HUNTDROID

(Protect your Personal Information)

By:

Muhammad Umair Nadeem

Muhammad Bilal Jafery

Rana Abdul Wahab

Muhammad Abrar Riaz

Supervised by:

AP Waleed Bin Shahid

Submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology, Islamabad,
in partial fulfillment for the requirements of B.E Degree in Electrical (Telecom) Engineering.

June 2022

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"HUNTDROID"

is carried out by

Muhammad Umair Nadeem

Muhammad Bilal Jafery

Rana Abdul Wahab

Muhammad Abrar Riaz

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical (Telecom.) Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor
AP Waleed Bin Shahid
Department of EE, MCS

Date: 23rd May 2022

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan'Wa'Tala is the sole guidance in all domains.

Our parents, colleagues, and most of all supervisor, AP Waleed Bin Shahid without your guidance.

The group members, who through all adversities worked steadfastly.

Plagiarism Certificate (Turnitin Report)

This thesis has 11% similarity index. Turnitin report endorsed by Supervisor is attached.

Muhammad Umair Nadeem
00000263107

Muhammad Bilal Jafery
00000247443

Rana Abdul Wahab
00000248572

Muhammad Abrar Riaz
00000278794

Signature of Supervisor

ABSTRACT

With the great usage of smartphones, millions of mobile Applications are being developed to provide rich functionality for users by accessing certain Personally Identifiable Information (PII), causing serious privacy concerns. Many ways to detect privacy disclosures in Android Applications have been developed to address this challenge, however they all fail to automatically decide if the privacy disclosures are occurring through an Application or not. As a result, while using such methodologies for Application analysis, security analysts may encounter a lot of false positives.

HUNTDROID is an analysis system, comprised of an Android Application, that tries to automatically identify privacy breaches by determining whether an Android Application is leaking user's personal information or not. Any Android Application, if shares personal information of the user, falls under the category of privacy disclosure. Given that most mobile Applications are really web Applications, and most of them communicate in an unsecured fashion, such information disclosure is significant and can be pervasive.

Table of Contents

List of Figures

Chapter 1: Introduction

Engineering is a new discipline of science focused on professional abilities based on scientific experimental knowledge and the science of numbers. In many aspects, this discipline of study is tremendously vital to society and its constituents, particularly developmental engineering research that is raising humanity's standard of living.

Engineering encompasses everything from visual constructs on software to on-site actual work, and it is not confined to research topics. Not only that, but this branch of research also establishes safety policies and practices. Engineers that follow engineering principles use their domain knowledge to design and build products that address societal issues. It could be a problem with privacy, health, food, transportation, astrology, financial, environment, or entertainment.

With the advancement in engineering knowledge, research in information security has also increased up to a great extent. Information security refers to a collection of procedures for securing data from unauthorized access or modification while it is being stored as well as when it is being transferred from one device or geographic area to another. It's also called as data security. Information security has drastically changed the research area.

Nowadays, the main concern of Android Users is privacy that needs to be dealt with on priority basis.

1.1 Definition of PII

Personally Identifiable Information refers to any information of a person or an individual that can be helpful in identifying him or her. Personally Identifiable Information is most referred shortly as “PII”. PII can be thought as a unique identifier of a person and includes a lot of sensitive/private information about the person. PII includes name, phone number, email address, CNIC No., location, date of birth, religion, credit card no's, images, and a lot of other personal information.

1.2 Overview

The tech world is changing day by day. Many people are shifting towards smartphones. Android has been the most used operating system. There are almost 2.5 billion active users of Android Phones.

Android Applications have been developed keeping in mind the need of the Android users. Android Users get attracted to these applications and install them on their phones without considering the potential threat to their privacy.

Personally Identifiable Information (PII) is an important aspect of an individual's privacy. Personally Identifiable Information includes name, CNIC, mobile number, credit card number, country, images, location and much more. These needs to be protected from being leaked.

And here comes the "HUNTDROID", it detects the privacy disclosures done by the Android Applications and prompt the user to take the necessary action.

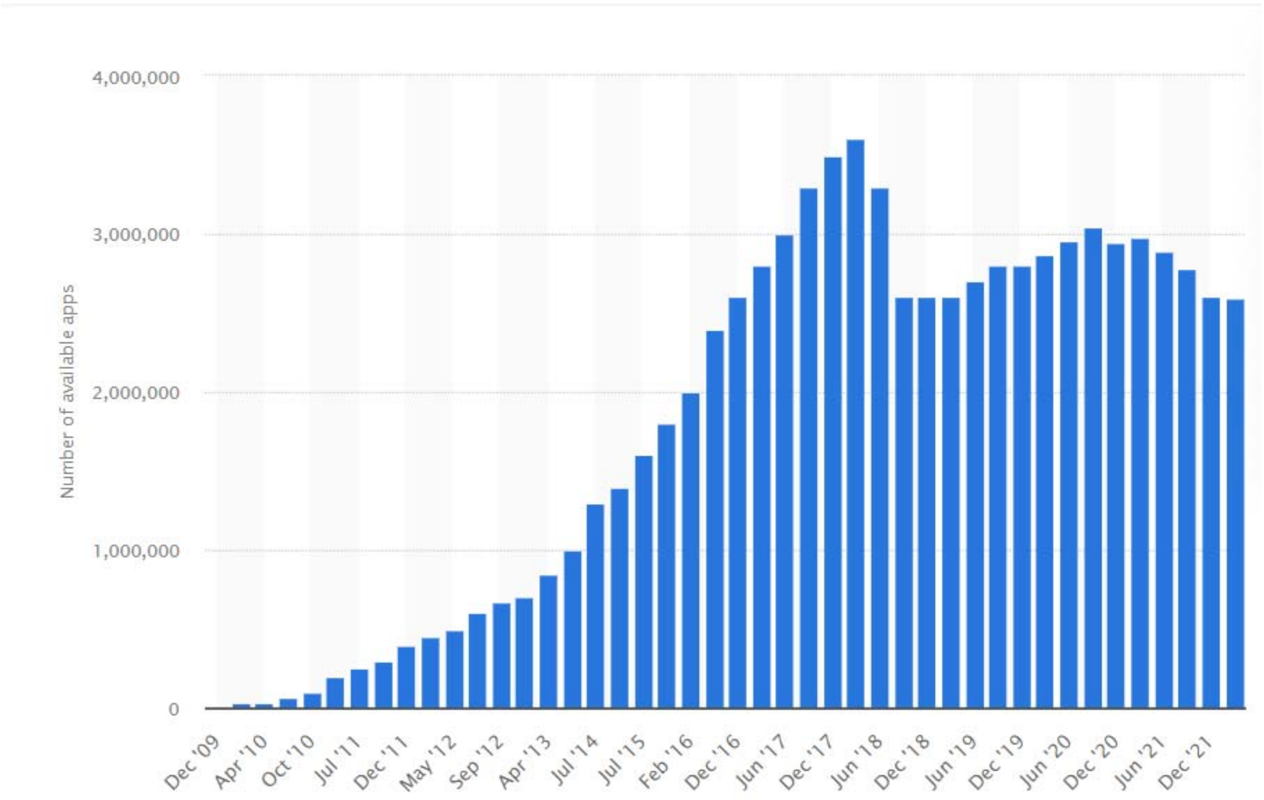


Figure 1: Google Play store Android Applications till March 2022

Above shows the detailed statistics of the Android Applications available on Google Play store. There are 2.5 million Android Applications for the users, some of them are too good and concerned with the privacy of the users. But there are many Applications that does not care the user privacy and users' data is being disclosed with unauthorized parties.

1.3 Problem Statement

Android Phones continuously keep the record of the user's personal information like user's location, audiovisual environment, and day to day activity. They also store passwords, banking details, schedules, and other sensitive data. The main procedure adopted for the protection of user's privacy is an audit, which is primarily done by the platform provider. This audit ensures that the malware is removed from the Application store.

There are mainly three types of privacy disclosures done by Android Applications i.e., Incidental privacy leaks, Accidental privacy leaks, and Malicious leaks.

Incidental privacy leaks occur when the Android application's operation is compromised. For example, an application gets the user's current location as a query from database and leaks that information to the provider's site as a part of each query.

Accidental privacy leaks are the result of Application vulnerabilities and bugs in it. These are source of disclosure of the user's private information to attackers or other applications.

Malicious leaks are the leak when an application shares the user's private information without the consent of the user to third parties.

All these privacy leaks can occur when the application requests for the private information and access is granted but that was not required for the functionality of the application.

1.4 Proposed Solution

A lot of existing solutions are serving the purpose of identifying data leaks via android apps but most of the solutions are commercial in nature. Open-source solutions are also available, but they provide limited functionality.

Our project "**HUNTDROID**" aims to identify, analyze, and prevent users' private information via Android Applications. Our project will try to identify and cure all types of data leaks via Android applications including **Incidental, Accidental,** and **Malicious** leaks.

1.5 Working Principle

The project mainly works on the principles of detection of PII Leaks using Network Analysis and then comparing the results with the PII RegEx. PII RegEx are compared with the network traffic to check if the privacy disclosure being done by an Android Application or not.

The project is divided into different modulus and every module is inter-woven with the next module. The list of modules is as under:

- Network Traffic Analysis
- PII RegEx
- Real Time Analysis of Android applications
- Results
- Android application

1.5.1 Network Traffic Analysis

An integral part of the project is the network analysis of Android Applications. In test phase, we did traffic analysis from multiple tools such as Wireshark, Burp suite, and Fiddler. A PCAP file is being analyzed after the Network Traffic Analysis is completed.

1.5.2 PII RegEx

PII RegEx, are the general regular expressions which are used to compare the network traffic and analyze whether the RegEx matches with the traffic being transmitted by the Android Application or not.

PII RegEx have been designed for Mobile Number, CNIC, Location, Credit Card Numbers, and many more.

1.5.3 Real Time Analysis of Android Applications

We have tested 200 Android Applications installed from Google Play store and analysis have been done to detect privacy disclosures.

Procedure to test the Android Application was to register on the application for the very first time to see whether the personal data is being leaked by the Application or not.

Result of the Analysis is discussed in the Result section.

1.5.4 Results

We have compiled our Android Application’s Analysis in the form of PIE Chart. The results are shown below:

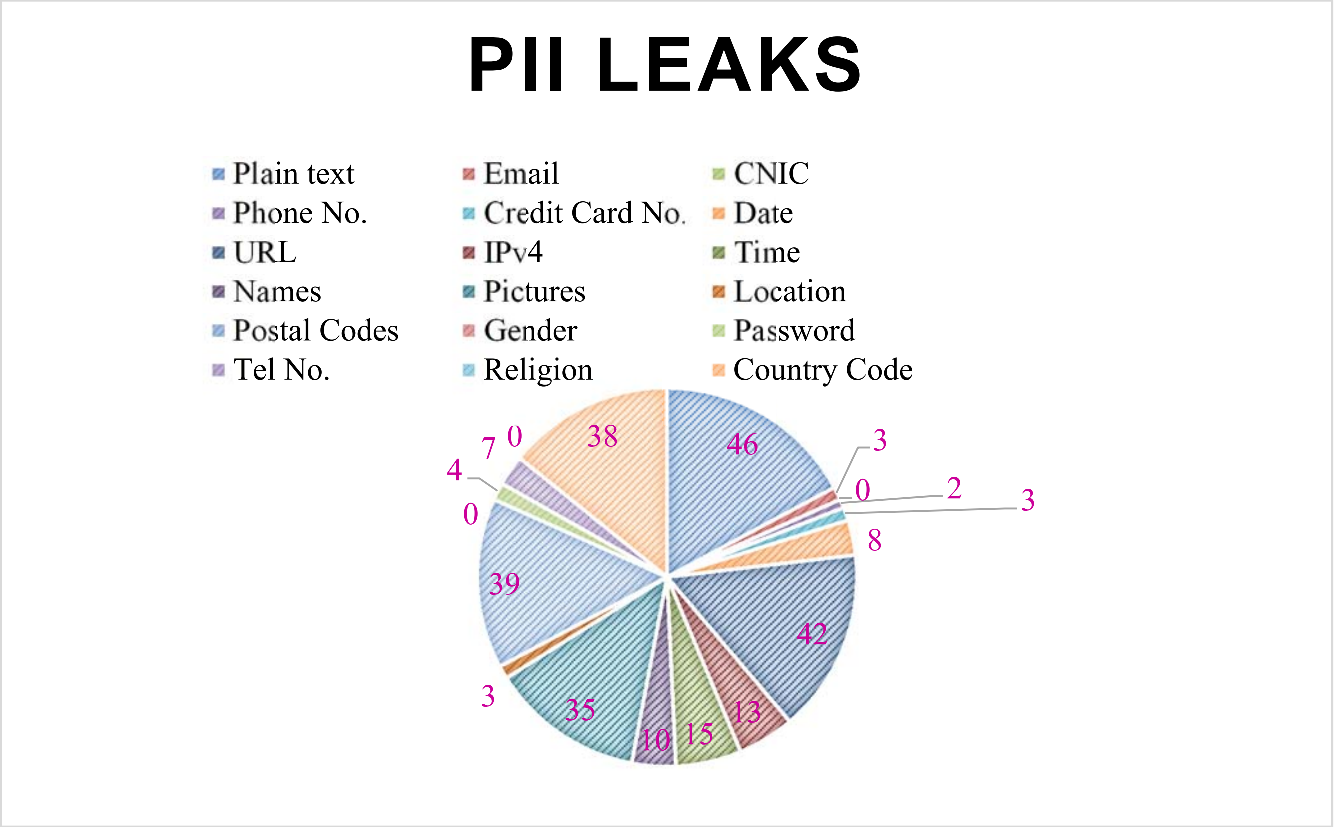


Figure 2: Results

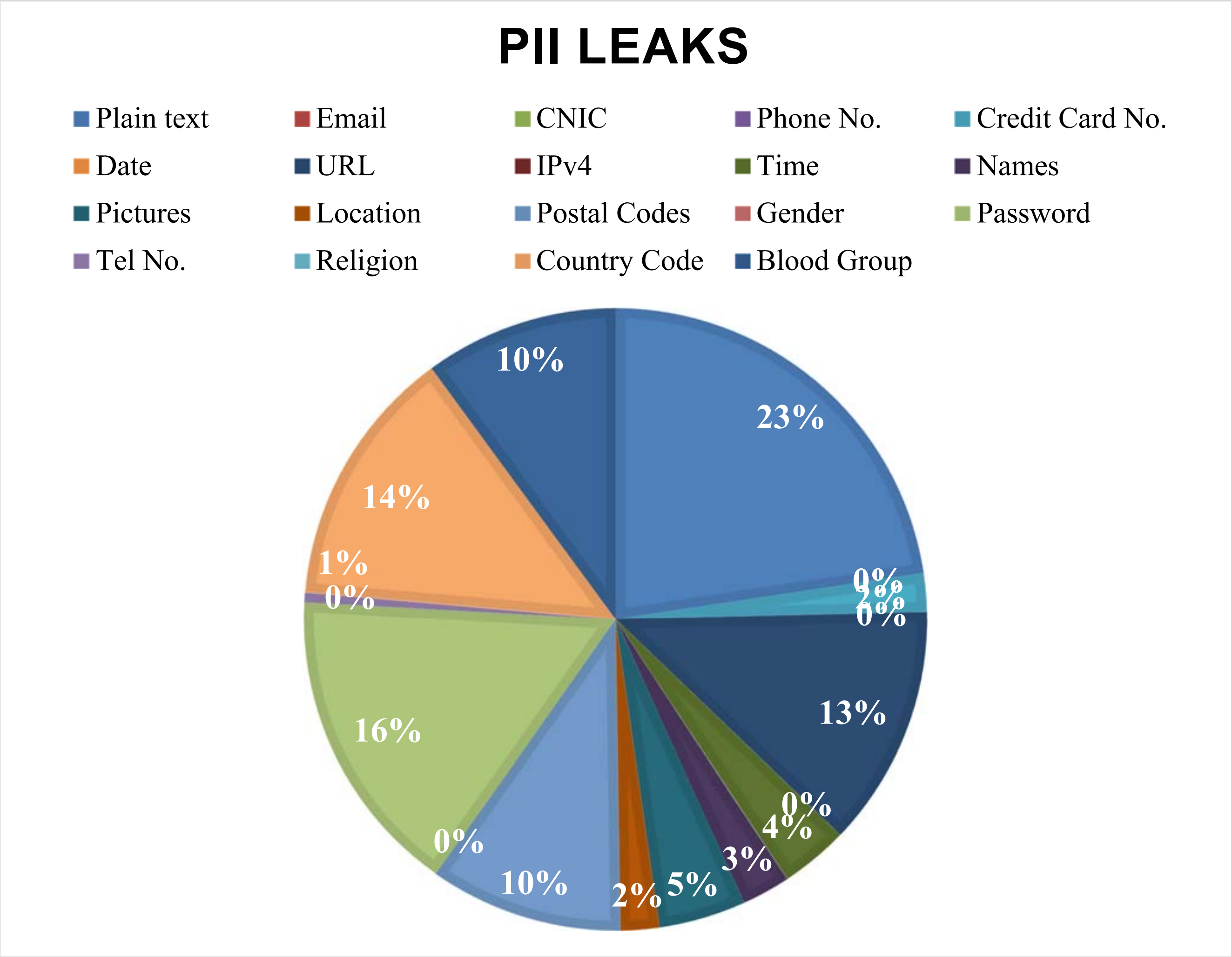


Figure 3: Results

1.5.5 Android Application

All the functionalities of the "HUNTDROID" have been integrated to develop an Android Application. Users can install it on their Android phones for the detection of privacy disclosures done by the Android Applications.

1.5.6 Block Diagram

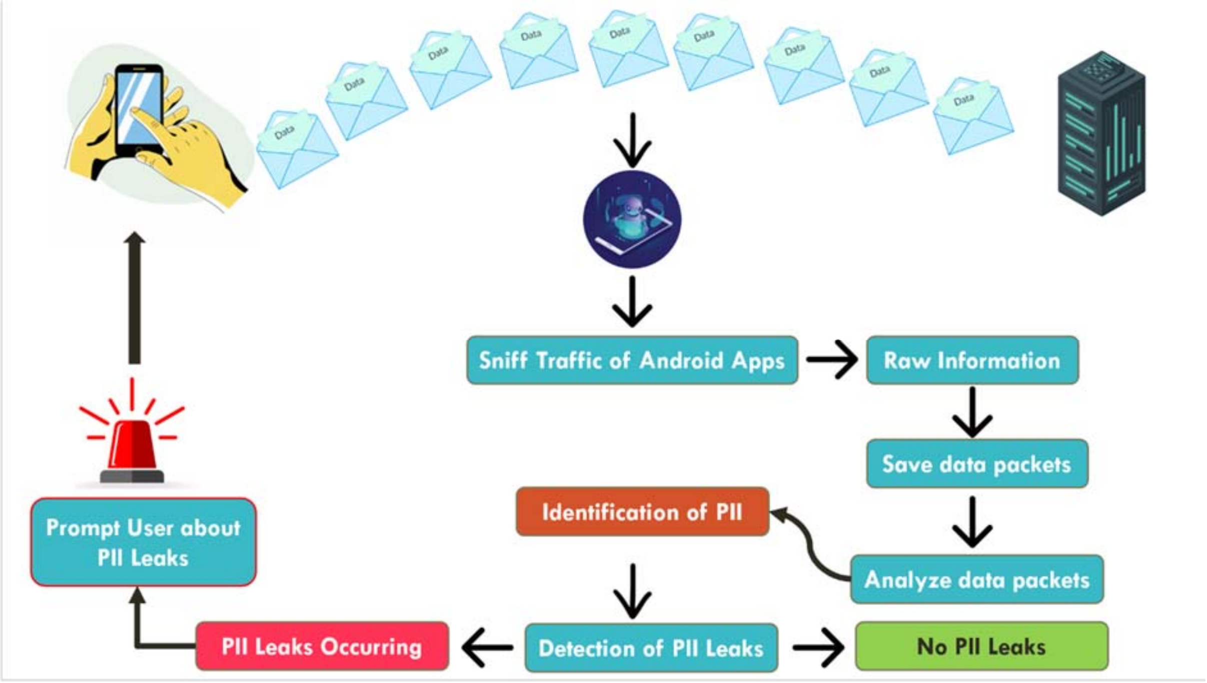


Figure 4: Block Diagram

1.6 Objectives

1.6.1 General Objectives:

“To develop an innovative state of the art Android Application that will accurately detect privacy leaks done by Android Applications, providing an efficient way to Android Users to protect their personally identifiable information.”

1.6.2 Academic Objectives:

The educational objectives of the HUNTDROID include knowledge of:

- ✓ Entire Android Development lifecycle
- ✓ Network Analysis
- ✓ Monitoring Android applications
- ✓ Identifying Personally Identifiable Information
- ✓ Identify and monitor PII Leaks

1.7 Scope

The development of our project “HUNTDROID” requires sound knowledge of:

- ✓ Network Analysis
- ✓ Data Breach in Android applications
- ✓ Android Application development cycle

To prevent the data leaks via Android Applications, we should have knowledge of:

- ▪ Android Application Development
- ▪ Programming language, Java

1.8 Deliverables

1.8.1 Android Application

The final deliverable is an Android Application which can be installed by the Android Users on their smartphones. It will be used as the tool for them to detect privacy disclosure done by the Android Application.

1.8.2 Object of interest:

It can detect the object of interest by real time analysis of Android Applications. By detecting the object of interest means detection of Personally Identifiable Information disclosures using "PII RegEx Technique".

1.8.3 Special privileges:

It provides the special privileges to the Android Users by prompting them about the privacy disclosures done by the Android Applications. The users will be then independent to take any action based on the results, i.e., whether the user wants to keep using the Android Application or want to block some irrelevant permissions.

1.9 Relevant Sustainable Development Goals

Our project "HUNTDROID" aligns with the Sustainable Development Goal 9 i.e., Industry, Innovation, and Infrastructure. The privacy concerns have been increasing day by day with more Android Application coming in the business. These Android Applications are a risk to the Personal Privacy of the individuals as well as the businesses.

HUNTDROID aims to deal with these privacy concerns by detecting the privacy disclosures done by the Android Applications. These Applications discloses personal data of the user with 3rd party that could be used for marketing purposes as well as for criminal activity against an individual or business.

1.10 Structure of Thesis

Chapter 2 contains the literature review and the background and analysis study this thesis is based upon.

Chapter 3 contains the goals and design of the HUNTDROID.

Chapter 4 contains the implementation of the project.

Chapter 5 contains the conclusion of the project.

Chapter 6 highlights the future work needed to be done for the commercialization of this project.

Chapter 2: Literature Review

A new product is launched by modifying and enhancing the features of previously launched similar products. A literature review is an important step in the development of an idea for a new product. Likewise, for the development of a product, and its replacement, related to privacy leaks, a detailed study regarding all similar projects is compulsory. Our research is divided into the following points.

- Industrial Background
- Existing solutions and drawbacks
- Research

2.1 Industrial background

As we all know that millions of mobile phones are used nowadays, and billions of mobile applications are developed which provide different functionalities to the users by letting them access their personal information which leads to privacy concerns. To bring this issue down, several solutions have been suggested to get to know the privacy leaks in mobile applications, but these solutions are widely failing to determine whether these privacy leaks are necessary or not for any functionality of the applications. And in a result of adopting any of these solutions, the analysts got the false information of application analysis.

Personal mobile phones are continuously monitored and kept a record of the location, distinct environment, and activity of their users, and are also used for storing critical information like passwords, schedules, banking information, and other users' sensitive information. The basic process for making sure of the user's security and privacy is the inspection by the service provider, which generally makes sure that obvious malware is tagged from the application business.

The resources third-party applications can access are controlled by the mobile platforms which include iOS, Android, and Windows phones. When an application is being installed on any Mobile phone for the first time the users are asked to give permission grants to the resources like camera, location, contacts, networks, microphones, etc. Previous research has shown that the users have to approve so many requests for the use of functionalities, as they are not able to make any decision about whether the application has any need of that resource or not and they are also not aware of the risks that will occur by given the access.

Unfortunately, these leaks by the applications are common nowadays and come out in the following ways:

- Incidental
- Accidental
- Malicious

2.1.1 Incidental

These are the privacy leaks that occur as the side effect of any operation held by the application. For example, an application that inquiries about the database on the service provider's site for information about the user's interest and information about the location of the user to the provider along with each query.

2.1.2 Accidental

These are the privacy leaks that occur because of application bugs and the liabilities that leak the personal information of the user to attackers, network eavesdroppers, or other applications.

2.1.3 Malicious

These are the privacy leaks that occur when an application needlessly and deliberately forwards the user's personal data to unauthorized parties.

These types of leaks are enhanced when an application requires access to private data even though it is not required for its functionality.

Numerous solutions are available in the market for identifying data leaks by the android applications, but most of them are commercial. Some open-source solutions are also available, but they provide limited functionality.

2.2 Existing solutions and Drawbacks

Different solutions are previously being provided for the personal information leaks (PII leaks) problem. Following are some solutions that are already being prepared and being implemented.

- Privacy Capsules
- LeakDr
- ReCon
- ClueFinder

2.2.1 Privacy Capsules

In Privacy Capsules, the mobile applications are executed in two forms i.e., the sealed form and the unsealed form. These are implemented by mobile phones, also these are independent of language, and it can be done by doing some changes to the application. Using a PC implementation prototype in Android, PCs have energy overhead and low performance, and these are suitable for a bigger class of applications.

2.2.2 LeakDr

LeakDoctor aims to automatically detect each privacy disclosure and diagnose its necessity for app functionality. LeakDoctor is designed from both dynamic analysis and static analysis perspectives:

2.2.2.1 Dynamic analysis perspective

LeakDoctor continuously modifies the value of the leaked private information at runtime and monitors how these changes impact remote responses to see whether they change the application functionality.

2.2.2.2 Static analysis perspective

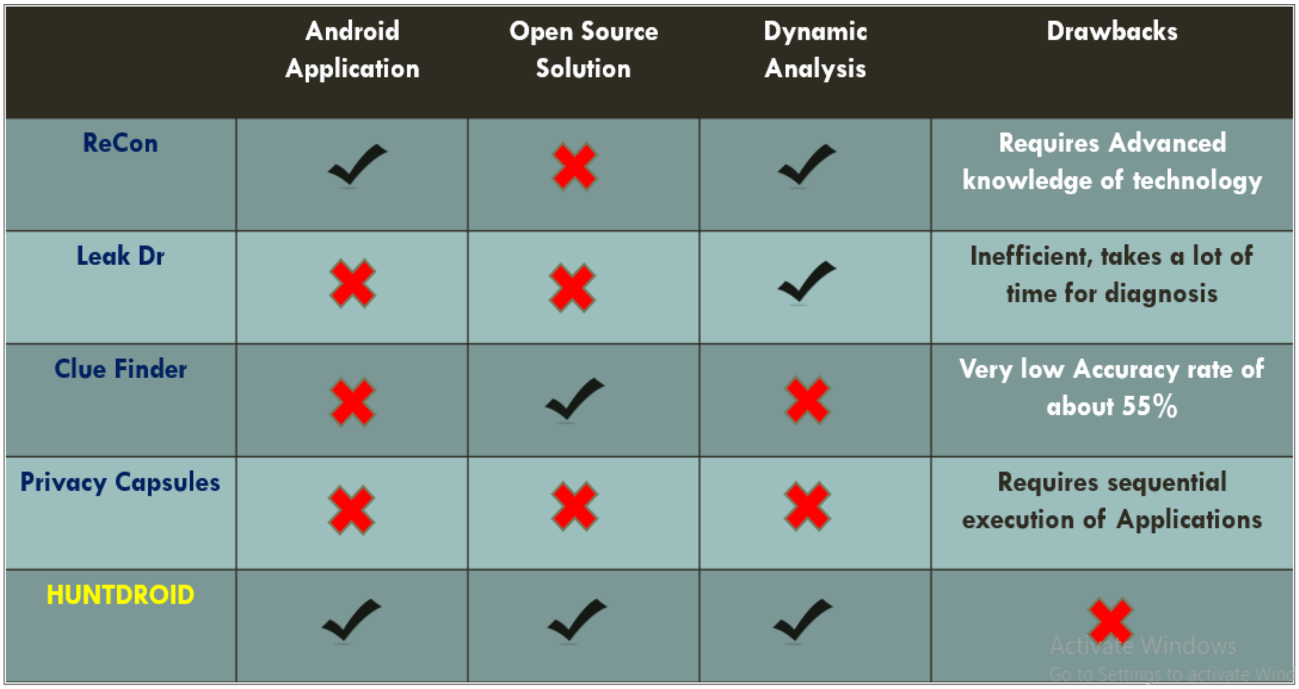
LeakDoctor aims to correlate every privacy disclosure to the functionality of a specific application (i.e., SPS-related APIs) to see whether it is useful or not.

2.2.3 ReCon

ReCon uses machine learning to detect the PII leaks by inspecting the network traffic of the system and provides users the tool by which they get the ability to control these PII leaks by substitution and blocking of PII. But this is not an open-source solution and provides limited functionality.

2.2.4 ClueFinder

ClueFinder firstly utilizes the set of keywords, unique acronyms, and prefixes that represents the numerous types of user-sensitive personal information to recognize the program components that might involve sensitive content such as passwords and schedules. Then the elements which are classified are inspected through the mechanism known as NLP (Natural Language Processing) and remove those which do not represent any information or sensitive content.



	Android Application	Open Source Solution	Dynamic Analysis	Drawbacks
ReCon	✓	✖	✓	Requires Advanced knowledge of technology
Leak Dr	✖	✖	✓	Inefficient, takes a lot of time for diagnosis
Clue Finder	✖	✓	✖	Very low Accuracy rate of about 55%
Privacy Capsules	✖	✖	✖	Requires sequential execution of Applications
HUNTDROID	✓	✓	✓	✖

Figure 5: Literature Review

2.3 Hardware Overview

The project consists of the following hardware components which are described below.

2.3.1 Raspberry pi

Raspberry pi is a microprocessor board, and it requires the operating system to run. it consists of a high-speed processor, 4GB memory, and connectivity. It uses an ethernet adapter for internet connectivity. The project is software based so this is needed to show the working of the project in hard form. The project is implemented on the board and shows the result.

Figure 6: RaspberryPi

2.3.2 Android Device

Android device is a device, and it requires the operating system to run. It consists of a high-speed processor, 64GB memory, and connectivity. It also requires an internet connection. The result of the project is an application it runs on the android device.

Figure 7: Android Device

2.3.3 Battery/ Power bank

The battery is the device that provides the power to any operating power system. Batteries are of different powers. i.e., 9V, 12V, etc., and power banks are the bunches of batteries or cells connected and used for the higher power operating systems.

These are needed for the power support to the android devices and the raspberry pi module.

Figure 8: Power Bank

2.4 Software Overview

The project requires the following software related tasks:

2.4.1 PyCharm

Python is a high-level language that is easy to learn and use and has a simple syntax. It is used to develop GUI, web applications, and websites. It makes programming easier with the wide variety of built-in libraries that it provides. PyCharm is an integrated environment used for python. For this project, we used PyCharm version 3.3. In HUNTDROID it runs the code to detect the PII leaks in the traffic captured.

2.4.2 Android Studio

Android studio is the software that is used for different purposes. Android studio is used to build applications for android phones and many other devices. The structured modules are used in android studio and allow you to divide your projects into different units of functionalities.

2.4.3 Java IDE

Java IDE is the software application that is used to debug or write Java programs more easily. Although every IDE has the same feature of syntax highlighting and code completion that helps the user to run their codes easily.

Chapter 3: HUNTDROID Goals and Design

3.1 Identification of Personally Identifiable Information

There's no doubt that the Android market is rapidly expanding, with 2.9 million apps in the Google Play store. It has provided us with several economic opportunities. However, it has frightened us about the data security issue. Because the Play Store does not closely check applications, as the iPhone App Store does, there is a greater risk of malicious Android applications being approved.

People use applications for nearly everything these days, including banking, shopping, and booking, therefore it's up to the developers to keep an eye on security. After that, create a user-friendly and secure experience.

The first step in our solution involves system analysis and analysis of network traffic. The next step would be the identification of PII while monitoring the Android applications. After the identification of PII, our analyzer will detect PII leaks occurring in the Android applications, and it will prompt users about the data breach and take necessary actions to stop that breach.

3.2 Non-Goals

HUNTDROID is not meant to be a complete replacement for existing mobile device privacy solutions. If an Android application is communicating in a secured fashion and shares encrypted information, HUNTDROID will not be able to classify it as a privacy disclosure. Importantly, HUNTDROID enables us to detect and prevent unencrypted PII in network flows from any device without the need for OS changes or taint tracking.

3.3 Deployment and Adoption

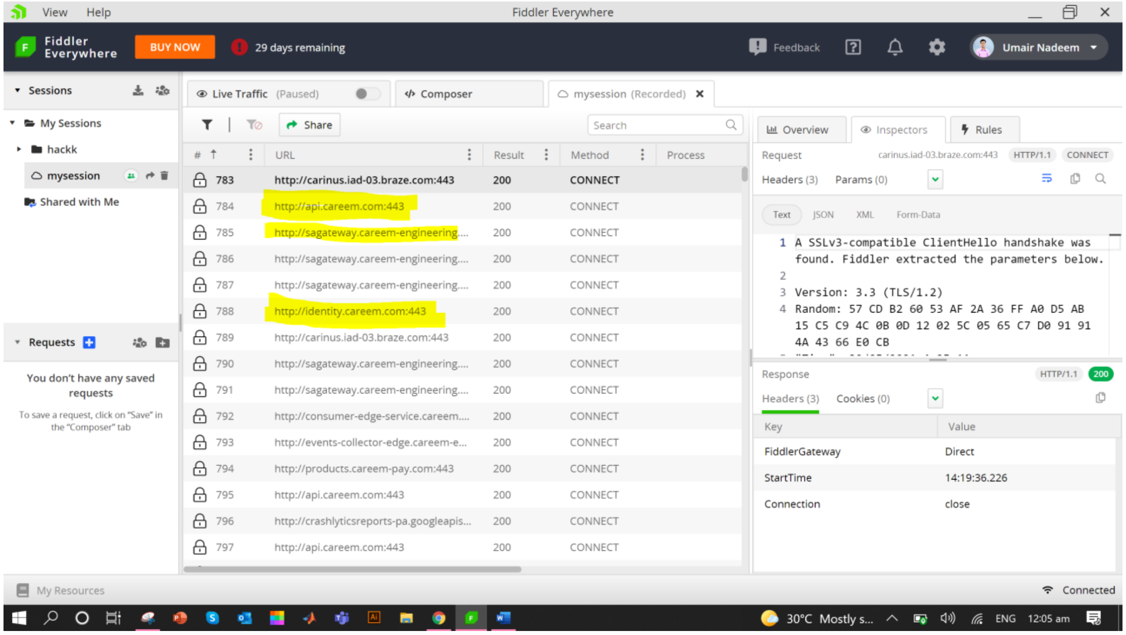
HUNTDROID accepts a range of deployment types, such as in the cloud, on gadgets, or on smart phones, because it only needs access to network traffic to discover and identify PII leaks. This service is now hosted via an Android application since it gives immediate access with minimum overhead.

Chapter 4: Experimentation

This chapter includes all the experimentation that we did before starting our project. This experimentation provided us a roadmap to analyze and understand the network traffic flows. It also proved to be a landmark for boosting our interest and enthusiasm.

4.1 Traffic Analysis Using Fiddler

We analyzed the traffic of multiple Android applications through Fiddler, which is an opensource platform for traffic Analysis. Below are the results shown, displaying traffic of Android applications captured by Fiddler:



You don't have any saved requests

To save a request, click on "Save" in the "Composer" tab

#	URL	Result	Method	Process
783	http://carinus.iad-03.braze.com:443	200	CONNECT	
784	http://api.careem.com:443	200	CONNECT	
785	http://sagateway.careem-engineering....	200	CONNECT	
786	http://sagateway.careem-engineering....	200	CONNECT	
787	http://sagateway.careem-engineering....	200	CONNECT	
788	http://identity.careem.com:443	200	CONNECT	
789	http://carinus.iad-03.braze.com:443	200	CONNECT	
790	http://sagateway.careem-engineering....	200	CONNECT	
791	http://sagateway.careem-engineering....	200	CONNECT	
792	http://consumer-edge-service.careem....	200	CONNECT	
793	http://events-collector-edge.careem-e...	200	CONNECT	
794	http://products.careem-pay.com:443	200	CONNECT	
795	http://api.careem.com:443	200	CONNECT	
796	http://crashlyticsreports-pa.googleapis...	200	CONNECT	
797	http://api.careem.com:443	200	CONNECT	

Request carinus.iad-03.braze.com:443 HTTP/1.1 CONNECT

Headers (3) Params (0)

Text JSON XML Form-Data

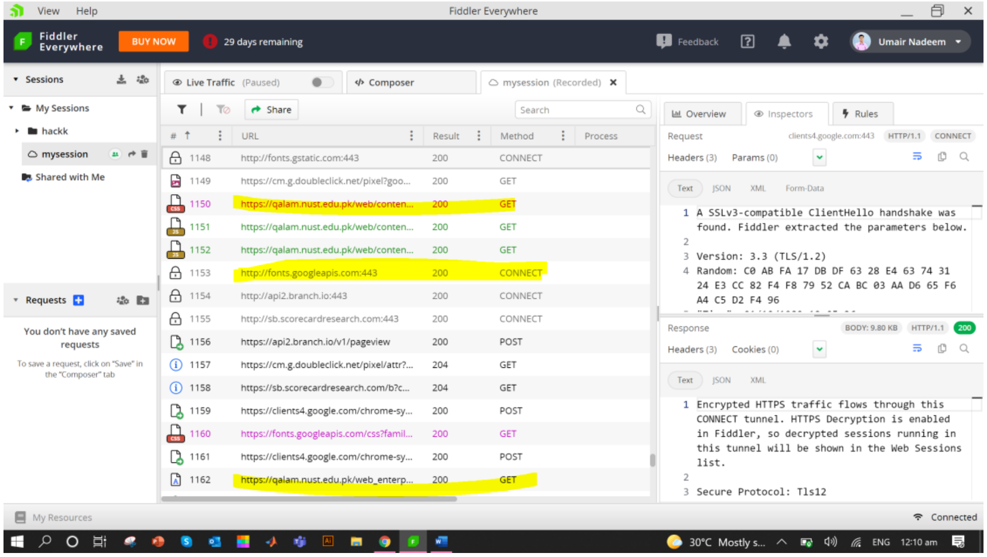
```
A SSLv3-compatible ClientHello handshake was
found. Fiddler extracted the parameters below.

Version: 3.3 (TLS/1.2)
Random: 57 CD B2 60 53 AF 2A 36 FF A0 D5 AB
15 C5 C9 4C 0B 0D 12 02 5C 05 65 C7 D0 91 91
4A 43 66 E0 CB
```

Response HTTP/1.1 200

Headers (3) Cookies (0)

Key	Value
FiddlerGateway	Direct
StartTime	14:19:36.226
Connection	close

Figure 9: Traffic Analysis Using Fiddler



#	URL	Result	Method	Process
1148	http://fonts.gstatic.com:443	200	CONNECT	
1149	https://cm.g.doubleclick.net/pixel?goo...	200	GET	
1150	https://qalam.nust.edu.pk/web/conten...	200	GET	
1151	https://qalam.nust.edu.pk/web/conten...	200	GET	
1152	https://qalam.nust.edu.pk/web/conten...	200	GET	
1153	http://fonts.googleapis.com:443	200	CONNECT	
1154	http://api2.branch.io:443	200	CONNECT	
1155	http://sb.scorecardresearch.com:443	200	CONNECT	
1156	https://api2.branch.io/v1/pageview	200	POST	
1157	https://cm.g.doubleclick.net/pixel/attr?...	204	GET	
1158	https://sb.scorecardresearch.com/b?c...	204	GET	
1159	https://clients4.google.com/chrome-sy...	200	POST	
1160	https://fonts.googleapis.com/css?famil...	200	GET	
1161	https://clients4.google.com/chrome-sy...	200	POST	
1162	https://qalam.nust.edu.pk/web_enterp...	200	GET	

Request clients4.google.com:443 HTTP/1.1 CONNECT

Headers (3) Params (0)

Text JSON XML Form-Data

```
A SSLv3-compatible ClientHello handshake was found. Fiddler extracted the parameters below.

Version: 3.3 (TLS/1.2)
Random: C0 AB FA 17 DB DF 63 28 E4 63 74 31 24 E3 CC 82 F4 F8 79 52 CA BC 03 AA D6 65 F6 A4 C5 D2 F4 96
```

Response BODY: 9.80 KB HTTP/1.1 200

Headers (3) Cookies (0)

Text JSON XML

```
Encrypted HTTPS traffic flows through this CONNECT tunnel. HTTPS Decryption is enabled in Fiddler, so decrypted sessions running in this tunnel will be shown in the Web Sessions list.

Secure Protocol: Tls12
```

Figure 10: Traffic Analysis using Fiddler

4.2 Development of a Client-Server Application

We developed two Android applications, a client side and a server side. The client side was sending data to server in an unsecured manner. So, we included some PII at the client side and sent it to the server. We captured this communication and used it as a test case for our project.

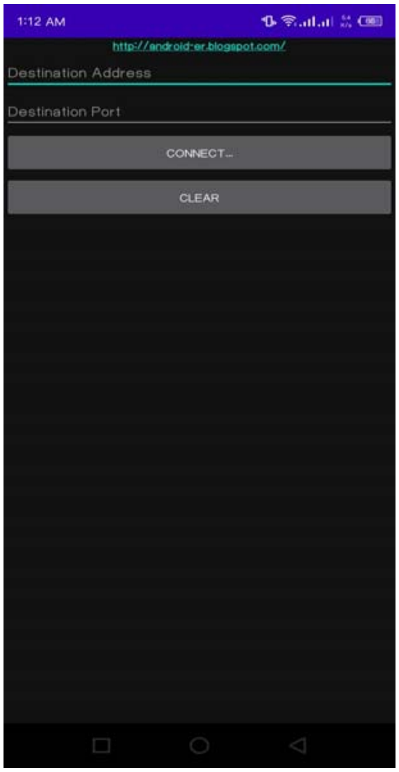
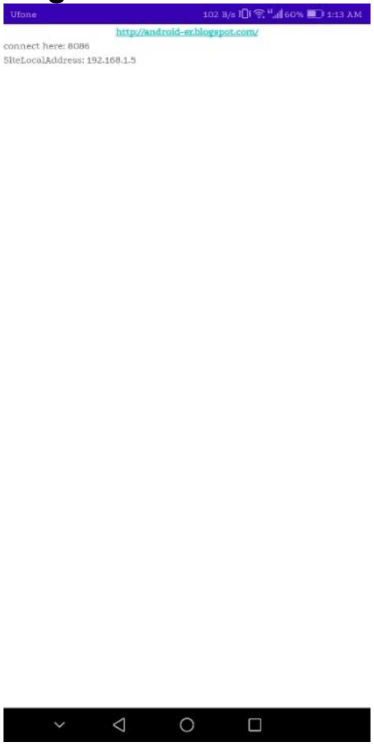
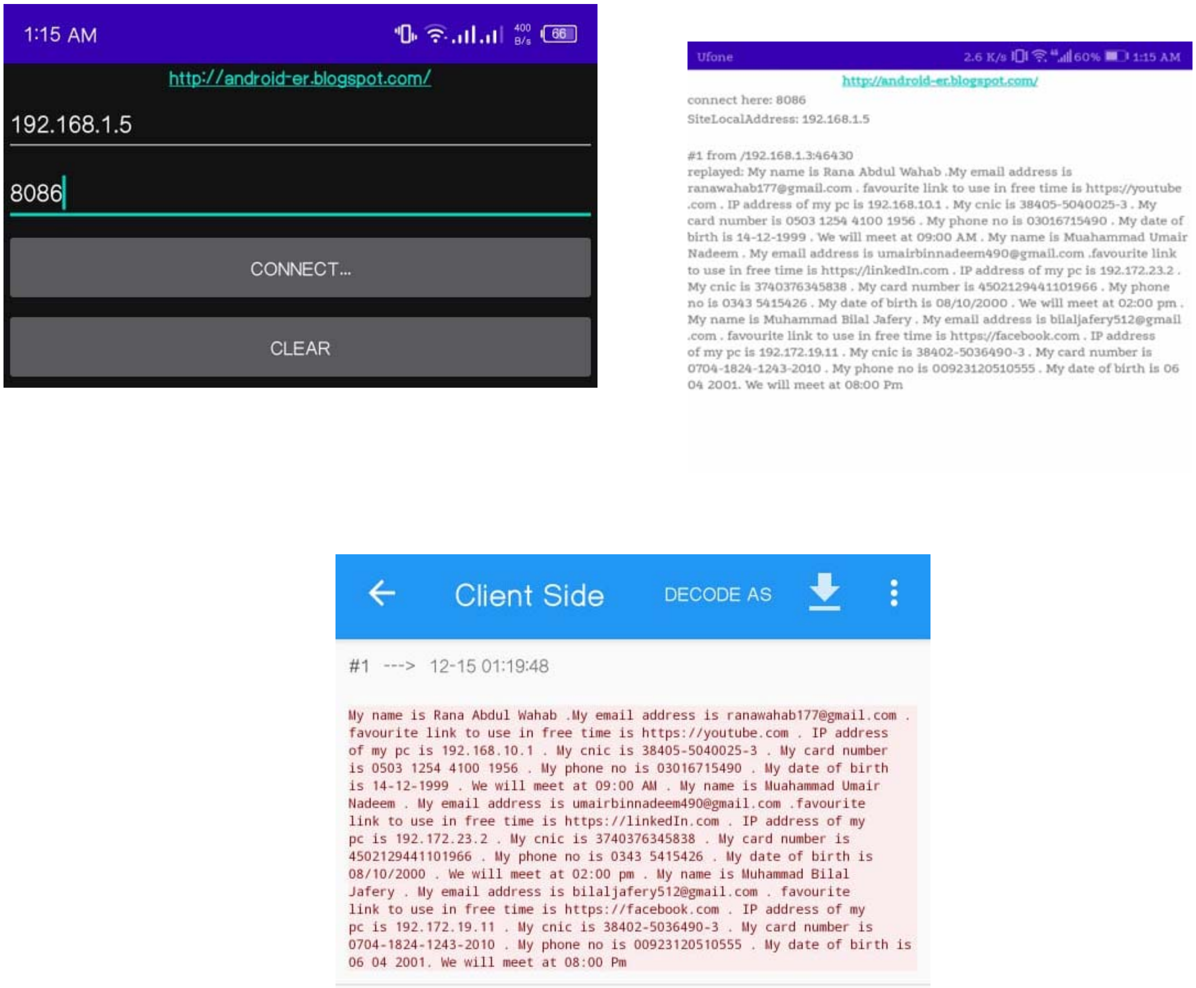


Figure 11: Server Side



```
connect here: 8086
SiteLocalAddress: 192.168.1.5
```

Figure 12: Client Side



```
192.168.1.5
8086
```

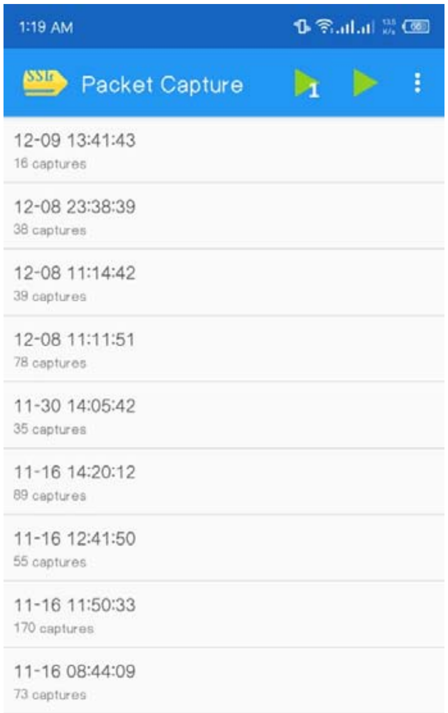
```
http://android-er.blogspot.com/
connect here: 8086
SiteLocalAddress: 192.168.1.5

#1 from /192.168.1.3:46430
replayed: My name is Rana Abdul Wahab .My email address is
ranawahab177@gmail.com . favourite link to use in free time is https://youtube
.com . IP address of my pc is 192.168.10.1 . My cnic is 38405-5040025-3 . My
card number is 0503 1254 4100 1956 . My phone no is 03016715490 . My date of
birth is 14-12-1999 . We will meet at 09:00 AM . My name is Muahammad Umair
Nadeem . My email address is umairbinnadeem490@gmail.com .favourite link
to use in free time is https://linkedIn.com . IP address of my pc is 192.172.23.2 .
My cnic is 3740376345838 . My card number is 4502129441101966 . My phone
no is 0343 5415426 . My date of birth is 08/10/2000 . We will meet at 02:00 pm .
My name is Muhammad Bilal Jafery . My email address is bilaljafery512@gmail
.com . favourite link to use in free time is https://facebook.com . IP address
of my pc is 192.172.19.11 . My cnic is 38402-5036490-3 . My card number is
0704-1824-1243-2010 . My phone no is 00923120510555 . My date of birth is 06
04 2001. We will meet at 08:00 Pm
```

```
#1  --->  12-15 01:19:48

My name is Rana Abdul Wahab .My email address is ranawahab177@gmail.com .
favourite link to use in free time is https://youtube.com . IP address
of my pc is 192.168.10.1 . My cnic is 38405-5040025-3 . My card number
is 0503 1254 4100 1956 . My phone no is 03016715490 . My date of birth
is 14-12-1999 . We will meet at 09:00 AM . My name is Muahammad Umair
Nadeem . My email address is umairbinnadeem490@gmail.com .favourite
link to use in free time is https://linkedIn.com . IP address of my
pc is 192.172.23.2 . My cnic is 3740376345838 . My card number is
4502129441101966 . My phone no is 0343 5415426 . My date of birth is
08/10/2000 . We will meet at 02:00 pm . My name is Muhammad Bilal
Jafery . My email address is bilaljafery512@gmail.com . favourite
link to use in free time is https://facebook.com . IP address of my
pc is 192.172.19.11 . My cnic is 38402-5036490-3 . My card number is
0704-1824-1243-2010 . My phone no is 00923120510555 . My date of birth is
06 04 2001. We will meet at 08:00 Pm
```

Figure 13: Captured packet

4.3 Analysis using Opensource applications

We also did analysis of Android applications using opensource applications, available on Google PlayStore (PacketCapture, NetCapture) to see the behaviour of Android applications and predict the nature of encryption on these Android applications.



12-09 13:41:43
16 captures

12-08 23:38:39
38 captures

12-08 11:14:42
39 captures

12-08 11:11:51
78 captures

11-30 14:05:42
35 captures

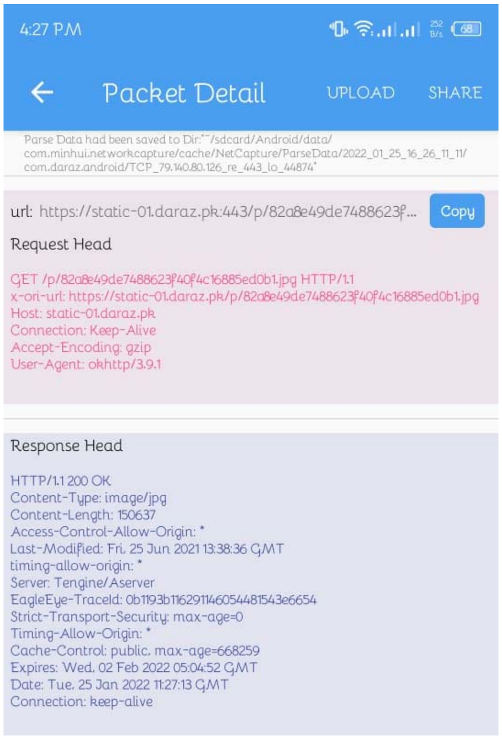
11-16 14:20:12
89 captures

11-16 12:41:50
55 captures

11-16 11:50:33
170 captures

11-16 08:44:09
73 captures

Figure 14: Analysis using PacketCapture



Parse Data had been saved to Dir:"~/sdcard/Android/data/com.minhui.networkcapture/cache/NetCapture/ParseData/2022_01_25_16_26_11_11/com.daraz.android/TCP_79.140.80.126_re_443_lo_44874"

url: https://static-01.daraz.pk:443/p/82a8e49de7488623f...

Request Head

```
GET /p/82a8e49de7488623f40f4c16885ed0b1.jpg HTTP/1.1
x-ori-url: https://static-01.daraz.pk/p/82a8e49de7488623f40f4c16885ed0b1.jpg
Host: static-01.daraz.pk
Connection: Keep-Alive
Accept-Encoding: gzip
User-Agent: okhttp/3.9.1
```

Response Head

```
HTTP/1.1 200 OK
Content-Type: image/jpg
Content-Length: 150637
Access-Control-Allow-Origin: *
Last-Modified: Fri, 25 Jun 2021 13:38:36 GMT
timing-allow-origin: *
Server: Tengine/Aserver
EagleEye-TraceId: 0b1193b116291146054481543e6654
Strict-Transport-Security: max-age=0
Timing-Allow-Origin: *
Cache-Control: public, max-age=668259
Expires: Wed, 02 Feb 2022 05:04:52 GMT
Date: Tue, 25 Jan 2022 11:27:13 GMT
Connection: keep-alive
```

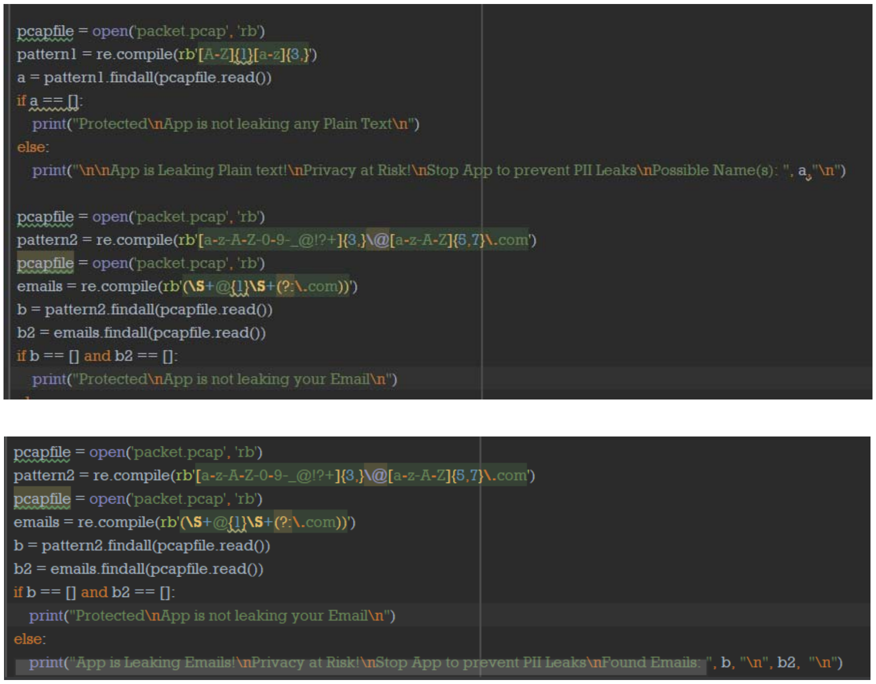
Figure 15: Analysis using NetCapture

Chapter 5: Coding and Testing

This Chapter includes the coding and development of PII RegEx patterns to identify PII and detect PII leaks occurring through Android Applications.

5.1 Coding

The main portion of our coding is composed up of PII RegEx patterns, that are the heart of this project. We have developed these patterns on our own and include various templates of PII to maximize the accuracy and broaden the scope of the project.



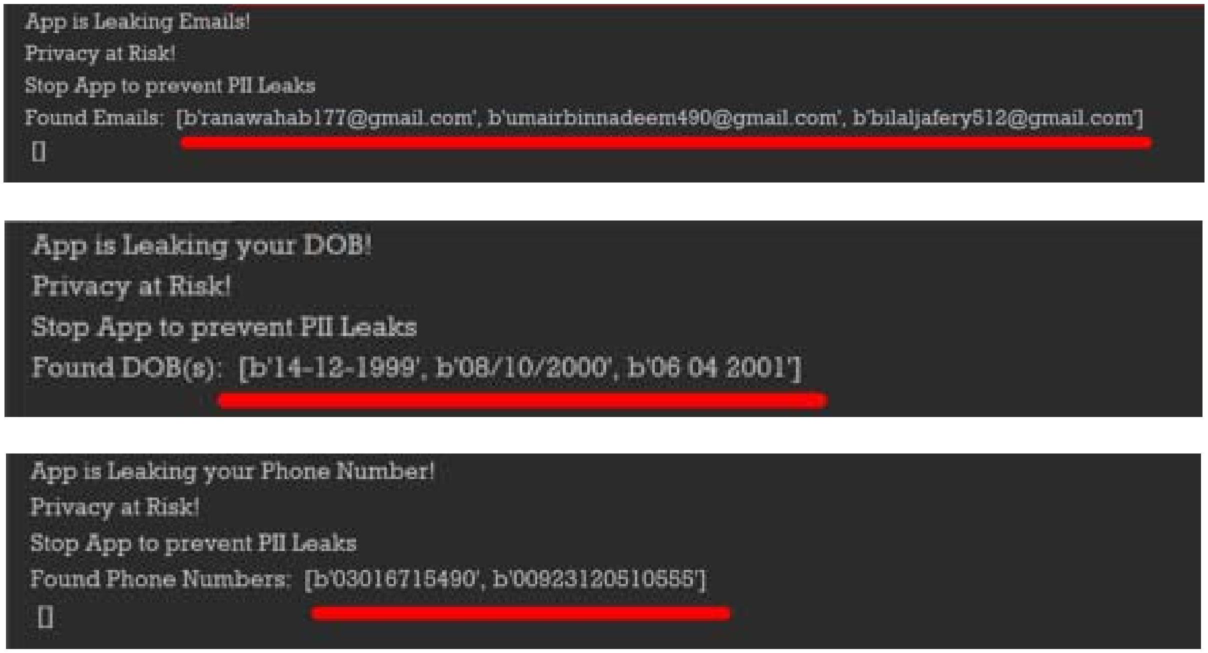
```
pcapfile = open('packet.pcap', 'rb')
pattern1 = re.compile(rb'[A-Z]{1}[a-z]{3,}')
a = pattern1.findall(pcapfile.read())
if a == []:
  print("Protected\nApp is not leaking any Plain Text\n")
else:
  print("\n\nApp is Leaking Plain text!\nPrivacy at Risk!\nStop App to prevent PII Leaks\nPossible Name(s): ", a,"\n")

pcapfile = open('packet.pcap', 'rb')
pattern2 = re.compile(rb'[a-z-A-Z-0-9-_@!?+]{3,}\@[a-z-A-Z]{5,7}\.com')
pcapfile = open('packet.pcap', 'rb')
emails = re.compile(rb'(\S+@{1}\S+(?:\.com))')
b = pattern2.findall(pcapfile.read())
b2 = emails.findall(pcapfile.read())
if b == [] and b2 == []:
  print("Protected\nApp is not leaking your Email\n")
```

```
pcapfile = open('packet.pcap', 'rb')
pattern2 = re.compile(rb'[a-z-A-Z-0-9-_@!?+]{3,}\@[a-z-A-Z]{5,7}\.com')
pcapfile = open('packet.pcap', 'rb')
emails = re.compile(rb'(\S+@{1}\S+(?:\.com))')
b = pattern2.findall(pcapfile.read())
b2 = emails.findall(pcapfile.read())
if b == [] and b2 == []:
  print("Protected\nApp is not leaking your Email\n")
else:
  print("App is Leaking Emails!\nPrivacy at Risk!\nStop App to prevent PII Leaks\nFound Emails: ", b, "\n", b2,  "\n")
```

Figure 16: PII RegEx patterns

5.2 Testing

After developing these PII RegEx patterns, the next phase was to test them. For this purpose, we used our self-developed Android application and different other applications available on PlayStore. The results were accurate and quite successful.



```
App is Leaking Emails!
Privacy at Risk!
Stop App to prevent PII Leaks
Found Emails:  [b'ranawahab177@gmail.com', b'umairbinnadeem490@gmail.com', b'bilaljafery512@gmail.com']
 []
```

```
App is Leaking your DOB!
Privacy at Risk!
Stop App to prevent PII Leaks
Found DOB(s):  [b'14-12-1999', b'08/10/2000', b'06 04 2001']
```

```
App is Leaking your Phone Number!
Privacy at Risk!
Stop App to prevent PII Leaks
Found Phone Numbers:  [b'03016715490', b'00923120510555']
 []
```

Figure 17: Testing Results

Chapter 6: Implementation

Now we'll talk about the most important components of our implementation. The HUNTDROID pipeline starts with data packet capture and then sends each packet to an analyzer to find PII leaks.

6.1 PII RegEx

The implementation of HUNTDROID involves the use of PII RegEx (short form of Regular Expressions) Patterns, developed in Python programming language. Regular expressions are significantly used in any data discovery and classification solution to identify sensitive material. Regular expressions are a specialized scripting language that is like wildcards on steroids. You specify rules that define the strings you want to match using this small language. You can create a RegEx that matches email addresses, PII, or credit card numbers, for example.

6.2 Extraction of PII

HUNTDROID will be able to automatically extract the PII from the network traffic by using PII RegEx Patterns. The extraction of PII must be validated because the privacy disclosures will be identified on the basis of identification of PII Leaks.

6.3 Dataset

To check the accuracy of HUNTDROID, we gathered app-generated traffic from about 200 Android applications and performed static analysis of these applications. We identified more than 100 PII leaks occurring through them.

Chapter 7: Conclusion

In this thesis, we have discussed a major problem of privacy disclosures and proposed a solution that can handle this problem. A lot of existing solutions are serving the purpose including but most of the solutions are commercial in nature. Open-source solutions exist but they provide limited functionality. HuntDroid will be accurate, would have minimum overhead, will amend to sources of ground-truth information and feedback from the Android users. Our proposed solution has multiple advantages over the previous traditional system due to the latest algorithms and analysis of their Android application usage. Our final step is the implementation of our solution in the form of an Android app. Any individual who is a smartphone user will face no difficulty in adopting it. It comes with a very user-friendly interface and does not require a certain level of technical knowledge. Using current methodologies, the goal of enhancing productivity and solving issues in existing solutions is realized.

We believe that because PII leaks happen via the network, detecting them provides an easily accessible remedy. This technique, we feel, opens a new path for privacy research and provides potential to increase privacy for regular users.

Chapter 8: Future Work

Future milestones that need to be achieved to commercialize this project are the following.

8.1 Targeting Encrypted Data:

The main objective of this project is to produce an Android App that should be easily accessible and easy to use for every Android user. Our project only aims to find PII leaks occurring via Android applications that communicated in an unsecured manner. In future, we look forward to add some functionalities in our product that will be able to detect PII leaks occurring via Android applications, even if they share or disclose user's personal information in the encrypted form. In addition to that, we have not included any mechanism to automatically stop the data leaks occurring through Android applications, our product is only prompting the users about PII leaks. But in future, we will add a mechanism that will automatically block the data leakages occurring through the Android applications without disrupting their normal functioning.