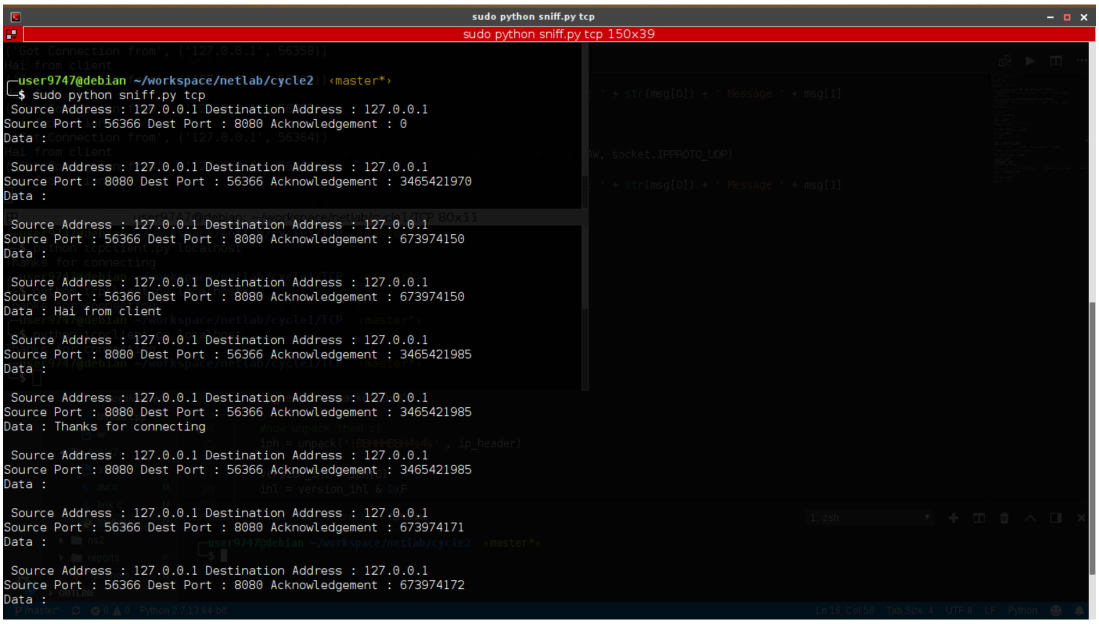
7.STOP

1.2.3 Program

```
import socket, sys
from struct import *

if (sys.argv[1]=="tcp"):
        try:
                s = socket.socket(socket.AF_INET, socket.SOCK_RAW,
                socket.IPPROTO_TCP)
        except socket.error,msg:
                print 'Socket could not be created. Error Code : '
                + str(msg[0]) + ' Message ' + msg[1]
                sys.exit()
elif(sys.argv[1]=="udp"):
        try:
                s = socket.socket(socket.AF_INET, socket.SOCK_RAW,
                socket.IPPROTO_UDP)
        except socket.error , msg:
                print 'Socket could not be created. Error Code : '
                + str(msg[0]) + ' Message ' + msg[1]
                sys.exit()
else:
        print "Specify protocol"


# receive a packet
while True:
        packet = s.recvfrom(65565)

        packet = packet[0]

        ip_header = packet[0:20]

        iph = unpack('!BBHHHBBH4s4s' , ip_header)

        version_ihl = iph[0]
        ihl = version_ihl & 0xF
```

```
iph_length = ihl * 4


s_addr = socket.inet_ntoa(iph[8]);
d_addr = socket.inet_ntoa(iph[9]);

print ' Source Address : ' + str(s_addr) + '
Destination Address : ' + str(d_addr)

tcp_header = packet[iph_length:iph_length+20]

tcph = unpack('!HHLLBBHHH' , tcp_header)

source_port = tcph[0]
dest_port = tcph[1]
acknowledgement = tcph[3]
doff_reserved = tcph[4]
tcph_length = doff_reserved >> 4

print 'Source Port : ' + str(source_port) + ' Dest Port : ' +
str(dest_port) + ' Acknowledgement : ' + str(acknowledgement)

h_size = iph_length + tcph_length * 4
data_size = len(packet) - h_size

#get data from the packet
data = packet[h_size:]

print 'Data : ' + data
print
```

2 Output



```
┌─user9747@debian ~/workspace/netlab/cycle2  ‹master*›
└─$ sudo python sniff.py tcp
 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 56366 Dest Port : 8080 Acknowledgement : 0
Data :

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 8080 Dest Port : 56366 Acknowledgement : 3465421970
Data :

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 56366 Dest Port : 8080 Acknowledgement : 673974150
Data :

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 56366 Dest Port : 8080 Acknowledgement : 673974150
Data : Hai from client

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 8080 Dest Port : 56366 Acknowledgement : 3465421985
Data :

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 8080 Dest Port : 56366 Acknowledgement : 3465421985
Data : Thanks for connecting

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 8080 Dest Port : 56366 Acknowledgement : 3465421985
Data :

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 56366 Dest Port : 8080 Acknowledgement : 673974171
Data :

 Source Address : 127.0.0.1 Destination Address : 127.0.0.1
Source Port : 56366 Dest Port : 8080 Acknowledgement : 673974172
Data :
```

3 Result

Packet catching and filtering application was implemented using python and output was obtained.