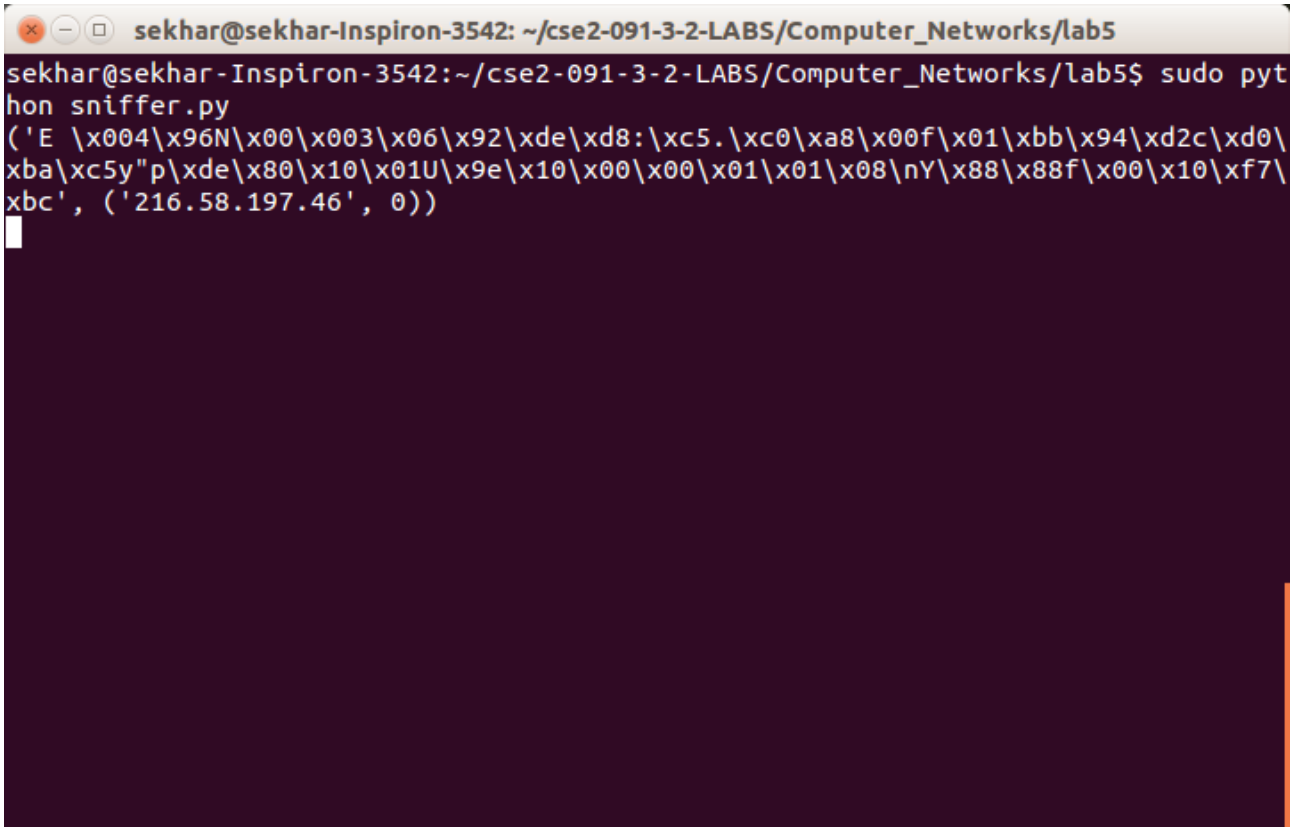
PROGRAM:-

```
#Packet sniffer in python
#For Linux
import socket
#create an INET, raw socket
s = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_TCP)
# receive a packet
while True:
        print s.recvfrom(65565)
```

OUTPUT:-



PROGRAM:-

```
#Packet sniffer in python for Linux
#Sniffs only incoming TCP packet
import socket, sys
from struct import *
#create an INET, STREAMing socket
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()
# receive a packet
while True:
	packet = s.recvfrom(65565)
#packet string from tuple
#	print "hi"
	packet = packet[0]
#take first 20 characters for the ip header
	ip_header = packet[0:20]
#now unpack them :)
	iph = unpack('!BBHHHBBH4s4s' , ip_header)
	version_ihl = iph[0]
	version = version_ihl >> 4
	ihl = version_ihl & 0xF
	iph_length = ihl * 4
	ttl = iph[5]
	protocol = iph[6]
	s_addr = socket.inet_ntoa(iph[8]);
	d_addr = socket.inet_ntoa(iph[9]);
	print 'Version : ' + str(version) + ' IP Header Length : ' + str(ihl) + ' TTL : '
+str(ttl) + ' Protocol : ' + str(protocol) + ' Source Address : ' + str(s_addr) +
'Destination Address : ' + str(d_addr)
	tcp_header = packet[iph_length:iph_length+20]
#now unpack them :)
	tcph = unpack('!HHLLBBHHH' , tcp_header)

	source_port = tcph[0]
	dest_port = tcph[1]
	sequence = tcph[2]
	acknowledgement = tcph[3]
```

```
        doff_reserved = tcph[4]
        tcph_length = doff_reserved >> 4
        print 'Source Port : ' + str(source_port) + ' Dest Port : ' + str(dest_port) +
'Sequence Number : ' + str(sequence) + ' Acknowledgement : ' +
str(acknowledgement) + 'TCP header length : ' + str(tcph_length)
        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
```

OUTPUT:-