School of Computer Science and Statistics

Mobile Handset Anomaly Detection

Joseph Fitzpatrick
14312993

Supervisor: Prof. Douglas Leith
April 9, 2019

A Dissertation submitted in partial fulfilment of the requirements for the degree of Master of Computer Science
Declaration

I hereby declare that this project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have also completed the Online Tutorial on avoiding plagiarism ‘Ready Steady Write’, located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

Signed: ______________________  Date: _______________________
Abstract

Mobile devices are black-boxes when it comes to network traffic. We install applications and games onto devices, granting them a wide array of permissions, but we have little visibility of the network traffic these applications are generating. This lack of clarity of the network traffic being generated leaves the device vulnerable to be compromised.

This dissertation will outline a process for the identification of anomalies in network traffic between applications and external services. This will be done by first collecting data on network connections from mobile devices. From this data, analysis can be carried out to identify anomalies in network connections. The applications associated with these anomalies will then be examined in detail. The network communications between these applications and the remote server they connect to will be analyzed by performing a man-in-the-middle attack, bypassing the connection encryption by hijacking the SSL connection.

Finally, a study of the network connections of applications in emulated environments will be carried out. These network connections will be compared to the network connections observed in the real device. This will be in an attempt to identify if application developers change the functionality of applications in an emulated environment.
Summary

This dissertation discusses the ability to observe application anomalies in mobile phone network traffic. This is achieved through the implementation of a participatory study on peoples mobile phone traffic. This is carried out by installing an Android application on the participant’s devices, which logs the network connections and sends them to a remote server.

Once the study had been completed, graphs are made to begin to understand the content of the data. These graphs include information on common applications and URLs. Once an understanding is achieved, a more in-depth analysis of the traffic was carried out. Applications were clustered based on commonality of URLs connected to. With Grafana Dashboards, an analysis of the data over time can be carried out. With this, anomalies in the data are identifiable. A number of anomalies, including obscure port usage, uncommon data transfer sizes, and the phoning home of applications were observed. Other interesting observations such as what countries applications connected to were also observed.

With the application’s data analyzed over a period of time, an in-depth analysis of a number of the most common applications which displayed anomalies can be carried out. This involved the rooting of an Android device, to bypass SSL checks, to decrypt packets transferred to observe the content. Findings from this study included a number of GDPR violations and serious security flaws in applications.

An examination of the difference in operation between emulators and real devices was carried out, as some applications checked to see if they were running on real hardware. This involved the analysis of some applications across both device and emulator. No significant or interesting differences in network connections were observed between the emulator and real device. The fact that some applications didn’t run on emulated hardware was noteworthy.

The main research objective of this project is to carry out a comprehensive analysis of applications and the network traffic they produce to identify anomalies. This was achieved through the process of monitoring the application over a number of days, carrying out a visualization of the traffic produced by the application, identifying if any interesting network connections were made by the application, looking into the secure SSL tunnel at the content being transferred between applications and external servers and analyzing the behavior of the application on emulated hardware. The process outlined in the dissertation is one which can be replicated across applications and devices to provide consistent and fair results.
Acknowledgments

I would like to thank my family for their continued support throughout my undergraduate degree and masters degree. Without their support, I would not have reached this stage.

I would also like to thank my friends and classmates in Computer Science. Their friendship over the past 5 years have made the countless hours spent in computer labs considerably more bearable.

Finally, I would like to thank my supervisor Douglas Leith, who motivated me to continuously improve upon the work which I was carrying out and also his willingness to share ideas and thoughts on different aspects of the dissertation.
Contents

Summary iii

1 Introduction 1
   1.1 Motivation .................................................. 1
   1.2 Research Objectives ........................................ 2
      1.2.1 Further Objectives ...................................... 2
   1.3 Dissertation Overview .................................... 2

2 Background 4
   2.1 Mobile Security ............................................ 4
      2.1.1 Penetration Testing .................................... 5
   2.2 Network Security ........................................... 5
      2.2.1 Network Attacks ....................................... 5
      2.2.2 Proxy Server .......................................... 6
      2.2.3 Virtual Private Network ............................... 6
      2.2.4 Secure Sockets Layer ................................. 7
   2.3 General Data Protection Regulation ....................... 8
   2.4 Rooting .................................................... 8
   2.5 Emulator .................................................. 9
   2.6 Data collection ............................................ 9
   2.7 Data Visualization ....................................... 10

3 State of the Art 11
   3.1 Data Collection and Analysis .............................. 11
      3.1.1 AntMonitor ............................................. 11
      3.1.2 Phonelab ............................................... 11
      3.1.3 ISP level network monitoring ......................... 12
      3.1.4 Network Connection .................................... 12
      3.1.5 Firefox Lightbeam ...................................... 12
      3.1.6 Visualization Frameworks .............................. 13
      3.1.7 Anomaly Detection .................................... 14
## List of Figures

2.1 Man-in-the-middle attack diagram. ............................................. 6  
2.2 Proxy server architecture. .................................................... 7  
2.3 VPN Service architecture ....................................................... 7  
4.1 An architecture overview of the data collection process. .................. 18  
5.1 The real-time network graph of connections generated on a device. ........ 24  
5.2 CPU Usage at varying node counts. ........................................... 24  
5.3 Internet Protocol version usage percentages. .................................. 26  
5.4 The top 25 applications that generated the most traffic. ..................... 26  
5.5 The top 25 web URLs which were connected to most frequently. ............ 27  
5.6 The clusters of applications which share connections to the same websites. 28  
5.7 The periodic phoning home every two hours by the Truecaller Application. 29  
5.8 The calls made over a number of days by the Ryanair Application. .......... 30  
5.9 Port Usage Anomalies ............................................................ 31  
5.10 Network Anomaly in the size of packets being sent on the network. Peri- 
    odization in this participants device usage can also be observed. .......... 31  
5.11 The difference in traffic from the messenger application. .................... 32  
5.12 World map with the Tinder connections visualized. .......................... 32  
5.13 Map of where network connections were destined. ........................... 33  
6.1 Initial Network connections generated by the Bank of Ireland application on 
    start-up. .................................................................................. 39  
6.2 Information sent in clear-text on login. ........................................ 40  
6.3 Request which was replayed resulting in the transfer of funds being carried 
    out twice. .............................................................................. 41  
6.4 Login attempt to the Permanent TSB application. .............................. 41  
6.5 The graph.facebook.com connection request containing pieces of sensitive 
    information which can be used to identify a device or user. ................. 42  
6.6 Opt-in flag set to true despite no agreement to send data being accepted. 43
6.7 Matched identity, with the same identification tokens as used in the Paddy Power application. Note, this information was sent from an external source to the mobile application. .................................................. 44

6.8 The significant amount of quick succession network requests made by the Ryanair Application. .................................................. 45

7.1 Difference in the connection TCP Connection Requests being created by the Just-Eat Application on start-up was minimal. ...................... 51
7.2 A graph of the connections made by the device over a 10 second period. 52
7.3 A graph of the connections made by the emulator over a 10 second period. 52
7.4 An initial request generated by the Tinder application to a third-party tracking website. .................................................. 53

A3.1 A sample of the data which was collected. ....................................... 63
A4.1 Page 1 of 3. Connections made by the KBC Privacy Policy Webpage. . 64
A4.2 Page 2 of 3. Connections made by the KBC Privacy Policy Webpage. . 65
A4.3 Page 1 of 3. Connections made by the KBC Privacy Policy Webpage. . 66

A5.1 The network connections generated by the Revolut mobile application. . 67
A6.1 The variation in packet sizes over 30 transmissions. ........................... 68
1 Introduction

Mobile handsets are largely black boxes to users, with little visibility or transparency available as to how applications are communicating with external network services. This dissertation will outline a process for the identification of anomalies in network traffic between applications and services. An examination of the network content will also be carried out. A comparison in application functionality on emulators compared to real devices will also be performed.

1.1 Motivation

In the past few years, the volume of internet traffic generated from mobile devices has grown exponentially. Currently, over 52% of all internet traffic comes from mobile devices (1). Handset security has generally been left to the device manufacturers or to the Operating System that is running on the device. This has left mobile devices as black-boxes when it comes to network traffic. We have very little visibility on what network traffic is generated, where they connect to and what information is sent in them.

We install applications and games onto our devices, granting them a wide array of permissions without much thought to what these applications are doing. Unlike on a traditional desktop computer, we can’t see the network connections with an inbuilt command such as tcpdump(2). More intricate monitoring techniques are required such as connecting to a proxy such as WireShark or Fiddler and examining the packets captured. This lack of clarity in the traffic being generated leaves devices vulnerable to be compromised.

Devices will need to be secured as more day to day activities move to their mobile devices. Currently, over 78% of Irish people use mobile banking to make everyday payments, a number which will only rise (3). With a growing number of attacks taking place against users, a vulnerable attack surface, which a mobile device is, attracts a malicious interest.
Mobile devices are expensive, and as such, testing applications on multiple real devices have a high cost associated with it. To avoid this cost, developers use emulators to simulate a real device. Many reports claim that applications function differently on emulators, therefore a comparison of the network traffic between an emulated device and real device would back up these claims.

1.2 Research Objectives

The main research objective of this dissertation is to identify anomalous network connections from applications. Further to this objective, there will be a number of intermediary research objectives which will need to be achieved.

- A data collection study will be carried out to obtain the data required to identify anomalies.
- A real-time visual representation of the network connections being generated on a device to allow the user to examine their connections in greater detail.
- An analysis of applications HTTPS traffic to view what information is being sent to third parties.

1.2.1 Further Objectives

Should the preceding objectives be achieved, an analysis of application behavior within an emulator will be carried out. This objective will be used to further validate the results from the physical device application analysis, and to investigate if application behavior has been modified to operate differently in an emulated environment.

1.3 Dissertation Overview

The dissertation will be structured as the following.

- Chapter 2 will be a discussion on the background of the project, explaining some of the areas of research in greater detail.
- Chapter 3 will contain the research carried out on the state-of-the-art work in this area, this section will provide the reasoning for the design decisions taken.
- Chapter 4 will be a description of the design of the study and subsequent processes of application analysis.
• Chapter 5 will be an overview of the data collection study, and how it was carried out.

• Chapter 6 contains the process for analyzing the applications.

• Chapter 7 will examine the differences between the emulated device and real device.

• Chapter 8 will be a conclusion on the work as well as a recommendation for future work that could be carried out.
2 Background

This chapter will aim to give an overview of a number of the topics covered within the dissertation, providing some additional information which will justify some of the decisions made later in the dissertation. Some of the privacy and security concerns related to the dissertation will also be covered.

2.1 Mobile Security

A mobile device is a computing device small enough to be operated with one hand. The device generally has a touch-sensitive LCD or LED display where user’s of the device can interact with applications and games on the device. The device generally has a number of additional sensors and utilities, such as Wi-Fi connectivity, Bluetooth, NFC, and cellular access. Cameras and speakers are generally integrated into the device. The device generally runs a mobile-specific operating system, usual either Android or iOS. These operating systems allow users to install applications onto the device from centralized "Application Stores" and from third-party sites. There are approximately 2 Billion "smart" mobile devices in the world.

Mobile devices have become ubiquitous in our daily lives. Over 97% of the Irish population have access to a smart device (4). With the growing number of devices, the threat associated with owning these devices is growing. Handset security has generally been left to the device manufacturers or operating system running on the device. This has left mobile devices vulnerable to attack when updates for the device stop being produced. As we move towards a society where a significant proportion of day-to-day activities take place on mobile devices security has become a bigger issue.

A common trend nowadays is for companies to adopt a "Bring-your-own-device" or BYOD policy on mobile devices, with over 60% of companies adopting a BYOD friendly policy (5). This policy, while reducing costs for companies, also increase the attack surface for corporate networks (6). A compromised device could be used as an attack vector on the network. To limit these risks, companies install anti-malware software on
devices to protect them.

Most anti-malware software solutions available on mobile devices are paid or subscription-based services or applications. These applications and services generally work as resource intensive applications which cause faster battery drain as well as a slow down in the device.

2.1.1 Penetration Testing

Penetration testing is a formulaic approach for finding and identifying vulnerabilities or issues in computer systems. Penetration testing comes under the umbrella term of ethical hacking, where the hacker testing the system discloses the vulnerabilities to the company prior to publishing their results. The testing is usually carried out by professional teams as part of companies compliance certification processes.

2.2 Network Security

Network security is an area of information security focusing on all aspects of network related activity. Areas including access to networks, authentication of network activity and integrity of information transmitted are all aspects of network security.

2.2.1 Network Attacks

A network attack is defined as a method or process used to compromise a network for malicious purpose. Attacks can be carried out by one or more attackers for a variety of reasons such as extortion of money from the attacked network, to gain access to sensitive information transmitted on the network or for recognition within the hacking community. Attacks can generally be classified into two main types, external attacks, where the attacker does not have access to the internal system and internal attacks, where the attacker has access to the internal system.

For the purpose of this dissertation, we will be examining some of the external attacks. Popular external attacks include man-in-the-middle attacks and devices being used as part of external bot-nets. Distil Research labs estimate that 5.8% of devices are being used to make bad bot requests on cellular network connections (7).

The attack that will be carried out as part of this dissertation will be a man-in-the-middle attack. This attack involves two parties communicating together over a channel. A third party then intercepts the traffic on the channel. The third party then forwards to the
designated destination. Neither party is aware of the third party intercepting traffic on the channel. With the large majority of all network traffic using HTTPS (8), the popularity of man-in-the-middle attacks has declined, meaning the intercepted packets cannot be read in clear-text format. More complex variations of this type of attack are explored in this dissertation.

2.2.2 Proxy Server

A proxy server is an application which operates as a gateway between a local network and an external network. Proxies are used for a number of reasons such as monitoring local network traffic, caching of content on local networks and security blocking certain websites from the local network. Proxy servers can be used with mobile devices to log all of the network traffic which flows through the device. A diagram of the proxy server architecture can be seen in Figure 2.2.

2.2.3 Virtual Private Network

Virtual Private Network (VPN) is a secure tunnel between a device and the internet. It is used to protect the users’ online traffic from external snooping. A VPN allows the device to communicate with the internet as if it were on its own private network. The request is encrypted on the device and sent to the external VPN server. This server then sends the request to the destination. The destination sees only the request coming from the VPN.
server, so the device’s IP address and location are not compromised. A diagram of the VPN architecture can be seen in Figure 2.3.

2.2.4 Secure Sockets Layer

The secure socket layer (SSL) is a cryptographic protocol used to communicate securely over a computer network. Websites and applications use SSL in the form of TLS or HTTPS transfer protocol to secure communication to a remote server.

The connection is secured by the party who is sending the packet across the network retrieving the digital certificate and public key of the intended destination from a Certification Authority (CA). The CA is a trusted third party which verifies the authenticity of the certificate. The packet is encrypted with the public key that was retrieved and sent across the network. Only the intended destination has access to the corresponding private key of the public key which was used to encrypt the packet. This private key can be used to decrypt the packet.
2.3 General Data Protection Regulation

General Data Protection Regulation (GDPR) was a piece of EU legislation passed in 2016, which came into force in May 2018, regarding the data protection and privacy of individuals in the EU (11). The regulation aims to give individuals control over their personal data. The regulations cover a wide array of data privacy issues, from data transport, data pseudonymization and data storage. For the purpose of this dissertation, some of the more relevant areas are around the consent for data collection.

Data collection of personally identifiable data such as device or Android ID information should be "opt-in", ie. consent for the collection of this type of information should be granted prior to being collected, or shared with third parties. The name of the company and name of any third party whom with this type of data is shared should also be informed to the user.

Failure to comply with these guidelines can result in fines of up to €20 Million or 4% of the previous financial years annual turnover, whichever is higher. This regulation has been in place for almost a year (at the time of writing), so companies would have had sufficient time to put the necessary protocols in place to avoid breaching them.

2.4 Rooting

Rooting is the process of unlocking a device’s operating systems root privileges. The Android operating system is based on the Linux kernel, so unlocking root privileges gives superuser access to the system. Rooting is generally carried out to circumvent limitations placed on the hardware by the device’s network or the device manufacturer. Rooting is common across all mobile operating systems, with "jailbreaking" being the colloquial term to describe the process on iOS devices.

A rooted device gives the user the ability to customize the device in a number of ways such as installing custom boot options or applications which require root privileges. Other advantages include the ability to modify device hardware including over-clocking the CPU and GPU, uninstall system applications and install custom ROMs and software on the device. In android Xposed and Magisk are two common pieces of software which give the user access to a wide array of root applications for system modification.

Rooting devices is not common beyond the technical community for a number of reasons. The process of rooting is not a simple one, with the many variations in android devices, guides on the process are purposefully vague. This leads to an erroneous process which
leads to many devices becoming "bricked" or unusable. Rooting also voids the warranty on the device as it is generally not recommended by manufacturers and carriers.

2.5 Emulator

An Emulator is a piece of software which allows for one operating system to operate as a guest on another operating system, the host. The guest uses the host’s peripherals such as network connection, pointing device, and keyboard. The emulated operating system imitates the functionality of the non-emulated version of the operating system.

Emulators are used for a wide variety of reason including change and security testing. Emulators are cheap to set up, often using free software, and also can be scaled up rapidly without significant hardware upgrades. Emulators give the user control over all system processes which allows for greater insights into what is actually happening on the operating system. Emulators can also act as a "sandboxed" versions of the operating system, where malicious acts can be carried out without the risk of the host’s physical hardware being compromised.

2.6 Data collection

Before any anomalies can be detected, data must be gathered to be examined. There were ethical considerations which were taken into account prior to this process. Ethics forms presented to the participants can be seen in Appendix A1 and A2. There are a significant amount of privacy concerns associated with this aspect of the dissertation, due to the sensitive nature of the information being gathered. The data which will be gathered through the use of a survey of participants can possibly contain information that the user would not like to have disclosed. Further to that, by carrying out analysis of the data that is collected it may be possible to identify a users traffic.

The data collection process in itself is a privacy concern. It is an intrusive service which may disturb user activities. However, this concern is highlighted when a participant signs up to partake in the study. Each user will be allocated a random large integer to identify the device. This number will be a pseudonymous link between the device and the data. As such it is an identifiable attribute, so it is not used in an unencrypted format in the transfer of data across the network to minimize the privacy concern.

As the data being collected contains timestamps of when the connection was made, it may be possible that if the stored data was compromised, a correlation between the user’s
data and some external event may be possible. To avoid this, I have stored the data in a secure server to avoid any unauthorized access to the data.

Identification of users from the data may also be possible. By analyzing the sites visited, it may be possible to identify a user based on the URLs they visit. In an attempt to mitigate this, only the hostname of a site will be stored. This will make the process more challenging as they would need to incept a significant amount of traffic to profile a user for identification.

The data that is collected as part of this dissertation is of time series format. This is where the data contains the timestamps of when the entry is created. Time Series data is used across a wide array of industries in the monitoring of change over time. The collection of time series data can also be referred to as Panel Data.

## 2.7 Data Visualization

The data is real-time internet traffic generated from a mobile device. The data contains both categorical data and quantitative data which adds to its complexity. It contains the URLs visited, the application name that generated the data, and the IP of the destination. It also contains information relating to port usage, the protocol used and connection timings.

Visual encodings will be essential in the visualization of the data due to the amount of categorical data we are presented with. Having separate colors for different applications, different sized points for connection times will be important. It will be essential to ensure that data will be easily discriminable so information can be understood quickly. Positional encodings will probably be the most impactful from the data, as it is a time series dataset. Graphing the change over time in particular trends of actions will be easily visible if correct positional information is used.
3 State of the Art

This chapter will examine a number of the state-of-the-art activities being carried out in mobile handset anomaly detection. It will also provide an analysis of current mobile network visualization tools. While this chapter will outline the most relevant research on state-of-the-art technologies, it represents just a sample of the research that was carried out throughout the project.

3.1 Data Collection and Analysis

3.1.1 AntMonitor

AntMonitor is a mobile application which acts as a VPN on the mobile device to log internet traffic (12). The application provides a significant amount of functionality through its logging and VPN attributes. AntMonitor was developed by a team in the University of California as a complete system for on-device passive monitoring, collection, and analysis of packet monitoring. The application is built around a VPN service, which routes device traffic internally and logs it.

AntMonitor contains a mobile-only VPN service which logs all of the packet information from network activity. It then sends this information to a remote server for further analysis. The high-level architecture design proposed by the AntMonitor paper will be used as a guideline for the data collection phase of the project.

3.1.2 Phonelab

Phonelab was an on-device monitoring application developed by the University of Buffalo (13). It allows phone interactions to be recorded and logged in an on-device database. This on-device database contains information regarding the devices state at set intervals i.e. what interactions in the application resulted in network activity. However, the application requires a rooted device which limits its widespread use. The additional pieces
of information from the applications, which were used in this project, will be added to the data which is collected from the devices. This will be for attempting to gain greater context of the application state when the connection was made.

### 3.1.3 ISP level network monitoring

A number of studies have been done on high-level network monitoring such as Garcia, 2007 (14) or Mansmann, 2007 (15). These studies are carried out at the ISP level and are used for network planning. While these studies gather great amounts of information on the traffic generated by users, it is mainly high-level network flows that are collected. This lacks the granularity for attributing network traffic to individual users and devices. However, as highlighted in these papers, having an overall view of network activity is important. Therefore an aggregated view of the network will also be examined.

### 3.1.4 Network Connection

Network Connection is one of the most popular Android Apps for examining network connections on the Play Store with over 500,000 downloads (16). The application works by linking the outbound IP connections of an application and logs this entry. The simplicity of this system makes it attractive for users who simply want to view all the connections the phone is making. Unlike most other applications in this area, it does not require the phone to be rooted, once again broadening its appeal to users.

The application has a number of limitations, however. It only allows users to view the connections. No ability to block connections or IP addresses is available. It also lacks any visualizations within the application. A log stream is used instead, limiting a users ability to gain meaningful insights. Overall the application, while it has user appeal due to its simplicity, is limited in its functionality.

### 3.1.5 Firefox Lightbeam

Lightbeam is a desktop only firefox extension for the visualization of network traffic with over 140,000 installs (17). The extension enables the user to see the third-party websites that the browser interacts with on a page load. It provides both a list and graph of the network, which is highly effective in showing all users the active connections that are being made.

The visualization on this extension is highly effective in clearly delineating to users their connections. The network graph was designed as a joint project with Mozilla and the
Emily Carr University of Art + Design.

While Lightbeam is highly effective, it is limited in that it only works on Desktop web browsers. However as more internet traffic is moved into dedicated applications, this browser focused architecture isn’t as effective. The network graph style presented by Lightbeam is what was adapted to work within this dissertation for the visualization of the network traffic.

3.1.6 Visualization Frameworks

D3.js

D3.js is a highly effective data visualization tool which is built on top of Javascript. It leverages the portability of HTML with the functionality of Javascript to produce high-quality SVG based visualizations. With all of this functionality, a performance hit is taken. As power consumption and processing power on mobile devices are limited, D3.js is a less effective choice.

Vis.js

Vis.js once again uses both HTML and Javascript to produce SVG based graphs. Vis.js has significantly less functionality than D3.js however, is more suited to create dedicated network graphs. It allows for greater control over the physics aspect of the visualizations meaning a more complex graph can render quicker, by limiting the physics-based computation at the start. PyVis is also a python based extension for Vis.js for rendering graphs.

Grafana

Grafana is a data visualization tool that provides a framework to create and analyze data in a visual dashboard. Grafana works as a front-end visual interface which works with a back-end time-series database such as Graphite, or InfluxDB. Grafana is one of the leading time-series graphical monitoring tools with companies such as PayPal and eBay using the service for their own monitoring infrastructure.
3.1.7 Anomaly Detection

A renewed interest in anomaly detection in time-series data has arisen over the past number of years. There are a wide variety of approaches being worked on by a number of different universities (18), (19) and companies (20), (21). Data Visualization is the initial step taken to detect anomalies in data. If the data is graphed correctly, the anomalous traffic should be identified as being clearly different from the rest of the traffic.

Following from the visual inspection, a number of different technical approaches are available for automatic anomaly detection in multivariate data.

- Multivariate Statistical Analysis. This process employs multiple statistical methods to detect anomalies in data. The process works with a dataset of entries with a certain distribution, the probability distribution of the dataset is calculated $p(X)$. Each new data-point $x$ is then compared to a threshold value $r$. If the $p(x)<r$ the data-point is then observed as an anomaly.

- Artificial Neural Network (ANN). This approach is based on an auto-encoder neural network. The network analyses the data and learns a representation of that dataset similar to the probability distribution, as seen in the statistical analysis above. The network’s aim is to reduce the number of variables required to observe if the new data-point is an anomaly. ANN come in many forms with most consisting of an input layer, hidden layer, and output layer.

3.2 Application Analysis

3.2.1 OWASP

OWASP is the leading cybersecurity organization in the world and provides a standard for application security verification (22). This standard forms the basis of security ratings that are given to applications. The standard is aimed at each member of the development life cycle of application including the QA tester and end-user.

It contains information regarding the majority of common security vulnerabilities in applications. Each vulnerability is examined and its impact is given. Guidelines on how to test for this security vulnerability are also presented. Each vulnerability is given a severity rating based on the impact and likelihood of it being attacked.
3.2.2 Decompiling Applications

Decompiling applications downloaded from the app store and modifying the bytecode is a common method of application analysis. The method of decompiling applications to examine how they communicate with external services is discussed in Micheal Grace’s 2012 paper on Unsafe exposure analysis of mobile in-app advertisements (23). Applications are downloaded from the Play Store and decompiled. The resulting decompiled application includes the AndroidManifest.XML file. This file is altered to examine the permissions the application is using, and examining the advertising agencies the application connects to.

Ravitch, 2014, also examines application behavior by decompiling it (24). This paper examines the decompositions of applications and modification of byte code to manipulate permissions beyond their scope. It also examines the opportunity of collusion between applications. The ideas presented by Ravitch of collusion between applications and services is also a feature of the analysis of applications within this project.

3.3 Emulator vs. Real Device

3.3.1 TaintDrive

TaintDrive, a study between Intel Labs and Duke University, uses emulated Android devices to examine privacy concerns in a number of different applications (25). The study focuses on the generated Java byte code produced by the applications to monitor when sensitive information is extracted from the device. They then use the Dalvik VM Interpreter to convert this back to a human-readable format for examination.

The study examined 30 applications and found that 20 of them had potential misuse of user data. The misuses were identified by examining the generated binary packages within the Virtual Machine memory. Packets leaving the virtual machine were also inspected. The process for the application study as outlined in the paper was used as a guideline for the process carried out in analyzing the applications, it, however, focuses on the lower level aspects of application decomposition, which was not carried out.

3.3.2 Mobile Application Testing

Testing new applications on both emulators and real devices in an important part of the development lifecycle of an application (26). The ability to scale tests on emulators
without the cost implications of renting or buying new devices makes their use significantly more advantageous in the testing of applications. The real device, however, should be tested on as well to imitate real-world constraints of physical devices. There are multiple different types of testing on mobile applications which are carried out throughout the development process.
4 Design

This chapter will outline the process used to identify and examine anomalies in mobile network traffic. The process has three main components in data collection, application analysis and comparison of application functionality in an emulated environment compared to a real device.

4.1 Problem Setup

Detecting anomalies in network traffic is a data-based problem. As, by its definition, anomalies are rare, the more data on network connections that can be obtained, the more likely anomalies are to be identified. For this reason, a participatory survey will be required to gather the data. The survey will cover a range of devices over a two-week period of time.

The data will only explain part of the story. Sequencing the network traffic with in-application actions will be important in deciding if the traffic is anomalous. In-depth analysis of the actions within an application, and linking them with the network connections will be required.

Some developers encode certain features in the application to detect if the application is being run on an emulated device. It can be hard to identify why, or if, a developer is doing this. By comparing the network connections generated from an emulated device compared to a real device, we can decide if application network behavior is altered on emulated devices.

4.2 Data Collection

To be successful in the identification of anomalies in mobile internet traffic, data on network connections will need to be collected. However, logging network connections directly cannot be done on Android devices. To get around this, an on-device VPN service
will be used to log traffic, similar to one proposed by the AntMonitor paper. The VPN service will route network traffic through it, and maintain a log of all of the connections. Some additional information related to the connection, such as the application and process ID of the application which generated the connection will also be logged. The log will then be synced with an external server to provide a centralized collection of the traffic for analysis.

![Diagram of data collection process]

Figure 4.1: An architecture overview of the data collection process.

The Android operating system was chosen as the target operating system for this dissertation. Android is the most popular mobile operating system with over 2.3 Billion devices (27). Android also offers a greater range of devices. With iOS only running on iPhones, security vulnerabilities can be easily patched across the devices in comparison to Android’s larger number of unique devices requiring custom patches for fixes.

For the purpose of this dissertation, an open source VPN application "NetGuard" will be used as a base for development. NetGuard is an Android application which acts as an on-device VPN application (28). A selection of other applications and services were considered prior to this choice. An initial survey using the "Corrata" VPN application proved unsuccessful with limited time granularity available.

A significant proportion of NetGuard’s functionality is only available as a paid service. The open source version’s functionality will be extended to include extra features such as logging and remote data upload to a centralized database.

The application will be extended to include a real-time visualization of the network traffic on the device also. This visualization will allow the user to engage with the application and gain value from it. This will be used to incentivize users to participate in the survey.
4.3 Application Analysis

When setting up the experiment for analysis, a number of design considerations needed to be considered. Standardizing the testing across the applications as much as possible was key in ensuring fairness across different applications.

4.3.1 Android Device

All of the applications will be tested on the same Android device. For the purpose of this dissertation, the selected device was an LG Nexus 5, running Android Marshmallow 6.0. This device was deemed appropriate for experimentation as there is a large volume of resources available on rooting the device and the device runs a stock version Android i.e. no device manufacturer’s applications or customizations are installed by default onto the device.

4.3.2 Xposed

Modifications to the device itself were needed to enable a more extensive analysis of the application’s traffic. To allow for this the Xposed Framework was used. The Xposed framework is an open source Android Application Package (APK) which acts as a manager for custom modifications to the Android operating system (29). The framework gives the user access to a number of modules to modify aspects of the device, from modifying camera software to custom boot launchers.

Xposed offers a number of modules for analyzing network traffic. These can be used to manipulate the encryption process used in securing network traffic by the application. The encryption of the SSL Tunnel from the application to the remote server will be targeted in this analysis. To do this the module SSL Unchecker is configured for use. The module works by bypassing the initial check to see if the client’s SSL certificate matches the remote Certificate Authority’s certificate. This bypass allows for custom certificates to be used by the application and application traffic to be viewed.

4.3.3 Fiddler

The traffic that flows from the device has to be captured and analyzed. A way to do this is to use a proxy server on a third-party device on the network that the device is connected to. Fiddler is a multi-platformed approach to this, giving the user a GUI to view and interact with live sessions (30). It allows the user to replay requests as if the
devices are making them. Fiddler also can provide a custom SSL certificate for use by the
device in the encryption process. This certificate, in combination with the SSL unpinning
provided by the Xposed module, allows for network connection’s content to be viewed in
plain-text.

4.4 Emulator vs. Real Device

Emulators are an important resource in application development. Mapping their
functionality to real devices would indicate any anomalies in the functionality of the
application. To mitigate any external influence on the emulators, the same version of
Android, as tested on the real device, was installed on the emulators. This version of
Android which was used is Android Marshmallow 6.0.

4.4.1 Emulators

For the purpose of this project, two emulated environments were examined.

- Android Virtual Device (AVD). The AVD is an emulator provided by Android
  Studio, the Android application development IDE. This IDE is provided by Google
  and is the most important piece of software for the development of Android
  applications. As such, examining the functionality of applications within this
  environment was important. Any deviations in network activity detected from the
device would indicate the possibility that application behavior had been changed
when running in an emulator.

- Virtual Machine VirtualBox (VMBox). VMBox was selected also to test the
  applications on. Virtual machines allow the user to manipulate all of the inputs into
the operating system running on them. All network traffic traveling through the
device can be monitored.

4.4.2 Network Analysis of Emulated Devices

Capturing the network traffic which flows in and out of emulators will be carried out
through the combination of a number of different tools. The traffic will firstly be
intercepted with a packet capture tool called WireShark. WireShark is highly regarded as
one of the best open source packet capturing tools (31). The program monitors all of the
network flows in fine-grained detail i.e. each individual packet has its own entry.
The captured network packets will then be exported in a Packet Capture (PCAP) file to Fiddler for a high-level network flow examination. This is done to ensure that the logged traffic appears in the same granularity as it does when examining the traffic from the real device. Similar to the real device, Fiddler’s encryption certificate will be added to the emulated devices for packet analysis.
5 Data Collection and Analysis

This chapter outlines the steps taken in gathering the required data for the study and the analysis of the data to identify anomalies.

5.1 Data Collection

Anomaly detection is a data problem. To identify a rare event or occurrence in data, first a large collection of data needs to be gathered. The data can then be profiled and used to gain insights on whether anomalous can be observed.

For the purpose of this dissertation, the data that will be collected are the network connections a device makes. These connections generated from the devices will contain information regarding the application which generated the data, network information, such as the protocol used and the IP address connected to, and a timestamp. These connections will then be used for preliminary analysis of applications.

To ensure a sufficient number of people could partake in the data collection survey, an Android application, which would require a non-rooted device was developed. Each logged network request would be sent to a remote centralized server for storage until enough data had been gathered.

5.1.1 NetGuard

The open source application, NetGuard, is the Android application that will be used as the base of development for the data collection segment of the project. Its existing VPN functionality will be extended to include data logging, data transfer, and data visualization tools. The application is written as a mix of C++ and Java code.
5.1.2 Device Data Logging

NetGuard’s functionality was extended to provide more in-depth logging capabilities to store information relating to the request itself and application which generated it. Fields added included connection times, Server locations, Application information, IP geolocation, and connection protocol. The data was sent to a remote server located in Trinity College Dublin via a RESTful API POST request.

A sample of the data which was collected can be seen in Appendix A3.

5.1.3 Data Visualization

Adding a data visualization piece was a non-functional requirement of the dissertation, however one which was used to encourage participants to use the application. The visualization was a real-time network graph of all of the network connections produced by the device. The visualization allowed the participants to view the application traffic.

Data Visualization on the Android Application was created using the Vis.js package. Vis.js is a browser-based visualization library built in JavaScript. The type of graph that was chosen to represent the traffic was a Network graph. These graphs are comprised of nodes and edges. Each node is connected to one or more other nodes. The rectangular nodes in the graph represent the application, and the circular nodes represent the websites that the application connected to.

To use this library in an Android Application, it needs to be rendered using a WebView Client call. This call generates an object on the page which allows for the display of HTML content. The HTML file loaded in this call is from the Android Assets folder, therefore making it read-only at run time.

To make the graph real-time and display the live connections to the user, an additional Javascript function was required. This function works by loading a JSON file, where the last recorded connection is stored, from the devices cache storage whenever a change is detected. This load is performed using XMLHttpRequest. The file is decoded and the new network connection is added. New nodes on the network are added using the addNode command, and edges are added using addEdge. If the connection is already on the graph, the weight of the line is increased to indicate the connection.

One of the most important factors in the visualization was optimizing the battery and CPU usage on the device. Too high of either would result in reduced application functionality. As the number of nodes grows, the render times in the graph do not increase with each node. However, dragging and zooming on the graph, grow the
complexity linearly with each new node. To limit the impact of this on the effectiveness of the graph, the nodes on the graph are reset once the application is left idle for a period of time. This optimization decreases the resource load of the application.

Figure 5.2: CPU Usage at varying node counts.

5.1.4 Server

The logged data was sent to a remote server to be handled and processed for use in anomaly detection. The server was a Python Flask application which logged the data
received via a POST request to a text file for use. As the stored data contained sensitive information relating to the participants, an access control method of securing the data was employed.

5.2 Data Analysis

The data that was logged in the Android Application was analyzed to uncover anomalies in network traffic. There were 350,000 individual connections logged across 5 devices in a two week period. The data that was collected contained a range of attributes including the timestamps, protocol, and IP address. In total, traffic from 34 application was logged, with traffic from 16 application being logged from all 5 devices.

5.2.1 Data Preprocessing

Before the data could be analyzed, the data was cleaned to remove any erroneous data which may occur in the data. The study was to analyze network connections from applications however, the network connections generated from a browser are related to the website the user is visiting, more so than the application itself. For this reason, entries which contained data relating to internet browsers were removed from the data.

The IP address contained in the application were also geolocated. This was achieved by performing a "whois" lookup of the IP, which returned the country code of the IP. Data entries which did not relate to an application, such as system DNS lookups, were also removed as analyzing these calls were beyond the scope of this project.

5.2.2 Application Analysis

Initial application analysis involved trying to get an understanding of the shape of the data. This involved the creation of bar charts and graphs to get an initial understanding of the data. From these graphs, further analysis of applications was carried out. With these graphs, the challenge of deriving context and insights was evident due to a large amount of data which was gathered. More complex graphs and graphing software would be required to be able to identify trends in the data and to map relationships between applications and their network connections.

Applications were clustered to observe the commonality in network connections between applications. The clustering was based on a hierarchal repulsion model, where the distance between applications is derived from the inverse of the commonality of the
connections between applications. To represent this a network graph was created as can be seen in Figure 5.6. The closer applications appear to the center of the graph, indicates that they shared network connections with the most number of applications i.e. connected to the same websites. Applications located towards the edge of the network shared connections with a fewer number of applications. Applications which did not share common network connections were not represented on the graph.

The "phoning home" of some applications was also observed in the data which was collected. Phoning home is where an application periodically sends a connection request to a designated server, even when the application or phone is not in use. An example of this was with the Truecaller application. Truecaller is an application which blocks spam calls. An example of the calls over a 12 hour period can be seen in Figure 5.7. The
application connects to presence-grpc-eu.truecaller.com every 2 hours.

Another application which partook in "phoning home" was the Ryanair application. Ryanair made periodic calls multiple times a day, over a number of days to 3 websites (graph.facebook.com, settings.crashlytics.com and 6178.eu-api.swrve.com) as seen in Figure 5.8.

5.2.3 Visual Anomaly Detection

Initial anomaly detection was done through a visual inspection of the traffic using the generated graphs from Grafana. These graphs could be viewed at a range of granularities. Observations were carried out on a per-application basis. Anomalies were classified as a visual deviation from the data trend that was presented in the graphs. A range of fields was graphed including protocol, port number, connection time, geolocation of destination IP’s, and process ID’s.

When an anomaly was identified from the visual examination, the entry corresponding to that anomaly was flagged for further examination. Figures 5.9 and 5.10 show some of the anomalous traffic identified from visual inspection.

Figure 5.9 shows usage of ports across a 24 hour period. 5 of the 300 data points observed across that time period are significant outliers in the data, therefore can be deemed anomalous. The anomalous entries connected to port 57760. Normal port usage
is generally port 443 and 80 for HTTPS and HTTP traffic respectively.

In Figure 5.10 a clear deviation in the pattern can be seen. The green dots represent the size of packets being sent across a connection. As this event is witnessed only once across the data, it is not standard in the network traffic observed.

The Messenger application had interesting geolocation of the IP addresses. All of the devices connected to addresses located in both Ireland and the United States, however one of the devices connected to IP’s located in other countries. As seen in Figure 5.11, connection requests were sent to IP addresses in Switzerland and Germany.

An interesting observation from the data which was gathered was from the Tinder application. The application is a location-based application which matches users with others within a small geographical area. When analyzing the geolocation of the IP addresses connected to by the application, over 80% of the IP’s were located in the United States. This was interesting as questions around what data is being sent outside
<table>
<thead>
<tr>
<th>Time</th>
<th>Service</th>
<th>Identity</th>
<th>AS</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-02-28 20:40:00</td>
<td>Truecaller</td>
<td>api.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-02-28 21:38:38</td>
<td>Truecaller</td>
<td>eu1.wzrkt.com</td>
<td>AS16509</td>
<td>Amazon.com</td>
</tr>
<tr>
<td>2019-02-28 21:39:10</td>
<td>Truecaller</td>
<td>api.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-02-28 21:50:32</td>
<td>Truecaller</td>
<td>api.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-02-28 23:02:28</td>
<td>Truecaller</td>
<td>graph.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 00:25:02</td>
<td>Truecaller</td>
<td>presence-grpc-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 00:50:16</td>
<td>Truecaller</td>
<td>api.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 01:37:53</td>
<td>Truecaller</td>
<td>graph.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 01:39:29</td>
<td>Truecaller</td>
<td>account-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 02:26:26</td>
<td>Truecaller</td>
<td>presence-grpc-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 03:43:56</td>
<td>Truecaller</td>
<td>account-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 03:43:57</td>
<td>Truecaller</td>
<td>account-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 03:43:57</td>
<td>Truecaller</td>
<td>eu1.wzrkt.com</td>
<td>AS16509</td>
<td>Amazon.com</td>
</tr>
<tr>
<td>2019-03-01 03:55:03</td>
<td>Truecaller</td>
<td>graph.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 04:26:27</td>
<td>Truecaller</td>
<td>presence-grpc-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 04:48:57</td>
<td>Truecaller</td>
<td>account-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 04:49:07</td>
<td>Truecaller</td>
<td>graph.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 06:26:28</td>
<td>Truecaller</td>
<td>presence-grpc-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 07:05:46</td>
<td>Truecaller</td>
<td>api.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 08:17:34</td>
<td>Truecaller</td>
<td>api.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 08:17:37</td>
<td>Truecaller</td>
<td>account-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 08:17:37</td>
<td>Truecaller</td>
<td>settings.crashlytics.com</td>
<td>AS14618</td>
<td>Amazon.com</td>
</tr>
<tr>
<td>2019-03-01 08:17:38</td>
<td>Truecaller</td>
<td>account-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 08:17:38</td>
<td>Truecaller</td>
<td>api.facebook.com</td>
<td>AS32934</td>
<td>Facebook</td>
</tr>
<tr>
<td>2019-03-01 08:17:39</td>
<td>Truecaller</td>
<td>account-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 08:17:39</td>
<td>Truecaller</td>
<td>presence-grpc-eu.truecaller.com</td>
<td>AS199544</td>
<td>Remote24 AB</td>
</tr>
<tr>
<td>2019-03-01 08:17:40</td>
<td>Truecaller</td>
<td>ads.mopub.com</td>
<td>AS63179</td>
<td>MoPub</td>
</tr>
</tbody>
</table>

Figure 5.7: The periodic phoning home every two hours by the Truecaller Application.
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>App</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-03-16</td>
<td>13:37:31</td>
<td>Ryanair</td>
<td>settings.crashlytics.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>13:37:31</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>13:37:32</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>14:29:55</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>14:29:55</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>14:30:10</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>15:30:03</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>15:30:05</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>15:30:20</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>16:30:07</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>16:30:07</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>16:30:07</td>
<td>Ryanair</td>
<td>settings.crashlytics.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>16:30:22</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>16:34:33</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>16:34:31</td>
<td>Ryanair</td>
<td>graph.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>17:28:39</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-16</td>
<td>17:28:39</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>12:38:22</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>12:38:24</td>
<td>Ryanair</td>
<td>settings.crashlytics.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>12:38:24</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>12:38:26</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>13:32:12</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>13:32:13</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>16:21:55</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-17</td>
<td>16:21:59</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>11:36:13</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>11:36:13</td>
<td>Ryanair</td>
<td>settings.crashlytics.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>11:36:14</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>11:36:14</td>
<td>Ryanair</td>
<td>settings.crashlytics.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>11:36:16</td>
<td>Ryanair</td>
<td>settings.crashlytics.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>11:36:24</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>12:27:22</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>12:27:22</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>12:27:37</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>19:00:45</td>
<td>Ryanair</td>
<td>settings.crashlytics.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>19:00:45</td>
<td>Ryanair</td>
<td>api.facebook.com</td>
</tr>
<tr>
<td>2019-03-18</td>
<td>19:00:45</td>
<td>Ryanair</td>
<td>6178.eu-api.swrve.com</td>
</tr>
</tbody>
</table>

Figure 5.8: The calls made over a number of days by the Ryanair Application.
Grafana and InfluxDB

With the sheer volume of data, going line by line was not an option. A graphical representation of the data would be required to easily observe the entries. As this data was in time-series format the use of a dedicated time-series database would be more effective in using the data. InfluxDB is the database that was chosen due to its ease of use and interoperability with the time-series graphing tool Grafana.

A number of Grafana dashboards and charts were created to analyze the traffic. These included graphs on connection lengths, protocols, geography and timings. The dashboard allowed a more high-level analysis of the application traffic over a longer period of time. These dashboards facilitated the visual anomaly detection of the data. They also provided the functionality to create complex maps, such as Figure 5.13 without difficulty.
5.3 Findings

From the data that was analyzed, there were a lot of interesting findings. In total over 350,000 network connections were analyzed over a 2 week period. This content related to 34 applications and over 4,000 unique website URLs. Chrome made the most external connections with just under 3,000 unique websites visited. In comparison Leap Card Top-Up connected to just two URLs.

It was clear that the majority of connections that were made from web browsers such as Chrome and Opera were to trackers and ad-related content. This wasn’t surprising as this kind of traffic can be seen in everyday browsing through a desktop machine. The Facebook application also saw a high number of tracker and ad-based connections. This was due to the fact that Facebook contains an inbuilt web browser which loads external websites from within the application.
graph.facebook.com was a common connection amongst 10 of the applications which were found on all of the devices. This connection is a call to Facebook’s ad-based RESTful API. It allows the application to link in with your advertising profile to generate ad-related content within the application.

A number of applications engaged in the process of "phoning home". The applications which engaged in this process often had very little traffic other than these connections, indicating that these applications have long-lived process which starts up periodically to send traffic back to a central server about the device.

Most applications monitored used port 443 for network communications apart from Spotify which regularly used port 80 for communications. Spotify also used a number of ports that are not in everyday use such as 57620. WhatsApp and Youtube also used high port numbers. It is likely that these connections were for long-lived streaming processes which would not interfere with normal web browsing over port 443.

Only 17.7% of network connections were made using an IPV6 address with the majority of traffic still being sent to IPV4 addresses (82.3%). Connections were made to servers in 29 different countries with data being sent outside the EU mainly consisting of US-based connections. Traffic was also sent to Russia, China, Brazil, and Japan.

5.4 Conclusion

The survey carried out provided greater detailed insight into the application traffic that was generated on mobile devices. It provided a platform to begin to analyze the traffic for
anomalous connections that were generated. A number of interesting insights were also gained through data fusion to add extra pieces of information to the network connections. Adding the server that the destination URL was hosted on and adding the location of the IP increased the usefulness of the data.

The survey provided an interesting analysis of people’s phone usage habits. The periodic nature of network connections from devices gives an idea of how much of the day people spend on their device. This highlights the importance of the work in ensuring that the device, and its applications, are secure and not being used with malicious intent.

The survey, however, raised a lot more questions than it answered. While the connections themselves were interesting, without viewing the content of the connections it is hard to decide if the connections are actually malicious. Due to the nature of the study, it was also hard to derive the context that network connections were made i.e. what in-application functions were being carried out.

For these reasons, an in-depth analysis of a selection of the applications will be carried out in Chapter 6.
6 Application Analysis

6.1 Introduction

This chapter will detail the steps that were carried out in an attempt to understand the application network traffic flows which were flagged in Chapter 5 as anomalous. To acquire this understanding, a deeper look at the content of the connections made by an application is required. To do this, a device will be rooted for the applications traffic to be decrypted, allowing the connection content to be viewed in plain-text.

6.2 Application Selection

There were a number of selection criteria for selecting the applications for further analysis. The application:

- must have appeared on all of the devices logged traffic.
- must have recorded at least one anomalous connection sequence or event.
- must have over 20,000 downloads in the play store.
- should contain some sort of sensitive data which a user would not approve sharing.

In total, 16 applications matched this criterion and were analyzed in detail:

- BOI Mobile
- KBC Mobile
- AIB Mobile
- Revolut
- Permanent TSB
- Ulster Bank
- Airbnb
- Paddy Power
- Candy Crush Saga
- Spotify
- WhatsApp
- JustEat.ie
- Ryanair
- Boots
- Booking.com
6.3 Device Preparation

The analysis of each application was carried out in a controlled manner and repeated at least twice on the controlled device. The device chosen to test the applications was a Google Nexus 5 running Android Marshmallow 6.0. This phone had a number of modifications to allow for the analysis to be carried out.

6.3.1 Rooting

The first step in the preparation of the device was to root the phone. Rooting is a process which allows the user to gain root access to the Android operating system. It grants the user "super user" or privileged access to the operating system. Rooting is not allowed by almost all manufacturers of devices, in doing so voids the warranty of these devices.

To root the device, the boot-loader was initially unlocked and a custom ROM was flashed onto the device using the Android debugging tools. The custom ROM allows for the Xposed framework to be installed on the device. Xposed is similar to Google Play Services on Android devices, except works as a packages manager for downloaded "modules" or packages from the internet.

6.3.2 Man-In-The-Middle

A man-in-the-middle attack required a third party to be listening on the network. The device was connected to a third-party proxy, Fiddler, via a WiFi connection.

The security X509 certificate of Fiddler was installed onto the device for use in SSL tunneling. The traffic routed through the proxy could then be decrypted using the private key of the proxy, making the contents of the SSL connection visible in plain-text. Fiddler’s UI allows for the capture and sequencing of network flows to fingerprint application’s network activity.

6.3.3 SSL Hijacking

Over 99% of the network connections relating to the applications to be analyzed were in an encrypted HTTPS format. As outlined in Section 2.2.1, a standard man-in-the-middle style attack fails on HTTPS traffic. As such, SSL hijacking of the connection is required. SSL hijacking involves using a third parties X509 certificate to encrypt network communication. However, to prevent this some applications use SSL Pinning.
SSL Pinning is the process of associating a hostname with their expected X509 certificate or public key. Once a certificate or public key is known or seen for a host, the certificate or public key is associated or 'pinned' to the host (32). This mechanism prevents a third party act as the host, as their certificate will not match the certificate which is required by the application. SSL unpinning skips this pinning check to allow for custom certificates be used in the encryption process.

SSL Unpinning was achieved through the use of an Xposed Module "SSLUnpinning 2.0". This module works by bypassing the certificate checks within the application itself. It allowed for a custom third-party certificate to be validated within the application and to be used as part of the SSL tunnel encryption.

6.4 Analysis Method

A formal analysis of the applications selected required a structured approach. Applications may cause third-party services such as Google Play Services to generate activity so being able to observe this behavior is important.

To achieve a trustworthy analysis a number of precautions and steps were taken.

- Each analysis session would begin when the application had been downloaded and end when the application was stopped.
- Only one application would be installed on the device during each analysis session.
- Any network traffic recorded, which could not be conclusively related back to the application, would not be recorded.
- Timings would be recorded to sequence network flows within the application.
- Each "in-application action" would be carried out leaving a 1-minute gap between the last observed network connection.
- Each analysis session of an application would be repeated at least twice to ensure fairness.

These steps aim to standardize the process across all applications. It is not possible to fully standardize interactions within the application as each application has different on-boarding and interaction flows. Connections relating to Google applications and services were disregarded in the analysis as these connections are related to the Android operating system (unless otherwise stated).
6.5 Analysed Applications

The applications selected to be analyzed were largely defined in one of two groups: Banking applications and Utility applications. Applications have to comply with EU laws in how data is used, specifically GDPR compliance. Failure to comply with GDPR guidelines can result in massive fines of €20 Million or 4% of annual turnover, whichever is greater.

Specific analysis of whether applications adhere to these guidelines alongside any other anomalous or malicious will be examined. The use of trackers and ad-ware will be expected by most application once the user has agreed to the Terms & Conditions and Privacy Policy of the application. Their presence itself is not noteworthy, however, the information they transmit will be examined to ensure compliance with the Policies agreed to.

6.5.1 Banking Applications

Banking applications are one of the most common ways to carry out day-to-day banking activities. There rise in popularity is evident with many banks now switching to an application focused Eco-system (N26 and Revolut), while other banks have invested in driving growth through there application (KBC Bank). As the popularity of these types of applications continue to rise a greater focus will be placed on their security.

Bank Of Ireland

Bank of Ireland is one of Ireland’s largest banks and as such, it is one of the most downloaded banking applications in Ireland. From the data that was gathered, there were a number of anomalies that were detected. The application regularly sent connection requests to an "extreme-push.com" website located in the US.

On start-up, a number of connection requests were made by the application. An interesting connection made on the initial launch is to data.altbeaon.com, a proximity-based advertising protocol. The devices Android ID is transmitted to this website.

When logging into the application a Session-ID token is created between the application and web service. This token is used throughout the session until the application is closed. Log in information is sent to Bank of Ireland in plain-text format. This includes your Unique Banking number and 3 digits of your 6 digit pin.
Each time the application is restarted, the application requires the user to log in again. However, it does not require the user to re-enter their 8 digit banking code. This is sent to the application from the remote server based on a combination of the device ID and phone information.

Within the application, the user is able to transfer money between accounts. This is done through selecting the payee, entering an amount and re-authenticating within the application. Once authenticated the transaction goes through. This transaction, however, does not contain a unique session key tied to the transaction and thus the network packet can be replayed to re-transmit the transaction.

KBC

KBC is a relatively new bank in Ireland. It is an online-only bank with a web and mobile presence only. On initial opening of the application, no unusual behavior is recorded. A screen is presented which prompts the user to log into their account. The privacy policy of the website is also linked here. When the privacy policy of the page is loaded, a significant amount of third party ad-ware and tracking websites are loaded. These include "ads.LinkedIn.com", "liveperson.net" and "hotjar.com".

The loading of these third-party services is in violation of GDPR guidelines. These connections send information about the user’s device to third parties. The privacy policy states that KBC only transfers information to trusted third party locations as outlined by the EU. On examination of hotjar.com, this tracker connects to IP addresses located in China. A full reference to these connections can be seen in Appendix A4.
Permanent TSB

Permanent TSB’s application is an HTML web viewer of their online web-page. Its content is served through the SWRVE Content Delivery Network. No obvious anomalies in the application mobile connections were found. Passwords for logging into accounts were sent in clear-text through an SSL tunnel, however (shown in figure 6.4). This is a low-risk security vulnerability (it is transmitted in an SSL tunnel) as the password should be hashed on the device prior to transport.

Revolut

Revolut is an application only bank with no physical presence in Ireland. It has grown to have over 200,000 Irish users amongst its 4 million customers around Europe. On starting the application a number of network anomalies occur. graph.facebook.com is contacted immediately on start-up. This connection sends information to Facebook about the user’s advertising id and application that was installed which can be seen in 6.5. This
Figure 6.3: Request which was replayed resulting in the transfer of funds being carried out twice.

Figure 6.4: Login attempt to the Permanent TSB application.

connection without the user’s consent is a violation of GDPR. The full sequence of events can be viewed in Appendix A5

Other Applications

Alongside the above applications, other banking applications tested were Ulster Bank and AIB Mobile Application. These applications did not work with a rooted device and as such were not tested.
6.5.2 Utility Applications

Alongside these banking applications, a number of utility based applications were also examined. These applications included betting, travel, and food-related applications. Each of the application contained some form of sensitive information that would be private to the user. These applications are also some of the most downloaded applications in the Google Play Store.

Paddy Power

Paddy Power is the largest online betting agency in Ireland. It was selected to be analyzed due to a number of connections to ad agencies. On start-up, the application connects to graph.facebook.com sending the users advertising information. Device and user information is then send to appsflyer.com, a marketing and analytics agency.

Information is also sent to urbanairship.com, another marketing agency. Connections continue to communicate with Urbanairship on the close of the application. Information on device location as well as where the application is running is sent. Interestingly a flag in the channel "opt-in" is set to true, despite no acceptance of any Terms & Conditions or Privacy Policy.

The information is all sent prior to any interaction with the application i.e. prior to log-in or consent for information to be shared. On first inspections, the application does not move beyond the initial loading screen as the application has detected it is installed on a rooted device.

JustEat.ie

JustEat is an online food delivery service which operates through a mobile application and website primarily. On the device start-up, a connection request is sent to graph.facebook.com containing information relating to the device and user. Information
about the device is sent to appdynamics, application performance monitoring. The data sent to this service contains information relating to the device.

Airbnb

Airbnb is a room sharing application. On startup of the application, a number of connections are created. Graph.facebook.com, accountkit.com, and branch.io are all connected to immediately. Device and user information was sent to these connections. Interestingly, the website mparticle.com was connected to as well. This service is used for streamlining customer data. The data sent to this service included the device id, amongst other sensitive data. What makes this connection interesting is that, in the initial call, the service connects with urbanairship, a service previously seen in the Paddy Power application. In the subsequent connection, the device ID is sent to the service, and the identities between the two applications are identified. This is an example where collusion between applications could occur as seen in state-of-the-art research.
Ryanair

Ryanair is a low-cost airline which allows customers to book flights through their application. The application is served via the SWRVE service which serves the web pages from their content delivery network. The application was flagged for further analysis due to the significant number of connections that were created in quick succession to the sync.ryanair.com website.

On examination of these network connections, there are a series of requests which contain the exact same pieces of information. No personal data is transferred, just a sequence number. These repeated connections occur continuously when the application is opened on the device, and stop when the application is closed. The possibility is that these connections are a naive implementation of a method to examine the screen-time of when the application is active.

Similar to other applications, on start-up, Ryanair sends information to graph.facebook.com. As well as this it sends information regarding the device to OMTRDC.com, a marketing tracker from Adobe.

Boots

Boots is a pharmacy who's mobile application can be used to purchase items as well as recording transactions which have taken place within the physical store. On startup of the
Figure 6.8: The significant amount of quick succession network requests made by the Ryanair Application.
application, the application connects to graph.facebook.com. The application also connects to a number of other tracking services including urbanairship.com and omtrdc.com. However, unlike other applications which use these services, no personal device information is sent which can uniquely identify the device.

Other Applications

Other applications which presented some anomalous behavior in the data collection included Candy Crush Saga and Spotify. On further examination of these applications, no untoward activity was observed. While Candy Crush did create a significant number of calls to graph.facebook.com alongside a number of other tracking and ad sites, this was clearly stated in the Terms of Service of the application. No traffic was generated prior to the Terms of Service being agreed. Spotify’s use of high port numbers was for long-lived music streaming connections.

6.6 Findings

From the analysis, a number of interesting results were found:

- 6 of the 16 applications were found to be transmitting user data to third-party services or websites without their consent. The main culprit was with the graph.facebook.com service. This service works as a RESTful API for the transfer of information into and out of Facebook. As part an extension of the study, all applications were tested with the graph.facebook.com URL blocked - no adverse impact was noted. This indicates that call to the API was for non-functional requirements of the application, such as the delivery of ad content.

- Bank of Ireland’s mobile application contains a number of security vulnerabilities that severely compromise the security of the application.

- Passwords and other sensitive information is regularly transferred in plain-text within SSL tunnels. These pieces of information should be at least hashed to provide an extra layer of security before being tunneled.

- Ryanair’s application sends an unusually large number of connection requests while their application is open. While the connection sends very little information, the volume of connection requests could be considered malicious.

- When applications are un-installed it is unclear if your data is removed from third-party tracker websites. In the case of Urbanairship, the phone ID, which does not change, was sent to the website when using the Paddy Power application. The
Paddy-Power application was deleted from the device while testing the Airbnb application, however, when the ID was sent to the urbanairship service, it resulted in a "matched identity" response from the service. The concept of collusion was discussed in Section 3.2.2 between applications, this collusion relates to collusion between third-party ad-services.

6.7 Conclusion

From the analysis of the applications, a greater insight was achieved on the content of the application connections as seen in Chapter 5. Some of the traffic which had been flagged as anomalous, such as the high port number usage, turned out to be benign and part of the normal usage of the application. Other applications traffic, however, did contain sensitive information which was sent to third parties.

A number of serious security flaws were also observed in some of the banking applications which were analyzed. These types of security flaws should have been flagged in the penetration testing of the application before it was released.

From the analysis, a number of questions were raised. One of which was why the findings which were identified in this dissertation were not flagged before the release of the application. One possibility for them not to be flagged would be if the testing of the application was carried out on an emulated device, rather than a real device. Emulated devices are much more popular for testing, so the functionality of the application could be possibly changed on an emulator in an effort to mask the behavior that was identified in this chapter.
7 Emulator vs. Real Device

This chapter will examine the network activity of the applications on an emulated device. A comparison of the network traffic generated between a physical device and emulated device will also be undertaken.

7.1 Introduction

Emulators are a key part of the development of a mobile application. They allow for the rapid testing and prototyping of new features and bug fixes in a controlled environment. Emulators can be used to mimic the hardware constraints of all different types of devices, without having to purchase the device. Emulators give access in greater detail to the internal working of the application also.

A comparison between emulators and real devices was undertaken to determine if application behavior was modified in an emulated environment. From Chapter 6, a number of network activities which resulted in personal information being sent to third parties were uncovered. This section will investigate whether the same network activity is viewed on emulated hardware.

A positive result in this analysis would be to see no difference in traffic between the emulated device and real device. This would indicate that the tests carried out in Chapter 6 were representative of normal application behavior. If a difference is observed, this would indicate that the developers of the application may be trying to hide something.

The applications analyzed included:

- BOI Mobile
- KBC Mobile
- Revolut
- Airbnb
- JustEat.ie
- Ryanair
- Boots
- Booking.com
7.2 Setup

7.2.1 Emulator

When setting up the emulator, a number of considerations were taken into account to ensure that application traffic would not be influenced by external factors. The Android version that was selected was the same as that on the physical mobile device (Android 6.0). The applications that were installed on the emulated device were also the same as on the physical device.

The emulator was set-up using both a rooted version of the Android operating system and the non-rooted version. To capture the traffic coming from the emulated device, once again a third party proxy was used to perform a man in the middle attack.

For the analysis, two different emulators were used. The first emulator was integrated as part of the Android Studio suite, which is used for the development of applications. The second emulator was a combination of Oracle’s VM Virtual Box software, a virtual machine emulator. This emulator was running the rooted version of Android. The Android 6 operating system was used on both virtual devices. This virtualized environment allows the user full control of all of the device functions from network activity to storage.

7.2.2 Man-In-The-Middle Attack

Setting up the third party to intercept the traffic on the emulated device was a more involved process that on the physical device. The traffic was routed through firstly a packet capturing software "Wireshark" which records all of the network packets in fine granularity. This routed traffic was recorded and sent via PCAP file to Fiddler, the HTTP proxy that had been used in analyzing the physical devices network traffic.

7.2.3 Evaluation Method

The area of comparing network connection flows between devices is a niche area and as such, there is very limited information available as a reference. To combat this a number of experiments were carried out to gather data and baselines for the variations that could be expected between connections, and to find the best ways to represent the data.

To find a suitable variation figure for different TCP connections, a connection was established to a web server repeatedly and a graph was created to find the size distribution of the packets as seen in Appendix A6. The data was calculated to be within
+- 0.11% of the mode of the data. Therefore a margin of 0.11% was allowed per connection in the difference in network flows (e.g. an additive fault tolerance across 10 network connections would result in an error margin of 1.1%). A point graph was chosen to represent the data packet size and sequence.

To compare network flows a "line of best fit" will be drawn from the connections point graph. The slope of this line will then be taken and used in the formula

$$\tan(\theta) = \frac{m_1 - m_2}{1 + m_1 m_2}$$

where $\theta$ will be the angle between the lines of best fit. An error margin will be calculated as

$$(\text{number of connections}) \times 0.11$$

7.3 Application Analysis

In analyzing the applications, the same process that occurred on the physical device occurred on the emulated device. This ensured that no sequencing issues would influence the results. Similar to the physical device, all applications were downloaded from the Google Play Store on the device where possible. If the device was not able to be downloaded from the play store, the application APK would be installed onto the device using the ADB command line tools. Only one application was installed on the device at each time.

7.3.1 Analysis Limitations

Some applications are not available on the Android application store for emulated devices. This can be for a variety of reasons as specified by the application developer, such as the requirement of some internal phone hardware for the production of random numbers for cryptography or simply the device "hardware" is not in the list of supported application for the device.

For this reason, a number of the applications were not able to be tested fully i.e. could not be loaded onto the emulator provided by Android Studio. Only applications which were available to be tested on the 3 devices are deemed to have been covered in this study.
7.3.2 Analysis

The applications that were tested, were all tested across all 3 devices. As not all devices were rooted, the analysis was limited to viewing the initial connection of the TCP handshake. While this does not provide the same level of detail as the unencrypted packet, it does provide a mechanism to see where the device connects to.

In the applications which were tested, there was little deviation in network activity observed. Some differences were observed in the number of connections, and the size of the packets transferred. However, the differences can be explained with bad connection issues, and with differences in the time-stamp or device ID causing a variation in packet size. Below is an example of the observed differences in TCP connections established between an emulated device and a real device.

![Figure 7.1: Difference in the connection TCP Connection Requests being created by the Just-Eat Application on start-up was minimal.](image)

Further analysis of network connections was carried out by graphing the data and producing a line of best fit. The slope of this line was then used to find the difference in a sequence of network connections over time with Formula 1. If the value derived for $\theta$ was greater than the threshold value, as per Formula 2 then the connections were deemed different.

As can be seen in Figures 7.2 and 7.3 the shape data in the traffic which is sent across the network is similar. The Figures also contain a line of best fit for the graph, which was used to calculate the difference between the connections. The slope of each line, $m_1$ and $m_2$, was used in formula (1) to calculate the distance between the network flows. The lines were deemed to be within a reasonable fault tolerance of each other if the angles between the lines were within 7.2° threshold, (2% of 360°, where Formula (2) was used to
calculate the 2% fault tolerance). In Figure 7.2, the slope is -2.4 for the line of best fit, and in Figure 7.3, the slope is -3.0. By using the formula (1), a value for $\theta$ of 4.76°, was observed, a value less than the threshold.

![Figure 7.2: A graph of the connections made by the device over a 10 second period.](image)

![Figure 7.3: A graph of the connections made by the emulator over a 10 second period.](image)

### 7.4 Findings

Throughout the process there was little deviation in network traffic flows across all of the devices, both emulated and physical. This finding is important for a number of reasons. It shows that the findings generated from the initial application analysis can be, and have been, replicated across multiple devices, showing that they were not one-off occurrences.
It is also interesting to note that the companies who own these applications, would have been able to view the network anomalies in their own emulated test environments. Meaning either they didn’t test the application fully for the sequence of when information is sent to third parties when an application is loaded, or they knew the sequence and are actively allowing for the transfer of information.

Another interesting finding was that some applications were not available to be downloaded on the emulator supplied by Android Studio. There are a number of reasons that could be the cause of this, such as the application needing some real physical device infrastructure to perform in applications actions, some glitches may occur when running on the emulator and as such application download is not available. Another reason may be to act as an additional security feature of the application, by not supporting an emulated environment, the attack surface of the application is reduced.

As can be seen in Figure 7.4, some applications sent information on whether the device running the application was real or not. However, no discernible difference was observed when this flag was set to false.

![Image](image-url)

Figure 7.4: An initial request generated by the Tinder application to a third-party tracking website.

### 7.5 Conclusion

Overall, there were very little differences in the traffic observed across the devices. This suggests that, in the tested applications, developers are not performing any checks to see if the environment is emulated and alter the functioning of the application.

The same network connections to third-party services with the same level of sensitive content were also observed. This would indicate that either the developers never checked to see what content was being sent to third parties, or they didn’t care sensitive information was being exposed.
8 Conclusion

This section will reflect on the work carried out throughout the dissertation, and outline a summary of the conclusions as a whole. A discussion on the limitations of the research and the future work that may be carried out will also be included.

8.1 Overview

This dissertation discussed the ability to observe anomalies in mobile phone traffic. This was achieved through the implementation of a participatory study on people’s mobile phone traffic. This was carried out by installing an Android application on their devices, which logged the network connections and sent them to a remote server.

Once the study had been completed, over 300,000 network connections were analyzed. Initially, simple graphs were made to begin to understand the content of the data, graphs on popular applications and URLs were created. Once an understanding was achieved, a more in-depth analysis of the traffic was carried out. Firstly, applications were clustered based on commonality of URLs connected to. Then, with Grafana Dashboards, an analysis of the data over time took place. With this, anomalies in the data were identified. A number of anomalies, including obscure port usage, uncommon data transfer sizes, and the phoning home of applications were observed. Other interesting observations such as what countries applications connected to were also observed.

With the application’s data analyzed over a period of time, an in-depth analysis of a number of the most common applications which displayed anomalies was carried out. This involved the rooting of an Android device, to bypass SSL security, and decrypt packets transferred to observe the content. Findings from this study included a number of GDPR violations and serious security flaws in a number of the applications.

Finally, an examination of the difference in operation between emulators and real devices was carried out. Through the research carried out on applications in the analysis section, the issue of functionality differences in applications in emulators was discussed. This led to an analysis of some application across a number of devices and emulators. No
significant or interesting differences in network connections were observed between the emulator and real device. The fact that some applications didn’t run on emulated hardware was of note though.

The main research objective of this project was to carry out a comprehensive analysis of applications and the network traffic they produce. This was achieved through the process of monitoring the application over a number of weeks, carrying out a visualization of the traffic produced by the application, identifying if any interesting network connections were made by the application, looking into the secure SSL tunnel at the content being transferred between applications and external servers and analyzing the behavior of the application on emulated hardware. The process outlined in the dissertation is one which can be replicated across applications to provide consistent and fair results.

8.2 Reporting of Results

The results which were observed in Chapter 6, included a number of findings which the researcher deemed as in breach of GDPR rules. A report which contained the findings, and how to replicate the breaches, was produced and sent to the Data Protection Commissioner for further analysis.

As well as the suspected GDPR breaches which were identified, a number of security vulnerabilities were also identified. These were reported to the companies involved for further action.

8.3 Future Work

There are a number of improvements and extensions of the project which could be carried out to extend the results of this dissertation.

An extension of the participatory survey with more participants over a longer period of time would greatly improve the variety of data to be analyzed. Due to delays in obtaining ethical approval for the survey to begin, reduced group size was used over a shorter period of time. This extension of the survey would also mean that more data on different applications would be gathered. With more data, a machine learning model would be able to be trained to observe the statistical properties of application data over a longer period of time.

This additional data could be used to further the analyses of applications. With more information on the functionality of applications, the anomalies in applications would be
clearer to identify, and thus reduces the amount of time required to replicate the anomaly.

Some of the most noteworthy findings of the study were about the GDPR violations surrounding the applications which were tested. Clearly, the regulations are not being adhered to by a significant number of applications. A method for automating the detection of GDPR violations across more applications would be a meaningful extension of the work.

Testing applications on more rooted devices running different versions of Android would be an interesting extension of the project. Also extending the project to include the iOS operating system would be a beneficial addition. Comparing the network connection content of the two Operating System and what data is sent from the device would be a beneficial extension of the project.
Bibliography


This study aims to provide people with better visibility of the activity of the applications installed on their phones. It will do this by logging the network traffic generated by the phone.

To participate in the study, you will install a small logging application on your phone.

Participation is voluntary, and participants have the right to withdraw from the study at any time without penalty. You can do this by simply uninstalling the study’s logging application.

The expected duration of the participants’ involvement is to record data for at least 1 week and up to 6 weeks, with the study finishing at the end of April 2019.

There are no anticipated risks to participants. All the data collected will be transmitted and stored in encrypted format in a way that cannot be related back to specific end users.

All data storage and processing will comply with Irish and EU data protection rules. The findings of the study are expected to be published in one or more academic papers, but we will take care to protect the anonymity of all participants.

At the end of the study participants should delete the logging app, although in any case this will be disabled once the study ends. No other debriefing is planned, although participants will be able to see the study outcomes when published.

Illicit activities will be reported to appropriate authorities.
A2 Ethics Approval - Informed Consent

TRINITY COLLEGE DUBLIN
INFORMED CONSENT FORM

LEAD RESEARCHER: Joseph Fitzpatrick

BACKGROUND OF RESEARCH: This project aims to carry out a measurement study to record actual mobile phone network activity with a view to making this more visible to users (via a dashboard) and highlighting anomalies/potentially interesting activity.

PROCEDURES OF THIS STUDY: The measurement study will involve users installing an application on their phones. This app will use the phones VPN API to log network activity and periodically upload this log to a server for storage. The logging will last for at least 1 week (and up to 4 weeks). All data will be anonymized, and the handset will be non-identifiable from the logged data.

PUBLICATION: Information gathered as part of the research will be used as part of my Dissertation.

Individual results may be aggregated anonymously and research reported on aggregate results.

DECLARATION:

- I am 18 years or older and am competent to provide consent.
- I have read, or had read to me, a document providing information about this research and this consent form. I have had the opportunity to ask questions and all my questions have been answered to my satisfaction and understand the description of the research that is being provided to me.
- I agree that my data is used for scientific purposes and
- I have no objection that my data is published in scientific publications in a way that
does not reveal my identity.

- I understand that if I make illicit activities known, these will be reported to appropriate authorities.

- I understand that I may stop electronic recordings at any time, and that I may at any time, even subsequent to my participation have such recordings destroyed (except in situations such as above).

- I understand that, subject to the constraints above, no recordings will be replayed in any public forum or made available to any audience other than the current researchers/research team.

- I freely and voluntarily agree to be part of this research study, though without prejudice to my legal and ethical rights.

- I understand that I may refuse to answer any question and that I may withdraw at any time without penalty.

- I understand that my participation is fully anonymous and that no personal details about me will be recorded.

- I have received a copy of this agreement.

PARTICIPANT’S NAME: PARTICIPANT’S SIGNATURE:

Date:

Statement of investigators responsibility: I have explained the nature and purpose of this research study, the procedures to be undertaken and any risks that may be involved. I have offered to answer any questions and fully answered such questions. I believe that the participant understands my explanation and has freely given informed consent.
A3 Data Sample

Figure A3.1: A sample of the data which was collected.
Figure A4.1: Page 1 of 3. Connections made by the KBC Privacy Policy Webpage.
Figure A4.2: Page 2 of 3. Connections made by the KBC Privacy Policy Webpage.
Figure A4.3: Page 1 of 3. Connections made by the KBC Privacy Policy Webpage.
Figure A5.1: The network connections generated by the Revolut mobile application.
A6 Packet Size Variations

Packet Size over 30 requests

Figure A6.1: The variation in packet sizes over 30 transmissions.