CMPE-209(1)

PROJECT REPORT

NETWORK TRAFFIC ANALYZER

UNDER THE GUIDANCE OF

Prof. Chao Li Tarng

Date: 16th May 2016

Team 6

Name	SJSU ID
Ruta Dhekane	010714860
Gauri Bodkhe	010739729
Sanket Dhami	010652733
Keertikeya Gupta	010034700

ABSTRACT

By: Gauri Bodkhe, Sanket Dhami, Ruta Dhekane, Keertikeya Gupta

From recent decades, there has been a colossal increment in computer networks, its complexities and in number of clients using internet web services. This prompts an increase in internet traffic flowing across the globe. Additionally, there are situations where a client downloads from an illegitimate site and if got caught, it denies from doing as such. With a specific end goal to be keep system smooth and secure, it gets to be vital to monitor the network.

Our project, Network Traffic Analyzer, examines and parses packets or packet captured file obtained from Wireshark which are observed and extricates basic payload data from the packet. This report discusses about network administration over small networks like home, small campus networks, how it is performed using certain tools and python script and the possible outcomes or results of this project.

To understand the project, the report also displays the detailed steps for carrying out this project along with screenshots and observation made at every steps.

Additionally, this report also discusses the application of this project.

Contents

Table of Figures

1. OVERVIEW

Now-a-days there has been an enormous increase in the number of users using internet services and lead to more network traffic over the nodes. Network Traffic Analyzer is used for the work to monitor the packets by analyzing and parsing them to get the important payload information. There will be some cases where user will download from illegitimate website and would deny if got caught doing so. Hence it becomes necessary to monitor the network traffic for smooth and secure functioning of the service.

In this project the Network Traffic Analyzer is used to analyze and parse the packets. The packets are monitored to extracts critical payload information. The analyzer finds out any user downloading software or any files from the blacklisted sites. Packet details like source IP, destination IP are analyzed and finds the location from where the packet traverses. Thus it provides a proof of a particular user downloading data from illegal website which cannot be denied. All the information about the analysis of the packets and geographical location gets stored in the database. The destination of the packet will get displayed on google map for easy tracking.

2. OBJECTIVES AND SCOPE

The objective of the project is to analyze the captured packets and find out the users who are downloading software from blacklist/illegal websites. All packet details like source IP address, destination IP address are analyzed and then find the location of user. User cannot deny the truth of downloading software from blacklisted sites, as we have the proof of his/her packet traversal. In addition to this, we are providing provision to store geographical location and all the packet details in database for future use. A detailed plot of network traffic will be displayed on google maps.

3. ARCHITECTURE

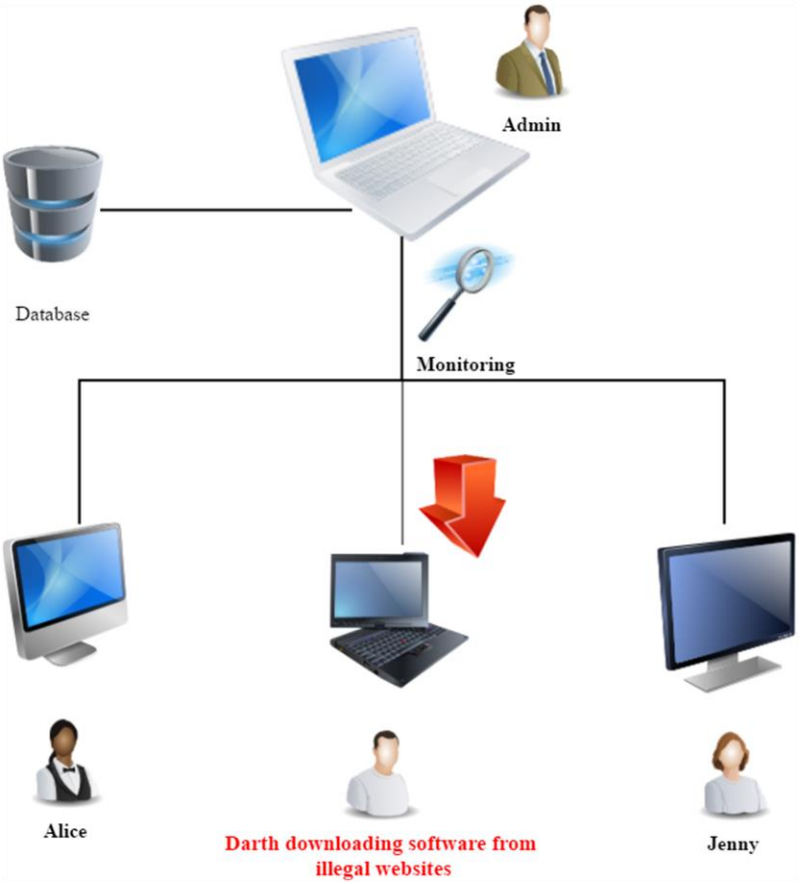

Figure 1 ***Architecture of Network Packet Analyzer***

The figure 1 describes the overall architectural working of the application. The admin continuously monitors the different activities performed by different users in the network. If any of the user is performing any illegal activity like downloading any software or files from blacklisted or illegal websites, then it gets detected at the admin side. Then by using PyGeoIp the admin correlates the IP address to Physical location of the destination of that packet. It is done by querying the database with a particular IP address. This returns the records containing the city, region name, postal code country name, latitude and longitude. Thus with help of it we could plot the location on Google earth. It gives a proof against the user doing the illegal activity and the user cannot deny from performing it.

The information like the URL, destination IP, MAC, Timestamp, latitude, longitude etc. is stored in database. It is stored in encrypted format to add on to the security of the system.

Also the database can be used for future analysis and creating statistics of the illegal activities performed which can help in enhancing the security.

3.1. Technologies and tools

1. The analyzer is developed in python version 2.7.
2. At the backend we have used MySql to store all the important information in the encrypted format for future purpose.
3. Also used GeoLite City database to provide visual location of destination using Google earth. PyGeoLite is used to correlate the IP address to physical location which retrieves the latitude n longitude city, region etc. of the user.
4. Wireshark which is an open source packet analyzer used to capture the packets using pcap.
5. Ettercap is an open source network security tool used in computers for security auditing and network protocol analysis. It is used for intercepting traffic on network segment, conducting active eavesdropping against a number of common protocols and capturing passwords. It can be run on systems like Linux, Mac, Windows, BSD and Solaris etc. We have used Ettercap for continuous monitoring of the activities of different users in the network.
6. Google map used to display the geographical location of the destination of packet.

3.2. Implementation Approach

1. Initially, the packets are captured by using Wireshark and live packets are using Ettercap.
2. Dpkt- a python module is used as an analyzing tool for parsing packets. It analyzes each individual packet captured and examines each protocol layer. It provides the IP address of the user downloading an application from illegal or blacklisted websites.
3. The geographical location of that IP address is found by using PyGeoIP which quires the GeoLiteCiy database.
4. Also socket libraries are used to resolve IP addresses to simple strings.
5. Google maps AP is used to give a geographical display of the destination of the packet that is being analyzed. All the information about the analyzed packet like its source and destination IP addresses, timestamp, URL, geographical latitude and longitude of the destination and the geographical location plotted gets stored in the database.
6. All the information that is stored in database is in encrypted format for security purposes.

4. RESULTS

4.1. Results

The results of the projects are as follows:

1. Location of each individual packet captured will be traced.
2. A track of all details of packets captured will be stored in database.
3. Identify a user' IP address and its location if he has downloaded any software or application from blacklisted websites.

4.2. Screenshots

1. Ettercap is used to capture live packets and for ARP poisoning.

Figure 2 Ettercap

2. ARP spoofing host and target selection using Ettercap.

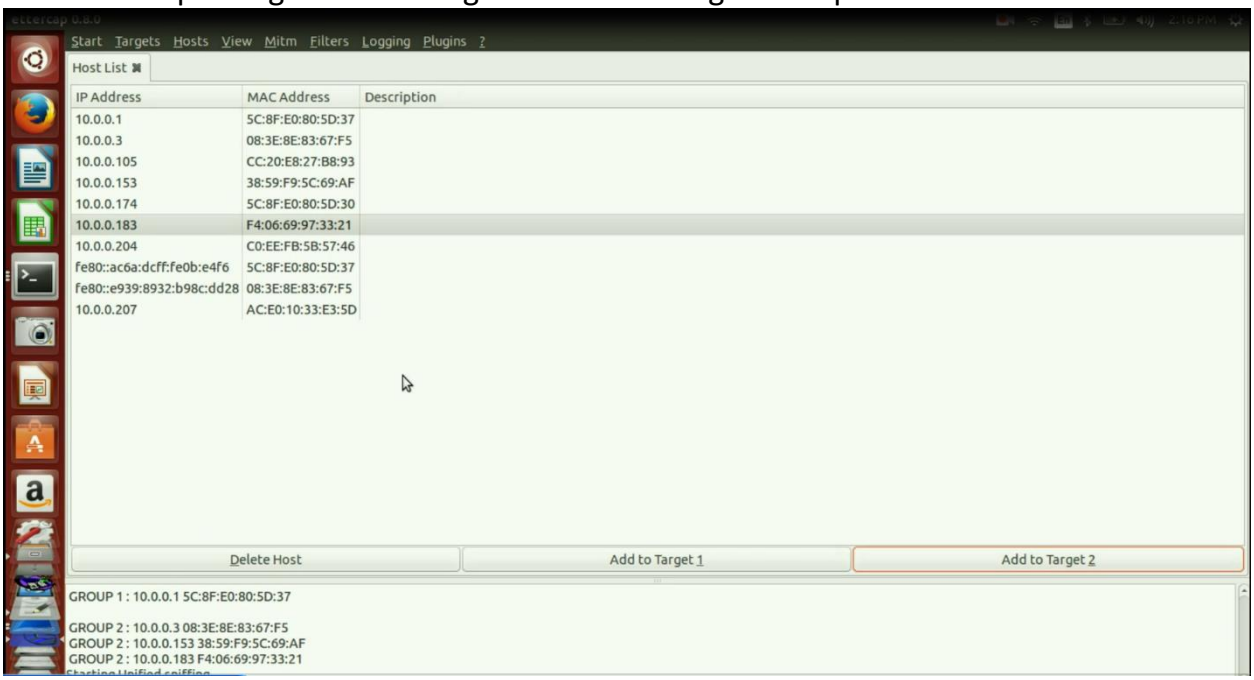

Figure 3 ***ARP spoofing using Ettercap***

3. Wireshark HTTP and TCP packet capturing.

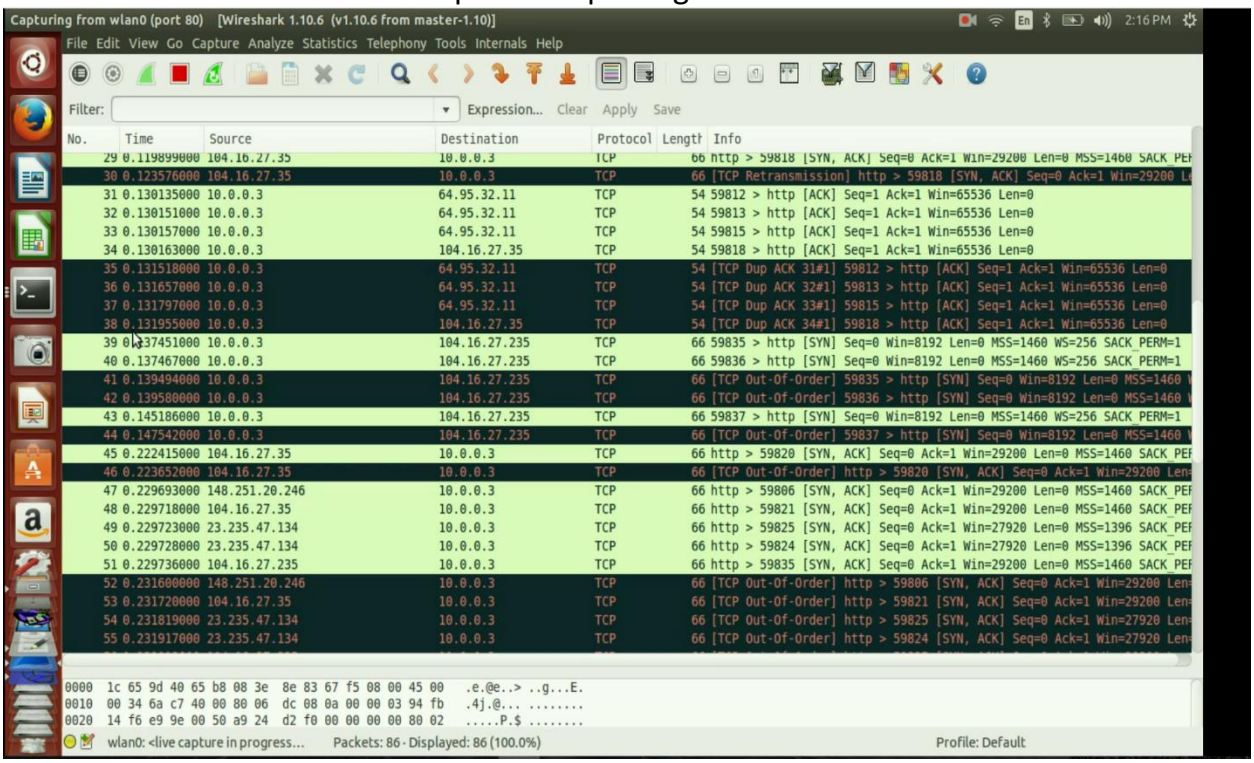

Figure 4 ***Wireshark capture***

4. List of blacklisted sites.

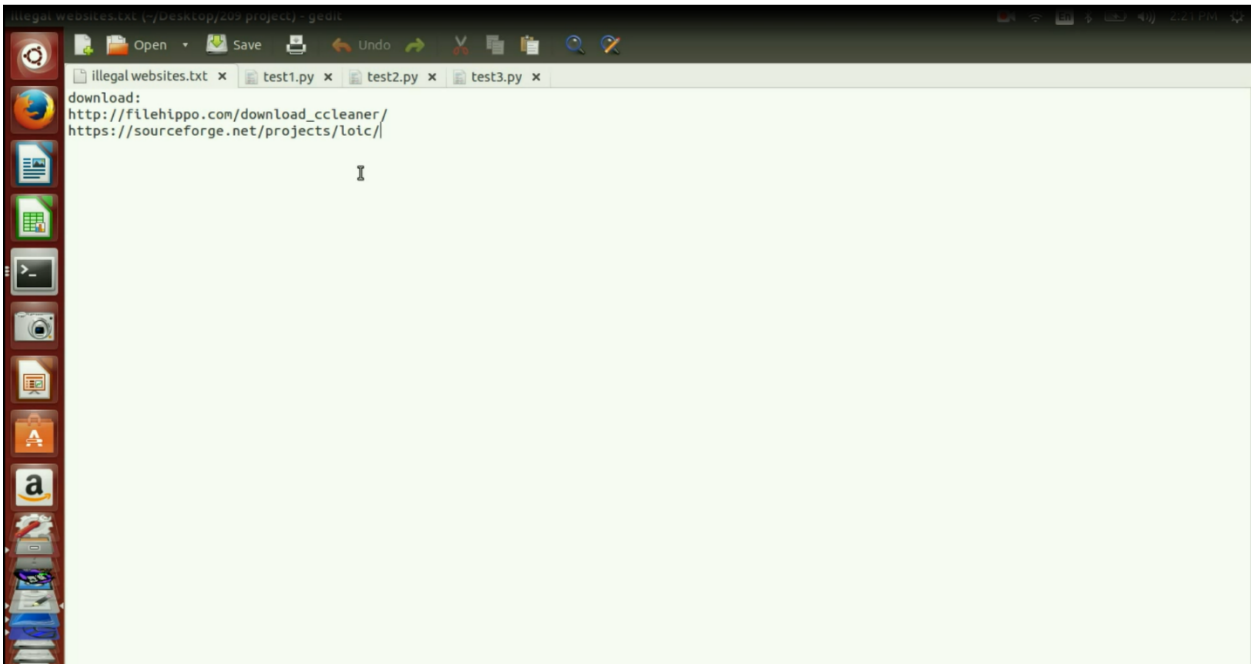

Figure 5 Blacklisted websites

5. User driven python script output menu.

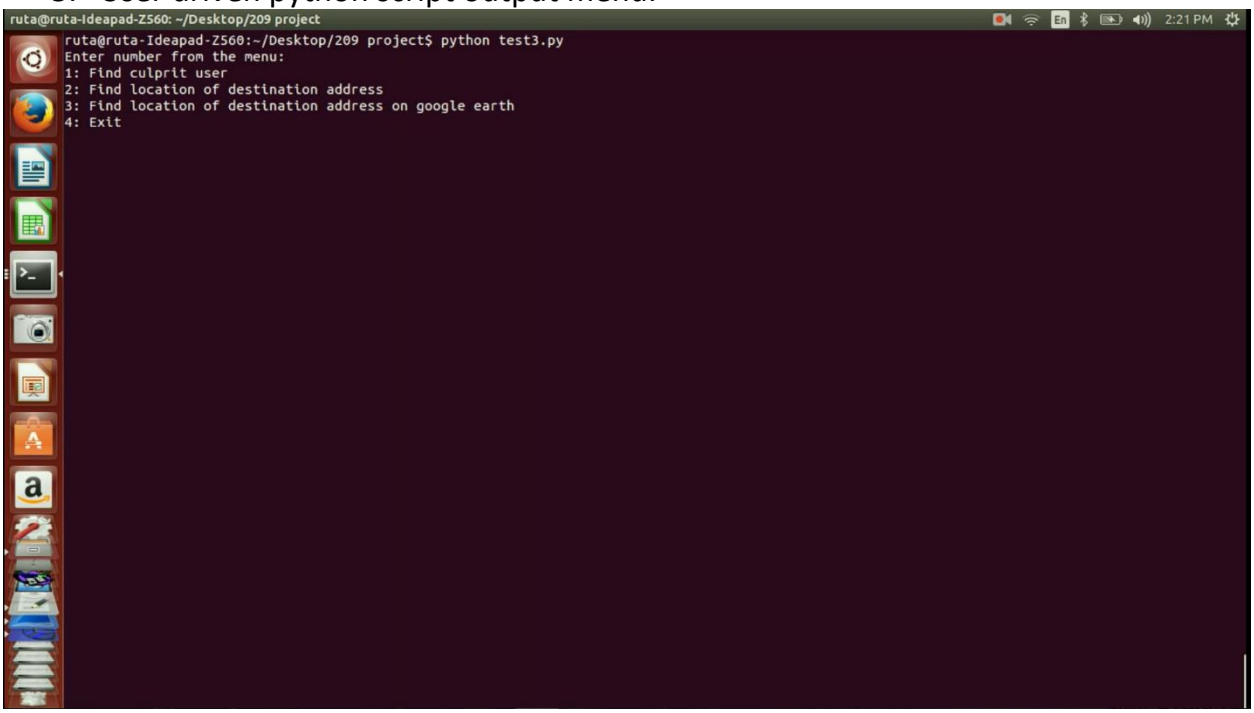

Figure 6 Python script

6. List of users who has downloaded software from illegal websites

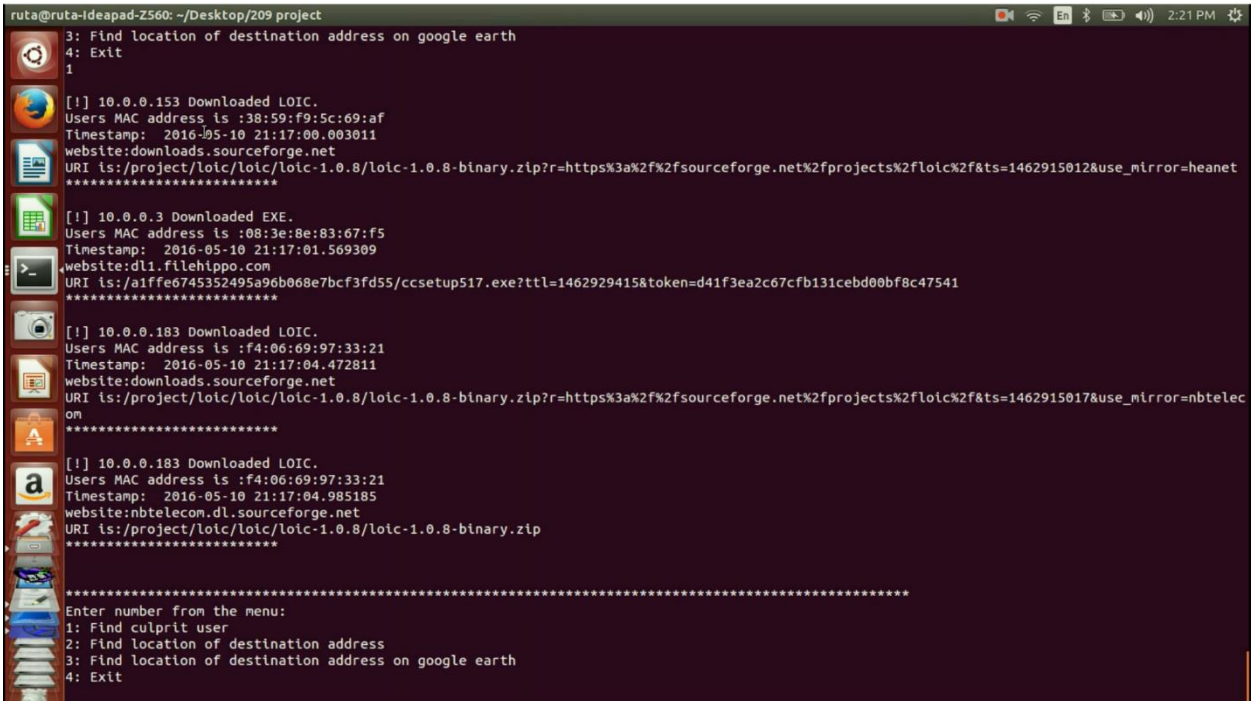

Figure 7 ***Output generated – Part I***

7. Packet and URL information.

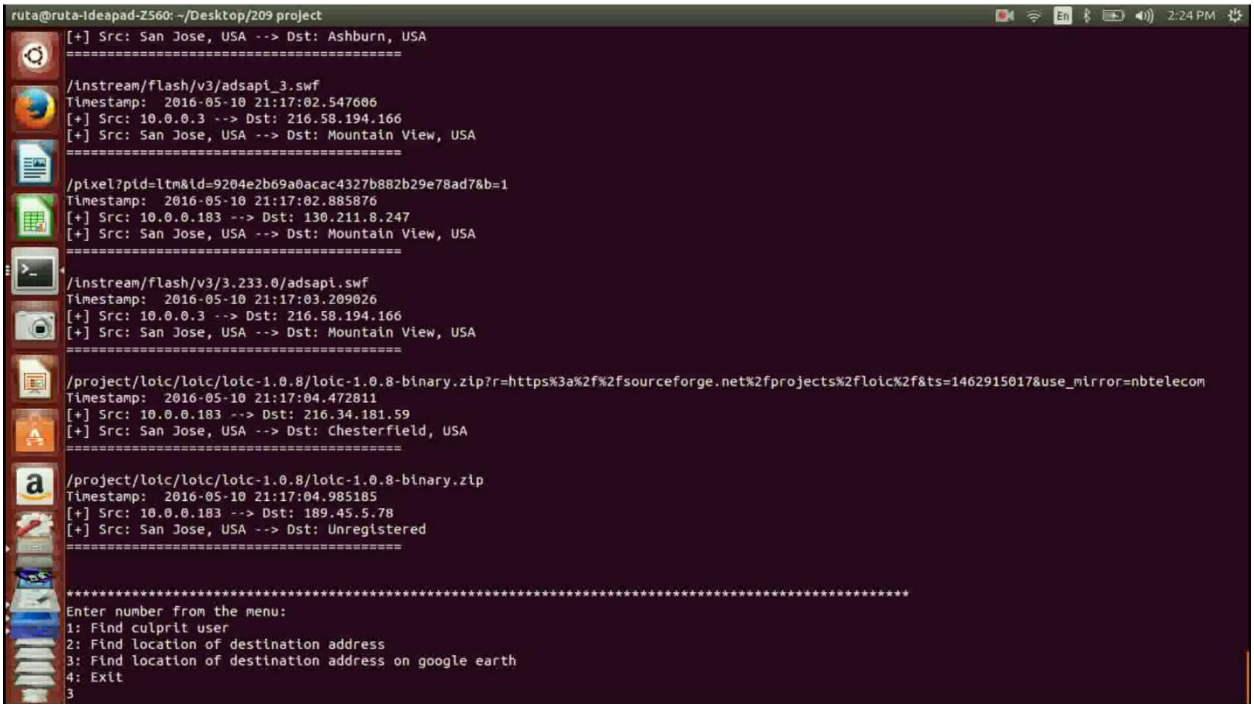

Figure 8 ***Output generated – Part II***

8. KML file is generated.

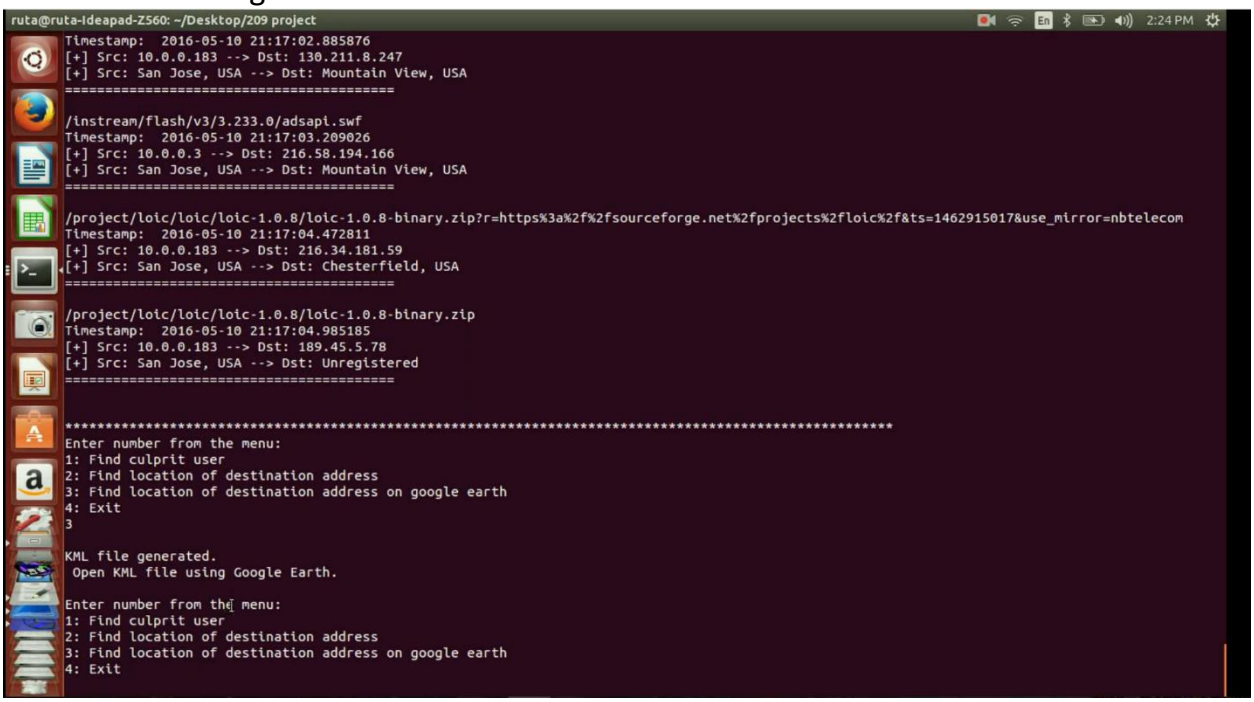

Figure 9 **Output generated – Part III**

9. Destination address on Google Earth.

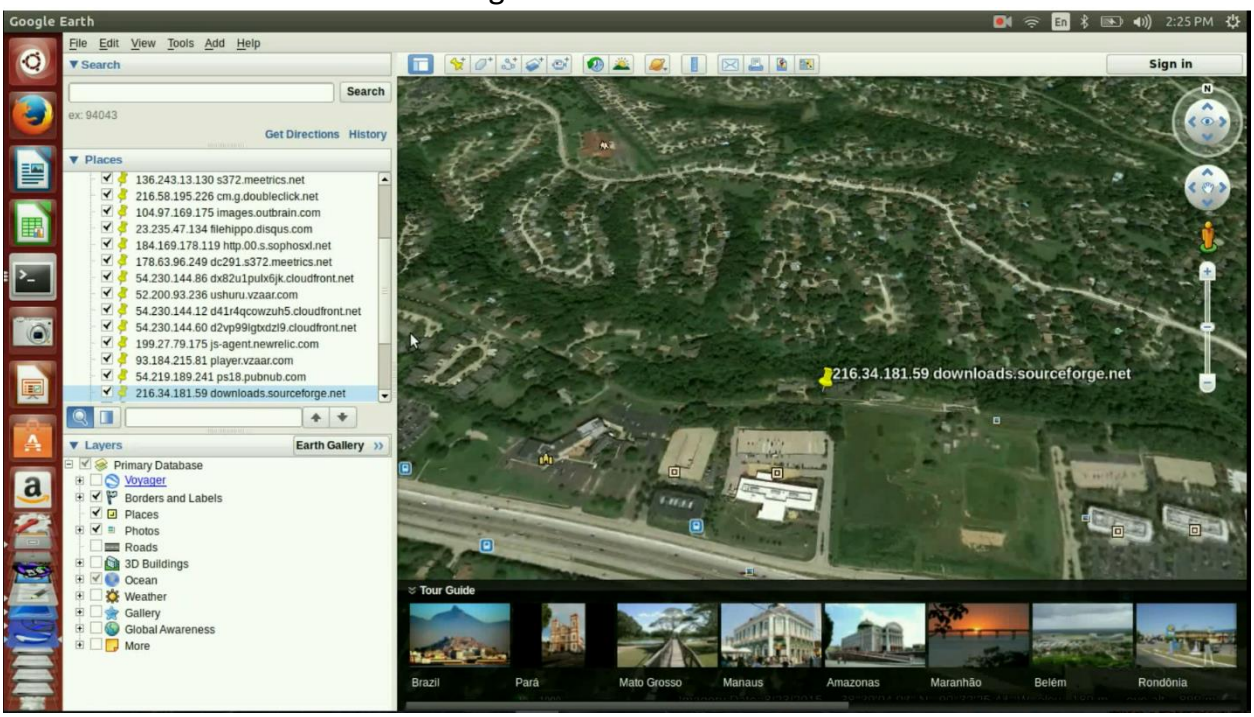

Figure 10 Google Earth

10. Database tables in encrypted format.

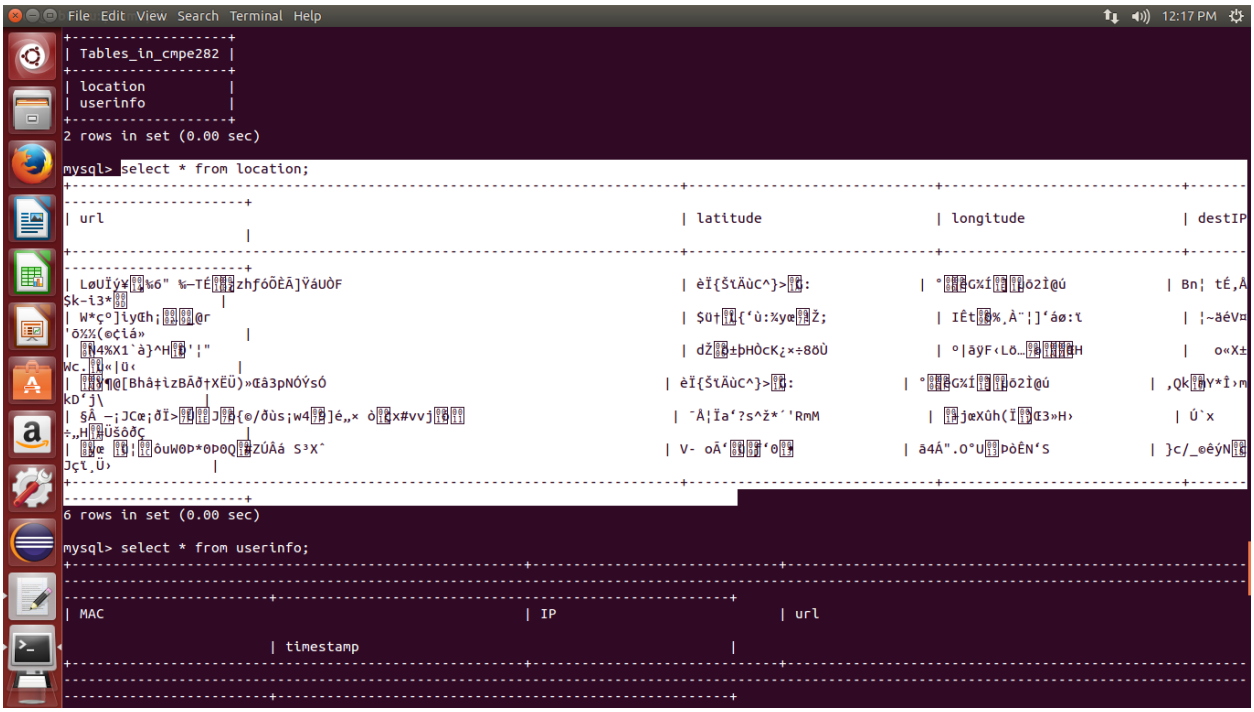

Figure 11 **Encryption in DB - I**

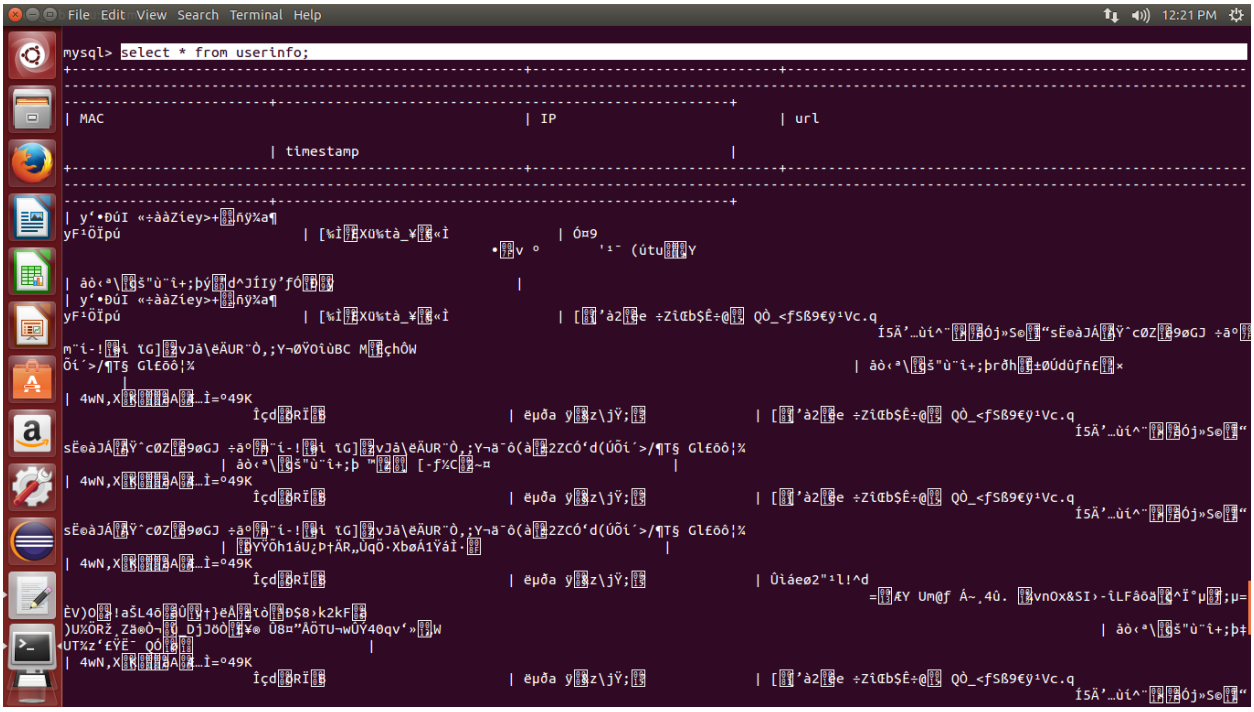

Figure 12 **Enccryption in DB - II**

Figure 13 **Encryption in DB - III**

11. Database tables in decrypted format.

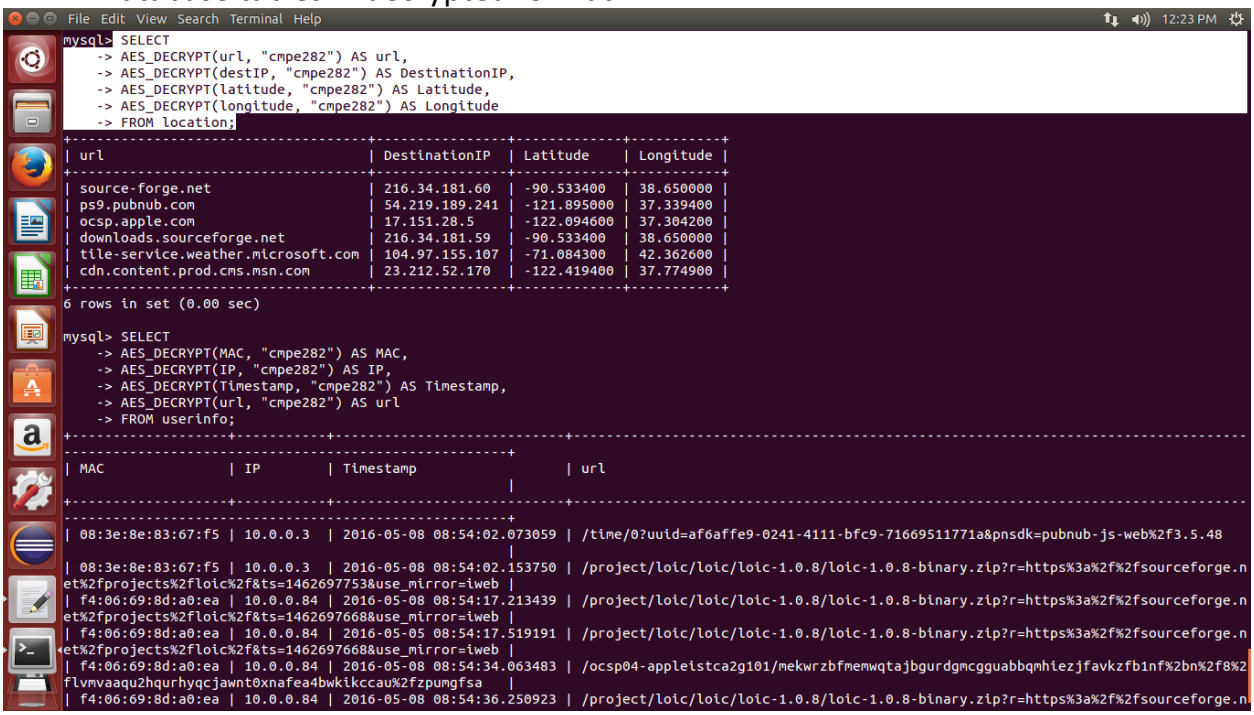

Figure 14 **SQL Queries - I**

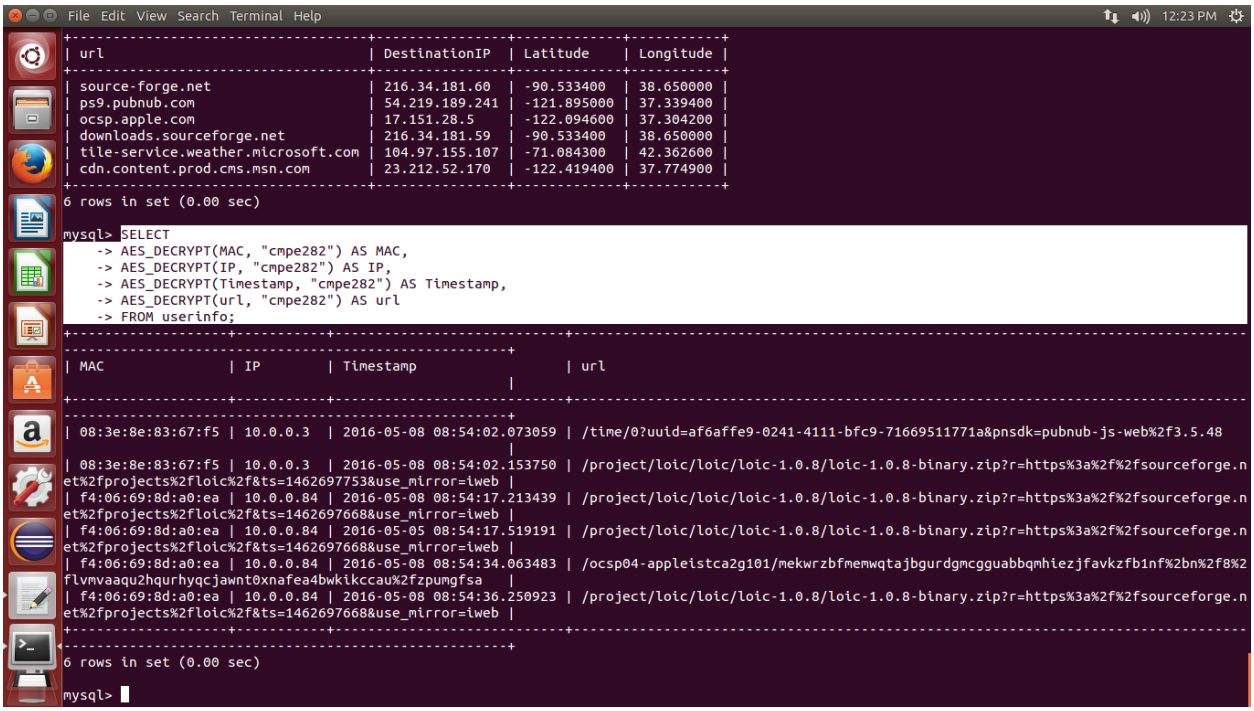

Figure 15 **SQL Queries - II**

5. APPLICATIONS

With the features that we have provided with our application in this iteration, our application is best suited for analyzing traffic in small networks where capturing packets through ARP poisoning is faster and more efficient. The system can be use in following areas

1. Labs in university.
2. Public areas like Cyber Café.
3. Small companies.

6. FUTURE WORK SUGGESTION

The project can be enhanced in future by adding few of the features as give below:

1. Develop a user friendly GUI to allow users to perform various operations in an easier and more intuitive way.
2. It can be extended to monitor remote desktops in a larger networking that are performing the illegal activities of downloading from illegal websites.
3. The information stored in database can be used for analysis purpose. The analysis would give the statistics of number of illegal activities, the IP addresses and geographical locations of the users performing them in the network. This would

help in future to predict and detect the suspicious users and help in strengthening the network security.

4. Develop a screen to capture and parse packets in real time, so that dependencies on Wireshark for pcap file will be removed.

7. CONCLUSION

With the Internet growing exponentially, providing network security has become an essential feature for many security systems, and traffic analysis is the first line of defense against security threats. It is highly important that the incoming and outgoing traffic be inspected for malicious attachments.

In our project, with the help of several tools we have implemented modules to analyze the captured traffic and retrieved important information such as the senders and receivers of the traffic, time of communication and the physical location from where the traffic is coming from.

In addition to this, we are storing this information in a MySQL database in encrypted form using AES encryption (128-bit). This will not only provide an extra layer of security, but also be useful later if we want to retrieve and analyze the information in the future.

To conclude, our application is providing the core essentials needed for traffic analysis, plus a secure way of storing the information that is being fetched through the analysis process. This application will greatly benefit small businesses, provide insight into Internet uses at homes and places like cyber café, and help owners to take preventive measures in time.

8. CONTRIBUTION

So far, following tasks are accomplished:

Gauri Bodkhe

- Study of Python Network programming and its related libraries.
- Analyzed GeoIP Database for locating the geographical location.
- Assisted Ruta in finding user IP downloading from illegal websites.
- Compiled up report and presentation.

Keertikeya Gupta

- Study of Python Network programming and its related libraries.
- Parsed all IP packets to find the geographical location using Python.
- Assisted Sanket in writing Python script for mapping IP to Google Earth.
- Setting up database to store information related to IP packets post capture.

Sanket Dhami

- Study of Python Network programming and its related libraries.
- Mapped all parsed IP packets captured through Wireshark and with Google Earth by generating KML file using Python.
- Assisted Keertikeya in setting up database for storing information related to IP packets post capture.
- End to End testing.

Ruta Dhekane

- Study of Python Network programming and its related libraries.
- Parsed the IP packets to find user IP downloading from illegal websites.
- Compiled up the report and presentation.

9. REFERENCES

- Book: violent-py(book).pdf

- https://www.techopedia.com/definition/29976/network-traffic-analysis

- https://www.google.com/search?espv=2&biw=1920&bih=919&tbm=isch&sa=1&q=wireshark+icon&oq=wireshark+icon&gs_l=img.3..0.24831.27675.0.28019.10.10.0.0.0.0.144.899.7j3.10.0....0...1c.1.64.img..1.9.786...0i10j0i7i30.jFgYQxpOHEU

- https://www.google.com/search?espv=2&biw=1920&bih=919&tbm=isch&sa=1&q=google+earth+icon&oq=google+earth+icon&gs_l=img.3..0l5j0i30l2j0i5i30l2j0i8i30.943273.950882.0.951187.19.14.1.4.4.0.112.1290.12j2.14.0....0...1c.1.64.img..0.19.1307.AjlQr_Ucwc8#imgrc=cpq49VO_LRe1xM%3A

- https://dev.mysql.com/doc/refman/5.5/en/encryption-functions.html#function_aes-encrypt

- https://pentestmag.com/ettercap-tutorial-for-windows/

- http://www.thegeekstuff.com/2012/05/ettercap-tutorial/

Appendix

Source Code:

Test1.py:

```
import dpkt
import socket
import pygeoip
import optparse
from requests import get
gi = pygeoip.GeoIP('/home/sanketdhami/Desktop/GeoLiteCity.dat')
def retGeoStr(ip):
	try:
		rec = gi.record_by_name(ip)
		city = rec['city']
		country = rec['country_code3']
		if city != '':
			geoLoc = city + ', ' + country
		else:
			geoLoc = country
		return geoLoc
	except Exception, e:
		return 'Unregistered'
def printPcap(pcap):
	for (ts, buf) in pcap:
		try:
			eth = dpkt.ethernet.Ethernet(buf)
			ip = eth.data
			src = socket.inet_ntoa(ip.src)
			src1 = src
			dst = socket.inet_ntoa(ip.dst)
			dst1 = dst
			if(src == "192.168.153.140"):
				src = get('https://api.ipify.org').text
				src= str(src)

			if(dst == "192.168.153.140"):
				dst = get('https://api.ipify.org').text
				dst= str(dst)
			print '[+] Src: ' + src1 + ' --> Dst: ' + dst1
			print '[+] Src: ' + retGeoStr(src) + ' --> Dst: '  + retGeoStr(dst)
```

```
                except:
                        pass
def main():
        parser = optparse.OptionParser(usage = 'usage: %test1.py -p test1.pcap')
        parser.add_option('-p', dest='pcapFile', type='string',help='test1.pcap')
        (options, args) = parser.parse_args()
        if options.pcapFile == None:
                print parser.usage
                exit(0)
        pcapFile = options.pcapFile
        f = open(pcapFile)
        pcap = dpkt.pcap.Reader(f)
        printPcap(pcap)
if __name__ == '__main__':
        main()
```

test2.py:

```
import dpkt
import socket
import pygeoip
import optparse
from requests import get
gi = pygeoip.GeoIP('/home/sanketdhami/Desktop/GeoLiteCity.dat')
l = []
def retKML(ip):
        ip = ip
        if ip in l:
                return
        else:
                l.append(ip)
                rec = gi.record_by_name(ip)
                try:
                        longitude = rec['longitude']
                        latitude = rec['latitude']
                        kml                                                                                     =
('<Placemark>\n''<name>%s</name>\n''<Point>\n''<coordinates>%6f,%6f</coordinates>\n''</
Point>\n''</Placemark>\n')%(ip,longitude, latitude)
                        return kml
                except:
                        return ''
```

```
def plotIPs(pcap):
	kmlPts = ''
	for (ts, buf) in pcap:
		try:
			eth = dpkt.ethernet.Ethernet(buf)
			ip = eth.data
			#src = socket.inet_ntoa(ip.src)
			#srcKML = retKML(src)
			dst = socket.inet_ntoa(ip.dst)
			dstKML = retKML(dst)
			kmlPts = kmlPts + dstKML
		except:
			pass
	return kmlPts

def main():
	parser = optparse.OptionParser(usage = 'usage: %test2.py -p test1.pcap')
	parser.add_option('-p', dest='pcapFile', type='string',help='test1.pcap')
	(options, args) = parser.parse_args()
	if options.pcapFile == None:
		print parser.usage
		exit(0)
	pcapFile = options.pcapFile
	f = open(pcapFile)
	pcap = dpkt.pcap.Reader(f)
	kmlheader = '<?xml version="1.0" encoding="UTF-8"?>\n<kml
xmlns="http://www.opengis.net/kml/2.2">\n<Document>\n'
	kmlfooter = '</Document>\n</kml>\n'
	kmldoc=kmlheader+plotIPs(pcap)+kmlfooter
	print kmldoc
if __name__ == '__main__':
	main()
```

test3.py:

```
import dpkt
import socket
import datetime
import os
import MySQLdb
import subprocess, sys
def findDownload(pcap):
```

```
for (ts, buf) in pcap:
    try:
        db=MySQLdb.connect("localhost","root","kk","cmpe282")
        cursor = db.cursor()
        eth = dpkt.ethernet.Ethernet(buf)
        ip = eth.data
        src = socket.inet_ntoa(ip.src)
        tcp = ip.data
        src_mac = ':'.join('%02x' % ord(b) for b in eth.src)
        if(src_mac == 'f4:06:69:97:33:21'):
            continue
        http = dpkt.http.Request(tcp.data)
        if http.method == 'GET':
            uri = http.uri.lower()
            if '.zip' in uri and 'loic' in uri:
                print 'Timestamp: ', str(datetime.datetime.utcfromtimestamp(ts))
                print src_mac
                print '[!] ' + src + ' Downloaded LOIC.'
                print '**************************'
                print
                sql1 = "insert into userinfo (MAC) values ('" + src_mac + "');"
                sql2 = "insert into userinfo (IP) values ('" + str(src) + "');"
                sql3 = "insert into userinfo (url) values ('" + uri + "');"
                sql4 = "insert into userinfo (timestamp) values ('"+str(datetime.datetime.utcfromtimestamp(ts))+"');"
                try:
                    cursor.execute(sql1)
                    cursor.execute(sql2)
                    cursor.execute(sql3)
                    cursor.execute(sql4)
                    print “\n\nsuccess\n\n”
                except Exception e:
                    print str(e)
            if '.exe' in uri and 'ccsetup517.exe' in uri:
                print 'Timestamp: ', str(datetime.datetime.utcfromtimestamp(ts))
                print src_mac
                print '[!] ' + src + ' Downloaded EXE file.'
                print '**************************'
                print
```

```
                                        sql1 = "insert into userinfo (MAC) values ('" + src_mac + "');"
                                        sql2 = "insert into userinfo (IP) values ('" + str(src) + "');"
                                        sql3 = "insert into userinfo (url) values ('" + uri + "');"
                                        sql4 = "insert into userinfo (timestamp) values ('"+str(datetime.datetime.utcfromtimestamp(ts))+"');"
                                        try:
                                                cursor.execute(sql1)
                                                cursor.execute(sql2)
                                                cursor.execute(sql3)
                                                cursor.execute(sql4)
                                                print “\n\nsuccess\n\n”
                                        except Exception e:
                                                print str(e)

                        except:
                                pass
while(1):
        print ('Enter number from the menu:')
        print ('1: Find culprit user')
        print ('2: Find location of destination address')
        print ('3: Find location of destination address on google earth')
        print ('4: Exit')
        x = int(raw_input())
        print
        if(x == 1):
                f = open('/home/sanketdhami/Desktop/test1.pcap')
                pcap = dpkt.pcap.Reader(f)
                findDownload(pcap)
                print
                print
'*************************************************************************
************************'


        elif(x == 2):
                os.system('python test1.py -p test1.pcap')
                print
                print
'*************************************************************************
************************'
```

```
elif(x == 3):
        os.system('python test2.py -p test1.pcap > test1.kml')
        print 'DONE. Open file using Google Earth'
elif(x == 4):
        print 'Exiting from application.....'
        exit()
else:
        print 'Exiting from application.....'
        exit()
```