Packet capturing and filtering application

Aim

To develop a packet capturing and filtering application using raw sockets.

Theory

Packet : A packet is a basic unit of communication over a digital network. When data has to be transmitted, it is broken down into similar structures of data before transmission, called packets, which are reassembled to the original data chunk once they reach their destination.

Libpcap : libpcap allows us to capture or send packets from a live network device or a file. This package is aimed at Debian based Linux distributions but may also work on Mac OSX. Not intended for Windows, but WinPcap is a port that is available. Compiling a pcap program requires linking with the pcap lib.

You can install it in Debian based distributions with
sudo apt-get install libpcap-dev

Algorithm

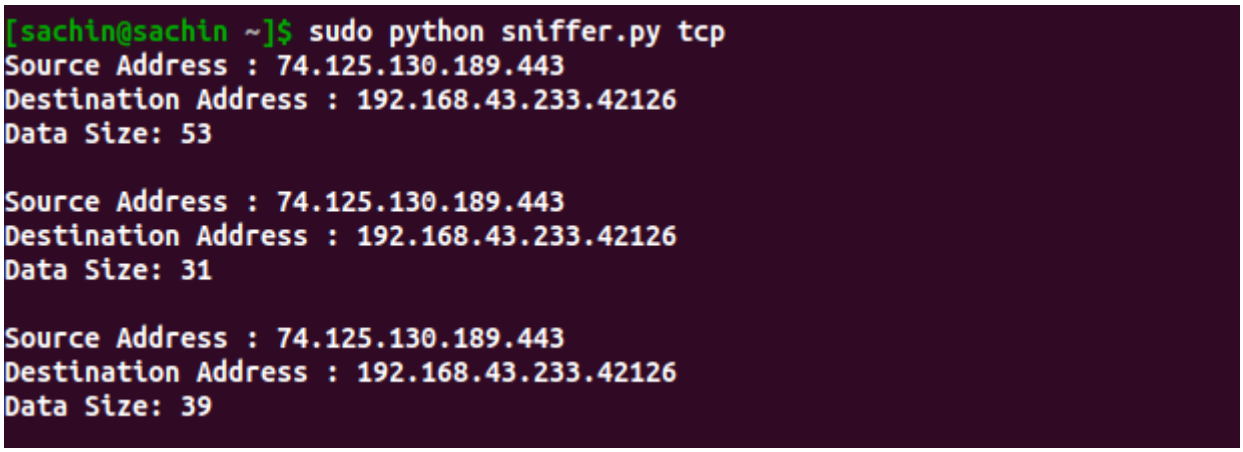
1.START
2.CREATE SOCKET
3.RECEIVE all packets
4.UNPACK the packets
5.FORMAT the packets
6.PRINT the filtered data
7.STOP

Code

```
import socket, sys
from struct import *
try:
      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()
except IndexError,msg:
      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]);

      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 Address : ' + str(s_addr)+'.'+ str(source_port) + '\
nDestination Address : ' + str(d_addr) +'.'+ str(dest_port)

      h_size = iph_length + tcph_length * 4
      data_size = len(packet) - h_size
      print 'Data Size: ' + str(data_size)
      #get data from the packet
      if(sys.argv[1]=='udp'):
            data = packet[h_size:]
            print 'Data : ' + data

      print
```

Output



Result

Developed a packet capturing and filtering application using raw sockets in python.