

Network Forensic Analysis of Secure Instant Messaging Application: A Case Study of Signal App



By

Asmara Afzal

Fall 2017-MS(IS) - 00000203097

Supervisor

Dr. Mehdi Hussain

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree
of Masters of Science in Information Security (MS IS)

In

School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(December 2020)

Approval

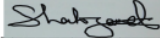
It is certified that the contents and form of the thesis entitled "Network Forensic Analysis of Secure Instant Messaging Application: A Case Study of Signal App" submitted by Asmara Afzal have been found satisfactory for the requirement of the degree

Advisor : Dr. Mehdi Hussain

Signature:  _____

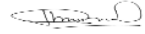
Date: 04-Dec-2020

Committee Member 1:Dr. Shahzad Saleem

Signature:  _____


Date: 08-Dec-2020

Committee Member 2:Dr. Arsalan Ahmad

Signature:  _____

Date: 04-Dec-2020

Committee Member 3:Dr. Sana Qadir

Signature:  _____

Date: 03-Dec-2020

Thesis Acceptance Certificate

Certified that final copy of MS/MPhil thesis entitled "Network Forensic Analysis of Secure Instant Messaging Application: A Case Study of Signal App" written by Asmara Afzal, (Registration No 00000203097), of SEecs has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____ 

Name of Advisor: Dr. Mehdi Hussain

Date: _____ **04-Dec-2020**

Signature (HOD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

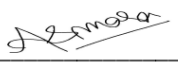
Dedication

I dedicate this thesis to my *Parents* and *Siblings* for their endless prayers, love and encouragement.

Certificate Of Originality

I hereby declare that this submission titled "Network Forensic Analysis of Secure Instant Messaging Application: A Case Study of Signal App" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEecs or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEecs or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: Asmara Afzal

Student Signature: 

Acknowledgment

First of all, I would like to thank Allah, the Almighty for giving me the ability and strength to carry out this research.

My deepest gratitude to my advisor Dr. Mehdi Hussain for his continuous support and guidance during my thesis. I could not have imagined having a better advisor and mentor for my master's degree. I am also thankful to my teachers for providing me with an academic base, which enabled me to complete this thesis.

I am thankful to all my fellows and friends for their support and motivation. Last but not the least, I would like to thank my parents for their endless prayers and support throughout.

Table of Contents

1	Introduction	1
1.1	Overview	1
1.2	Problem Statement	5
1.3	Solution Definition/Description	6
1.4	Thesis Motivation	7
1.5	Thesis Contribution	7
1.6	Benefits of the Scheme	8
1.7	Thesis Organization	8
1.8	Summary	9
2	Literature Review	10
2.1	Literature Review	10
2.1.1	Device Forensics	12
2.1.2	Network Forensics	15
2.1.3	Automated Tools and Frameworks	18
2.2	Summary	20
3	Research Methodology	22

3.1	Introduction	22
3.2	Research Types	23
3.2.1	Descriptive vs Analytical Research:	23
3.2.2	Fundamental vs Applied Research:	24
3.2.3	Quantitative vs Qualitative Research:	24
3.2.4	Conceptual vs Empirical Research:	25
3.3	Research Methodology Overview	25
3.4	Thesis Research Methodology	26
3.4.1	Define a Research Area	27
3.4.2	Literature Survey	29
3.4.3	Research Method	29
3.4.4	Data collection	30
3.4.5	Analysis and Interpretation	31
3.4.6	Presentation and Results/Findings	32
3.5	Summary	32
4	Experimental Setup	33
4.1	Overview	33
4.1.1	Pfsense Firewall Configuration	33
4.1.2	Access Point, Wireshark and Mobile device	34
4.2	Applying firewall rules	35
4.3	Justification for the need of a Firewall?	35
4.3.1	Setup for the smartphone	36
4.4	Summary	37

5	Results and Analysis	38
5.1	Overview	38
5.2	Ports and Servers Ranges	39
5.3	Validation and verification	40
5.4	Identifying Signal’s Traffic	40
5.4.1	Accessing/Opening the Signal App	41
5.4.2	Target Device (User A Typing) Pattern	43
5.4.3	User B Typing Patterns	45
5.4.4	Call initiated by User A (Caller)	45
5.4.5	Call Received by User A (Callee)	48
5.4.6	Media Messages	49
5.4.7	Video Calls	51
5.5	Summary	54
6	Validation and Verification	55
6.1	Blocking List of Servers	55
6.2	Blocking Chat Servers	57
6.3	List of Call Servers	59
6.4	Crime Scene Reconstruction	61
6.5	A Case Study	61
6.6	Case study:Data leakage using Media messages	62
6.7	Summary	64
7	Conclusion & Future Work	65
7.1	Conclusion	65
7.2	Limitation & Future Work	66

7.3 Summary	67
Bibliography	68

List of Figures

1.1 Overall Flow of Activities	5
2.1 Branches of Digital Forensics	11
3.1 Research Types	23
3.2 Research cycle with NIST Guidelines	30
3.3 Framework Based on NIST Guidelines	31
4.1 Experimental Setup	34
4.2 IP address of smartphones Traffic passing through pfSense . .	36
4.3 IP address of the Smartphone	37
5.1 Authentication Phase	42
5.2 Signal Application Opened	43
5.3 User A typing a message	44
5.4 User B is typing a message	46
5.5 STUN and TURN servers	47
5.6 Receiver's IP address	48
5.7 Patterns: User A calls to user B	48
5.8 Patterns: Call Terminated	49

5.9	Target User A gets a call from user B	49
5.10	UDP packets flow of a call	50
5.11	Media Messages	51
5.12	Video calls UDPatterns	51
6.1	DNS query textsecure-serrvice.whispersystems.org	56
6.2	Type A, Server Addresses	56
6.3	Message sending Failed	58
6.4	Response from blocked servers	59
6.5	Wireshark-based Graph of blocked servers	60
6.6	DNS Query and Response	63
6.7	Media messages patterns are detected	63
6.8	Temporal Analysis	63

List of Tables

2.1	Comparison of recent approaches	20
5.1	Identified Chat Servers	39
5.2	Traffic Characteristics of Signal Application on Android Device	53
6.1	Firewall Rules	59
6.2	Call Servers and Receivers IP Found – Needs to be Blocked . .	60

Abstract

Instant Messaging applications (apps) have played a vital role in online interaction, especially during COVID-19 lockdowns. Apps with security provisions are able to provide confidentiality through end-to-end encryption. Ill motivated individuals and groups use these security services to their advantage thereby using the apps for crimes of various nature. During an investigation, the provision of end-to-end encryption in apps increases the complexity for digital forensics investigators. This study aims to provide a network forensic strategy to identify the potential artifacts from encrypted network traffic of a prominent social messenger app *Signal* on android version 9. The analysis of the installed app has been done over fully encrypted network traffic. By adopting the proposed strategy one can easily detect encrypted traffic of chat, media messages, audio, video calls by looking at the payload patterns. Detailed analysis of the trace files helped to create list of chat servers, IP addresses of involved parties in the events. Analysis of the presented forensic analysis app is applicable to android mobile devices.

Chapter 1

Introduction

Chapter 1 elaborates the overview of basic concepts, significance and history of research work. This chapter describes the road map of thesis and briefly highlights the further organization and structure of the thesis. This chapter explains the motivation for carrying out the research work. This chapter also gives idea about the vital contributions, prominent benefits, scope of the work and key objectives of the thesis.

1.1 Overview

Social Media and Instant Messaging applications are in huge number of uses now a days. Users of different age groups use them on daily basis for the personal and business purposes. A significant increase is found during Covid-19 pandemic. Businesses, education, healthcare and marketing sectors has been using these applications for the continuity of the services. Malicious activities are also taking advantage of these applications as the security of these

applications is enhancing day by day. [1] Recent security breaches scandals of famous applications are also raising huge concerns. People tends to rely on more secure applications for communication. Applications like *Signal* claim using most secure protocols that gain a lot of attraction for illicit users as well. Device Forensics analysis of these applications is an important domain to locate important locations of evidences. But Network Forensics could also help in detecting live traffic monitoring that can support the evidences found by device forensics examiners. According to Juniper research, Instant messaging users to reach 4.3 billion in 2020. This growth is increased by 9 percent every year [2].

Mobile Forensics is the branch of digital forensics related to the collection of digital evidence often found on mobile apps. With advancement in technology; security and privacy have now captured the attention of the users. Confidence in mobile devices and their integration into daily life has been possible due to the provision of privacy and secrecy services. Commonly used social media apps like WhatsApp, Signal, Tumblr, and many more employ encryption techniques for storing and transmission of data thereby protecting data at rest and data in communication. The provision of security services in these applications has provided an attractive communication medium for digital crime. For the forensic investigator, the provision of end-to-end encryption service in the apps results in increased effort and the high possibility of failure scenarios [3]. This paper studies the popular *Signal* app in an attempt to demonstrate how artifacts of potential value can be extracted through network forensics analysis.

Signal app¹ is a popular cross-platform encrypted messaging service developed by the Signal Foundation (former known as Open Whisper Systems) powered by the open-source Signal Protocol (2013). The application is open source² and available for android, IOS, and desktop. The application provides common features like audio calls, video calls, voice messages, text, media messages, stickers, typing indicators, and many more. Each conversation has its safety number that can be verified among the communicating parties/users. Owing to the provision of security services the application has been endorsed by technologists and cryptographers [4].

Signal application encrypts all the messages using the *Signal* protocol that are based on secure cryptographic algorithms. The protocol consists of Double Ratchet Algorithm, XEdDSA, and VXEdDSA, X3DH, and Sesame. XEdDSA creates a single key pair that is used in the elliptic curve Diffie-Hellman and signatures. VXEdDSA is an extension of XEdDSA to make it as VRF (Verifiable Random Function) [5]. X3DH (Extended Diffie-Hellman) is used to mutually authenticate the users and provides forward secrecy [6]. After sharing a common secret key, the messages are communicated between users by using Double Ratchet encryption algorithm [7]. For session management of messages, Sesame (session management for asynchronous) is used with Double Ratchet Algorithm and X3DH [8].

Two main categories of mobile forensics are device forensics and network forensics. Device forensics helps to determine the file structure and extraction of useful information stored on the device. Network forensics is concerned

¹<https://signal.org/en/download/>

²<https://github.com/signalapp>

with tracing communications through network packets i.e. to find the traces of app's or users' behavior, data breaches, and other illicit activities [9]. Thus, network forensics plays an important role during the investigation as it uncovers network related artifacts which could be beneficial to the forensic investigator. Digital evidence related to networks involving apps is of great significance [10]. In the case of *Signal* app, majority of the communication data is in encrypted owing to which it becomes difficult and often impossible to obtain the corresponding plain text as the keys are unknown.

The *Signal* app is based on famous *Signal* Protocol that is also being used in famous WhatsApp application. Most literature emphasize on the device forensics of the social media applications. However, limited work has been done on network forensics of the Instant Messaging applications.

We have performed an extensive review of the traffic analysis of the *Signal* app and have incorporated the firewall approach to the investigation. The firewall helps to understand the pattern of connectivity and communication activities. We thus forced the *Signal* client to connect to its server in a controlled environment and this arrangement revealed the obscured design of *Signal* app. Additionally, we monitored the live traffic to dissect the bytes, payload-based patterns to identify the *Signal* app different activities, i.e. calls, text, typing indicator, etc. The workflow of network forensic analysis is shown in figure 1.1.

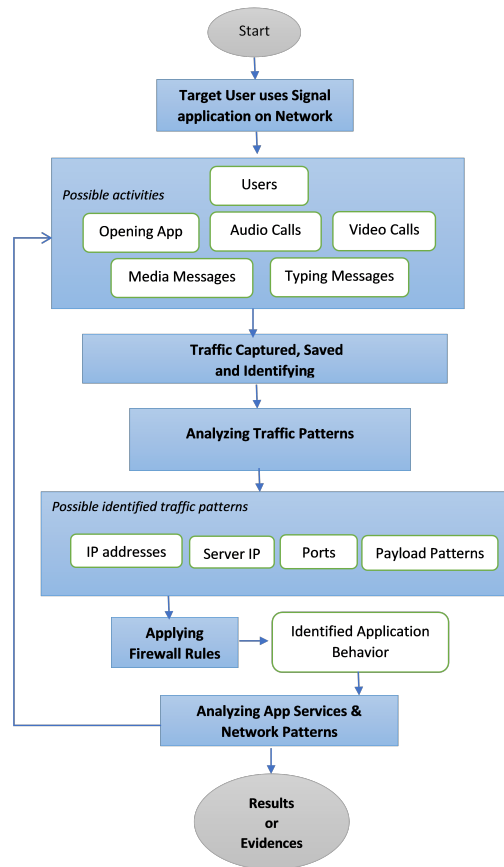


Figure 1.1: Overall Flow of Activities

1.2 Problem Statement

From the aforementioned studies, most of the work focuses on device forensics of social media applications with interesting artifacts. However, limited work has been found on network forensic analysis especially on encrypted network traffic for instant messaging apps. A comparison of existing work is shown in Table 1. There is still room to study the encrypted network traffic to identify various activities of user's and app's behaviors that further utilized for the evidences, either in direct/indirect or supportive manners in

investigation purposes. Therefore, the motivation of this study is to propose a methodology to investigate the encrypted network traffic to uncover the various patterns/artifacts. A research is demonstrated by using Signal app as a case study. Extensively identifying, collecting, analyzing and correlating the remnants acquired from network of *Signal* application usage. Although mobile devices can store valuable information that can help in investigations. But Network forensics can also help in investigation using a standard platform. To the best of our knowledge, limited work has been done from network perspective. *Signal* app uses string cryptography algorithms, so it is more interesting to explore its network side.

1.3 Solution Definition/Description

The proposed solution provides:

- Collecting and saving the ongoing traffic dumps at specific time of interest.
- User specific activities like opening the application, texting and calling etc.
- Collecting IP addresses of the involved parties and servers during communication.
- Repeating the activities to confirm the fixed bytes and packets patterns found against multiple activities.

- Validating results found during the study after blocking the list of servers in the firewall rules.

Through the implementation of the proposed framework, the attack surface is drastically reduced.

1.4 Thesis Motivation

The motivation behind this research is to find the important evidences and remnants left by the encrypted traffic of *Signal* application. As the network traffic is encrypted, so extensive analysis is much needed to determine the patterns generated by the application. Identifying user activities by looking at the encrypted traffic patterns of Signal app is the motivation of the thesis. Network forensics/behaviour analysis of the application helps to find out the traffic patterns specific to the *Signal* application.

1.5 Thesis Contribution

The main contributions are listed as follows:

- The artifacts/patterns generated by the app over the network are analyzed.
- *Signal* app's behavior over the network is monitored after applying different set of firewall rules.
- Extensive experimental analyses are performed to validate the traffic patterns of respective activities.

- If someone is already victim or culprit, by monitoring proposed network traffic, it could help to investigate the case.
- Live network monitoring of *Signal* app.

1.6 Benefits of the Scheme

- The currently proposed investigation strategy is useful for Android devices.
- Proposed strategy can be employed by an organization to disable the services of this app without disrupting the other services.
- Tools behavior and firewall role understanding.
- Finding application signatures.

1.7 Thesis Organization

The organization of the thesis is presented as follows. Chapter 2 throws light on previous work done related to forensics analysis of social networking apps. In chapter 3, the research methodology followed during the research has been discussed. The experimental setup is discussed in Chapter 4. Chapter 5 showcases the result of the experiment. This section also discusses the performed activities/events and their corresponding network communication patterns. Chapter 6 shows the validation and verification of the proposed network forensic strategy based on various use cases, i.e. crime scene recon-

struction and a hypothetical case study. Lastly, chapter 7 sheds light on the conclusion with possible directions for the future.

1.8 Summary

In this chapter, basic concepts are discussed regarding forensics and *Signal* application. It gives an overview of aim and scope of the thesis and presents the main objectives of the research work with overall thesis organization. In the next chapter we will look at the literature review that has been conducted for this thesis.

Chapter 2

Literature Review

Chapter 2 discusses the related work and terminologies. The related work is basically the research carried out by different researchers over the years which is related to the work done in this thesis and contributed towards making a new solution.

2.1 Literature Review

Owing to their extensive use and popularity, social media/ instant messaging apps have been previously been a subject of study from a network and device forensic perspective. A study [9] on *Signal* app showed that a man-in-the-middle (MITM) attack can be launched successfully by compromising the key-exchange service. The attack can be launched by using rooted smartphones with Cydia Substrate and SSLTrustkiller installation. A PC is designated to intercept the traffic using MITM proxy and also provides a WLAN hotspot for the smartphones. The authors have written a script to

aid with the traffic interception. As the *Signal* app is open source its code is modified and installed on the attackers device thus becoming a MITM. In the experiments, the authors demonstrate that 21 out of 28 users failed to verify the identity of the other users by comparing encryption keys. A short coming of the experiment is that it is only applicable to an obsolete version of the application running on the client's phone.

Digital forensics is not limited to acquire data from the computer as cyber criminals are now more active and use multiple devices included smart phones, tablets and flash drives extensively. These devices have volatile and non-volatile memory and investigators adopt different techniques to acquire the evidence from the devices. The technical aspect of an investigation is divided further into different branches including computer forensics, network forensics, database forensics, mobile forensics and cloud forensics as shown in figure 2.1.

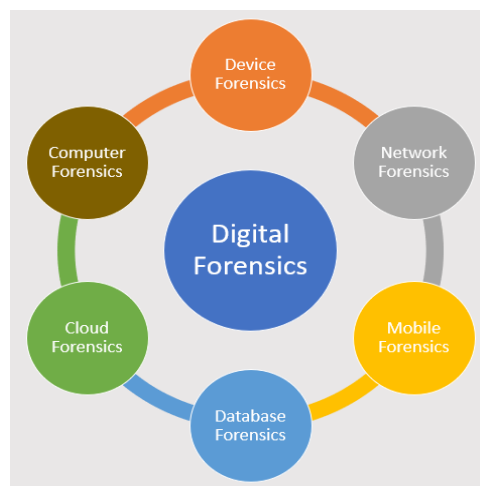


Figure 2.1: Branches of Digital Forensics

2.1.1 Device Forensics

For device forensic aspect, a study carried out by Hao Zhang et al. [10] showed that database artifacts of popular social media apps like Messenger, Hangouts, and Line were unencrypted, unlike WhatsApp which keeps it encrypted. After rooting the android device, the chat messages, timestamps, and contact lists were found in files. However, network forensics was not performed on these apps. Similarly, Fazeel Ali Awan [11] performed forensic analysis of Facebook, Twitter and LinkedIn on four different platforms i.e. IOS, Android, Windows, and Blackberry. The study revealed many important artifacts that can be recovered from device memory except the Blackberry phone. The extracted artifacts include important locations and databases having information related to the visited profiles, names, tweets, contacts, profile ID, etc [11].

Similarly Al-Mutawa et al. studied applications like Facebook, Twitter, and Myspace on android, IOS, and Blackberry devices. Through device forensic analysis of these apps it has been shown that valuable data resides in the device memory which can be extracted after logical acquisition except from Blackberry-based devices [12]. Recently, Shawn Knox et al. targeted the *Happn* social dating app for forensic analysis and successfully identified various artifacts for investigation purposes [13]. The extraction of artifacts from apps of personal nature poses security risks and could also result in privacy violations of various nature.

In a similar study, Cosimo Anglano et al. targeted the Telegram Messenger app for analysis. The decoding and correlating information helped

to reconstruct the contact list, contents of messages and logs of voice calls. Most experiments performed in the research were done on virtualized android smartphones. Subsets of these experiments were also performed on a physical device to validate the results. These results could help in solving the cases during forensics investigations [14]. However, this study did not target the network forensic analysis especially end-to-end encryption based traffic.

Christoforos Ntantogian et al. [27] evaluated the security of the android applications. They successfully recovered the authentication credentials from the physical memory by using forensics tools. After rooting the device important data and artifacts related to famous applications were found in the memory dump.

Mauro Conti et al. used machine learning techniques to sniff user actions. The network traffic was fully encrypted during the investigation which could not be decrypted. Steps used in this study include domain filtering, packet filtering, timeout and packets interval. The results showed that Facebook, Gmail, and Twitter apps could be used to infer user activities. This method could help in learning the behavior of one or more users thereby facilitating intelligent decisions making by both an adversary and security expert. The authors discuss that some countermeasures could also be taken to frustrate the attackers like adding some padding techniques etc. [18].

Humaira et al. [19] proposed a semi-automated model to conduct forensic investigation. The model described uses both previous standard methods and automated features against OSN (Online Social Networks) for digital investigation. As some processes can be automated in forensics but still

there is need of human decision-making skills to drive the investigation in the right way. As fully automated digital forensics systems are not accepted in judicial systems. However, the data set against which testing is conducted is not large enough to conduct rigorous testing.

K. Rathi et al. [31] have forensically analyzed the instant messaging applications namely WeChat, Viber, WhatsApp, and Telegram. Authors results show that it is possible to retrieve the encrypted database files of WeChat and Viber from a rooted phone. It is also possible to retrieve WhatsApp messages from un-rooted phone. Authors concluded that Telegram application is more secure than the rest however, Telegram messages can be recovered if the phone is rooted.

S. Wu et al. [32] have analyzed WeChat application installed on different phones. Authors have explored different scenarios like acquiring the user data and decrypting the encoded database, investigating the communication and data shared by the user. When the application is updated, changes in the structure occurs, therefore investigators must remain updated continuously.

C. Anglano et al. [34] have analyzed the artifacts of WhatsApp messenger application. They have managed to decode and interpret the artifacts. They have also presented a way to correlate these artifacts. Authors have claimed that by using their technique one can get information like contacts or chronology of messages. Only limitation of the paper is that it applies to Android only and not iOS.

V. Venkateswara Rao et al. [28] presented the forensics analysis of the important locations in memory of the device. They found locations related to chats, call history, contacts and images in the memory areas of the smart-

phone. Application specific databases were also found that can help to reconstruct any event.

Ahmad Ababneh et al. [29] performed forensics analysis of the famous communicating application IMO both on android and windows. They were able to recover remnants from NAND memory from the android device. Artifacts like contact lists, chat databases and contact lists were recovered from the windows RAM even after deleting the data and uninstalling the application.

2.1.2 Network Forensics

Network forensics is all about to trace the users involved in investigation cases. Non-Repudiation is the main challenge which requires deep knowledge of the network devices and forensics techniques. ISP's and routers could be traced but tracing the mobile devices is a challenge. Rachana Y. Patil et al. [21] proposed a protocol to generate a fingerprint that can be used as evidence in court.

Network forensics artifacts are considered critical evidences that can be collected automatically. This can help in the scenarios where large set of data sets like social networks. As data from the social media is of lot of importance and is increasing day by day. Criminals can use this data to achieve their goals like malware distribution, fraud harassment and so on. Law enforcement agencies need to analyze the social media platform to find any lead to track the offenders.

Anti-Forensics techniques are extremely challenging for the network foren-

sics examiners. Besides getting the IP address involved in the incidents, it is very important to find out the actual user whose identity is compromised. N. Clarke et al. [20] presented a novel technique to identify the users in the network by analyzing the metadata of the traffic. They collected 112 GB of 46 user's data spanned over two months and examined the novel user-interaction based feature extraction algorithm. The approach can distinguish and recognize the users with the accuracy of more than 90 percent. This reduces the traffic volume analyzed by the investigator and more focused investigation could be further done.

Daniel Walnycky et al. analyzed the device and network traffic of 20 applications. They were able to reconstruct the messages and collected evidence traces like password, screenshots, pictures, etc [15]. Network forensics analysis targeted by Mohd Najwadi Yusoff et al. on social networking apps i.e. Facebook, Twitter, and Telegram on Firefox Mobile OS simulator. In this study, simulator-based generated traffic was sniffed by the Wireshark network monitoring tool. The authors showed the IP addresses, ports, domains, and sub-domains. However, they were unable to decode the different patterns, events and activities [16].

In a recent research, forensic analysis of media app IMO was published in which the encrypted network traffic was interpreted. In this work, M.A.K. Sudozai et al. studied the IMO file structure for Android and IOS platforms. A firewall was incorporated to assess the encrypted network traffic of the IMO app. The authors show it is possible to identify the pattern of activities in encrypted traffic [17]. However the research is limited to only the IMO app.

G. B. Satrya et al. [25] discussed the remnant data form the three social media applications Telegram, Line, and KakaoTalk. They collected the meta-data of chats and analyzed the results of the mentioned applications. These artifacts were related to each other and helped the forensics analysts to start investigating the incident. This research involved two different smartphones, both online and offline android forensics, against normal and secrets chats using the applications. Offline forensics results showed that Telegram chats used no encryption for normal and secret chats. Some tables were encrypted in Line application. While KakaoTalk encrypted all the messages except the last one.

F. Karpisek et al. [33] have performed network analysis of the WhatsApp messenger application. They have managed to decrypt the network traffic and have obtained the forensic artifacts related to calling feature. They have explained the tools and methods to decrypt the traffic. Limitation of the paper is that their methodology is Android specific, they have not given any methodology related to iOS.

Mobile applications are used for illicit purposes by the criminals. These applications vary in nature and numbers. Xiaodong Lin et al. [22] proposed an automated solution named Fordroid to analyze different type of applications. Fordroid is an automated tool analyzes the android based devices to identify the important remnants in local storage. During the study against 100 random mobile applications, they found out 469 paths to store data and structure of the databases in many applications.

2.1.3 Automated Tools and Frameworks

Cosimo Anglano et al. [23] created a software tool that automates the forensics analysis of the android applications. Manually analyzing the artifacts of the applications and reconstructing the activities is tedious work. This tool is based on a black box approach, in which application is installed on a virtualized android device. Activities are performed with human interaction emulation with the applications and file systems are analyzed against the activities. Results showed that this tool simplifies the forensics analysis of the applications.

Humaira Arshad et al. [24] proposed a multi-layer framework that helps the investigators from the process of collection till the analysis of the evidences. Main component of this framework is hybrid ontology approach which helps to manage the unstructured data of social media and integrate them. Basic idea to use this approach is to collect the important evidences by automated trustworthy methods that are acceptable in court of law as well.

Joshua I. James et al. [26] discussed that although automation speed up the process of investigation, but it has many challenges. Famous tools like EnCase, Forensic Tool Kit and Autopsy Forensic Browser make things easier for the examiners and opposes to the perception that the investigators should have a manual knowledge only. However less knowledge and experience might not help to critically analyzing the tool-based results. Correctly implemented automation in right time of investigation phase will produce quality results.

Fu-Ching TSAI et al. [30] proposed a packet-based filter framework to de-

tect the criminal identity like IP address from huge network data collection. This research targets the famous whatsapp application. The framework simplified the complex networks to detect the connection between suspect and the victim.

Famous application Viber was forensically analyzed and derived a model to correctly detect its traffic. The model was able to classify the services chats, calls along with the IP addresses of the users by incorporating the port-based, byte patterns and payload sizes [35]. This research was accurate enough to be verified on both android and IOS platforms.

A novel framework that helps in profiling the secure apps to extract the hidden patterns of these apps [36]. This method helps to solve investigations as well as business solutions for the service providers. Form distinguishing the specific traffic of the target app to the activities of involved users can help an ongoing investigation.

To acquire Digital evidences from the physical memory of the mobile and desktop devices, a framework RAMAS is designed by the Diogo Barradas et al. [37] This framework is open source that can extract client email data and instant messaging data from the volatile memory. As compared to static analysis, physical memory analysis is much harder as it needs inspecting low-level memory structures on a specific OS. RAMAS tackle many different regular expressions issues related to string matching the application's syntax for extracting the evidences.

According to Australian law enforcement agencies, a significant increase has been noticed to analyze mobile devices forensically every year. Therefore, the volume of data is also increasing for intelligence analysis. Darren Quick

et al. [38] proposed a utility Digital Forensic Intelligence Analysis Cycle to locate information from mobile devices. Common extract solutions help to place data into XML or spreadsheet format, which can help in processing the data for other devices as well.

It has been observed that most of the aforementioned schemes targeted the device forensic for social media app while highlighting the interesting results. However, limited work has been found which focuses on forensics analysis of encrypted network traffic. There is still room to study the encrypted network traffic to identify various activities of users and apps behaviors. This forms the motivation of this study, i.e. to target *Signal* app from network forensics point of view to uncover various pattern in encrypted traffic. A brief comparison of the previous studies is also shown in table 2.1.

Table 2.1: Comparison of recent approaches

Research Work	Year	Apps	Mobile device forensics			Network Forensics		
			Acquiring the data	Decoding the encrypted messages in databases	Artifacts Locations	Detecting User Activities from the Packet	Important Artifacts	IP addresses of Involved parties
Forensic Analysis of Encrypted Instant Messaging Applications on Android - Khushboo Rathi, Umit Karabiyik	2018	WeChat, WhatsApp, Telegram, Viber	Yes	Yes	Yes	No	No	No
Forensics study of IMO call and chat app - Sudozai et al.	2018	IMO	Yes	Yes	Yes	Yes	Yes	Yes
Forensic analysis of WeChat on Android smartphones - Songyang Wu et al.	2016	WeChat	Yes	Yes	Yes	No	No	No
WhatsApp Network Forensics: Decrypting and Understanding the WhatsApp Call Signaling Messages - Filip Karpisek et al.	2015	WhatsApp	No	No	No	Yes	Yes	Yes
Network and device forensic analysis of Android social-messaging applications - Daniel Walhycky et al.	2015	Snapchat, Wickr, BBM, Tinder and 16 other apps	Yes	No	Yes	Yes	Yes	No
Forensic Analysis of WhatsApp Messenger on Android Smartphones - Cosimo Anglano	2014	WhatsApp	Yes	Yes	Yes	No	No	No

2.2 Summary

This chapter covers the background and the related work of the thesis. The related literature has been presented along with a critical analysis of the studies. Previous research work and schemes used in the literature helps in

formulating the solution to the identified problem. In the following chapter, we will discuss the research methodology that has been followed during the thesis.

Chapter 3

Research Methodology

Chapter 3 explains the research methodology that is followed to carry out this thesis research. A brief description of the methods that are used in our research methodology along with the phases followed in the research process, *i.e.* identifying the problem, formulating hypothesis, making important observations and evaluating the system is given in this chapter.

3.1 Introduction

The research is the procedure of probing or collecting information specific to the area under consideration. It can be defined as a scientific investigation to discover new information and facts [39]. Whenever there is a need to answer a problem and finding appropriate solution to that problem, research is performed [40]. As stated by Clifford Woody, research is the process of defining and redefining the known problem, formulating hypothesis and suggesting solutions to the known problems, assessing the collected data, making as-

sumptions, deriving conclusion and lastly testing the conclusion for verifying the stated hypothesis [41].

3.2 Research Types

Research methodologies which were proposed in the literature are given below and described in figure 3.1.

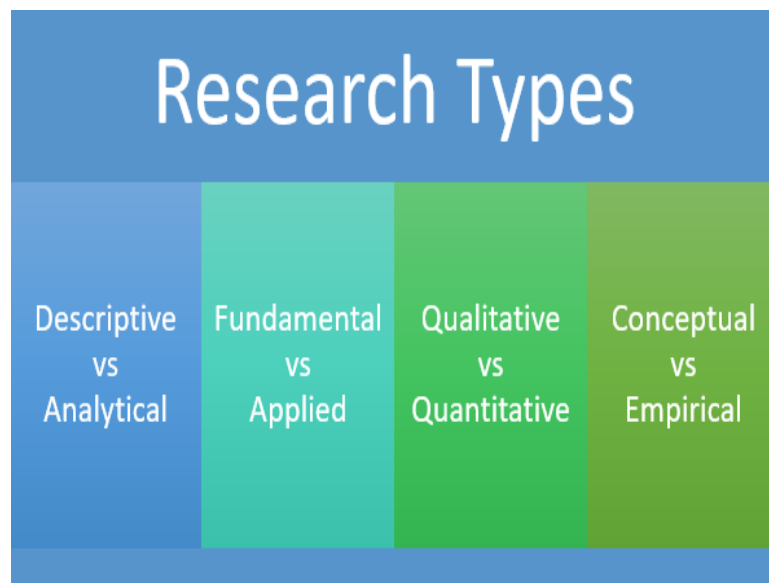


Figure 3.1: Research Types

3.2.1 Descriptive vs Analytical Research:

In descriptive research, the present work is discovered, and small reviews are conducted to find the essential and linked information and proofs. Its objective is to define the work about the specific area of research which can additionally be used in upcoming research work. The main features of descriptive research are that the researcher has no control over the data and

present literature; rather they can only find the proofs and methods. In analytical research, the researcher uses all the gathered information from the reviews and do the critical analysis and assessment to reach the conclusion [39, 40].

3.2.2 Fundamental vs Applied Research:

The research can either be fundamental or applied depending on the depth of the knowledge required. In fundamental research, the main purpose is to collect basic knowledge and discover the basics of the scientific phenomenon. In applied research, the focus of the researcher is to find some suitable solution to the problem faced by other researchers, organizations or society. Different experiments are performed by the researcher to investigate and analyze the problem and achieve in-depth knowledge of the problem area. The approach to find solutions of different problems on the based on performed experiments is very helpful [39, 40].

3.2.3 Quantitative vs Qualitative Research:

Quantitative research is based on the quantitative measurement of some features. It can be done in domains where things can be stated with respect to quantity. Whereas in qualitative research, things assessed based on the quality, e.g. the human conduct or opinions on specific things and the purpose behind those opinions [39].

3.2.4 Conceptual vs Empirical Research:

Conceptual research is based on abstract thoughts or the concepts normally used by the theorists. This is done to create new concepts or re-design the present ideas. The empirical research is based on the observations and experiments without trusting on any concept or the scheme. It is truly based on their own set of observations, experiments, and conclusions. In this type of research, firstly, hypothesis is made based on the facts and then try to assume the outcomes. Researcher then collects the proofs to accept or reject the hypothesis [39].

3.3 Research Methodology Overview

To conduct the research, different techniques, methods and procedures are used by the researchers which is called research methodology. The process is started with conducting reviews until reaching the results and this as a whole is termed as research methods. The research methodology is defined as the principles and procedures to carry out research using a scientific approach. It includes all the stages used by the researcher to find a solution to the problem. The research methodology is different for different types of research problems, and the researcher must know the correct way to be used to follow the research. The research objective should be very clear to the researcher towards the selection of a research methodology [39, 40]. In this thesis following research methodology steps are used:

- Explored digital forensics specifically device forensics, network foren-

sics and mobile forensics to gather significant knowledge for the target domain.

- Digging the specific area of network forensics to find out the behaviour of the instant messaging applications. Detecting the patterns from live network traffic of secure instant messaging application *Signal* specifically.
- Find the specific patterns/ behaviour against the significant activities performed by using the application.
- Derive hypothesis from the analysis of the existing literature.
- Validate the formulated hypothesis after implementation of the proposed system.

3.4 Thesis Research Methodology

This thesis used the hybrid approach throughout the research, which includes conceptual, applied and fundamental research methods. This thesis consists of some steps which are followed throughout the research phases starting from collecting information till obtaining the results which are then assessed. All the steps in our research process are given below.

- Define a Research Area
- Literature Survey
- Research Method

- Data collection
- Analysis and Interpretation
- Presentation and Results/Findings

3.4.1 Define a Research Area

Finding the research area is very critical part of the research phase. There are many domains in the field of Digital Forensics like Device Forensics, Network Forensics, Mobile Forensics and so on. The researcher needs to have a thorough knowledge of all of the above mentioned domains to further select a topic. After choosing a domain of interest, an extensive literature review helps to identify the problem statement. The problem statement is based on five main questions.

- **Who** is conducting this research, who is the beneficiaries of this study?

This research is conducted by Asmara Afzal that will be beneficial to the digital forensics investigators, forensics community and forensics practitioners.

- **What** is the significance of this study? And what steps involved in this study?

This study is about digitally analyzing the secure instant messaging application *Signal* forensically. It will help digital forensic investigators and research community to quickly analyze the application for evidences. This study is qualitative and qualitative search give us more freedom in data collection perspective. The purpose of this study is

to collect encrypted traffic artifacts of *Signal* app to perform network forensics and behavioral analysis. These artifacts are helpful to get the expected results. The analysis will be performed on different activities of the application. The adopted approach is based on case study which incorporates our research cycle. In case study methodology we have observed real world situation for application usage. The case study approach further divided into four steps:

- Case Selection and Situation identification
- Collection and recording
- Data interpretation
- Report writing

- **Where** did this study conducted?

This study is conducted at National University of Sciences and Technology H-12 Campus in KTH Lab.

- **When** this study is conducted?

This study is conducted from December 2019 till April 2020.

- **Why** this research is required?

Network Forensic examination of the application will help forensic investigators, practitioners and research community in forensic investigation of the Signal application. This will help in identifying, collecting and analyzing the important patterns of encrypted traffic over the network.

3.4.2 Literature Survey

Literature review is the most important phase during a research. Literature review helps researchers to better understand the area of research and also get to know previous research work done by other researchers along with the gaps between the existing study and previous study. To conduct an effective literature review, following steps are essential.

- Go through relevant research papers.
- Select papers from near past, not older than 4,5 years.
- Select 20-30 papers from chosen domain.
- Sort the papers with respect to research time and domain knowledge.
- Initially go through abstract and conclusion to find relevancy.
- Select papers which are relevant to your research.
- Write short summaries and useful points.
- Perform extensive analysis over those summaries.
- Shortlist recent and relevant researches.
- Thoroughly review selected papers.

3.4.3 Research Method

After the literature review, when the gap is identified in previous researches, the next step is to formulate research method. Research method helps in

answering the question in a systematic way. Research method identify all the aspects of the research from collection of data to time-frame required to implement the solution.

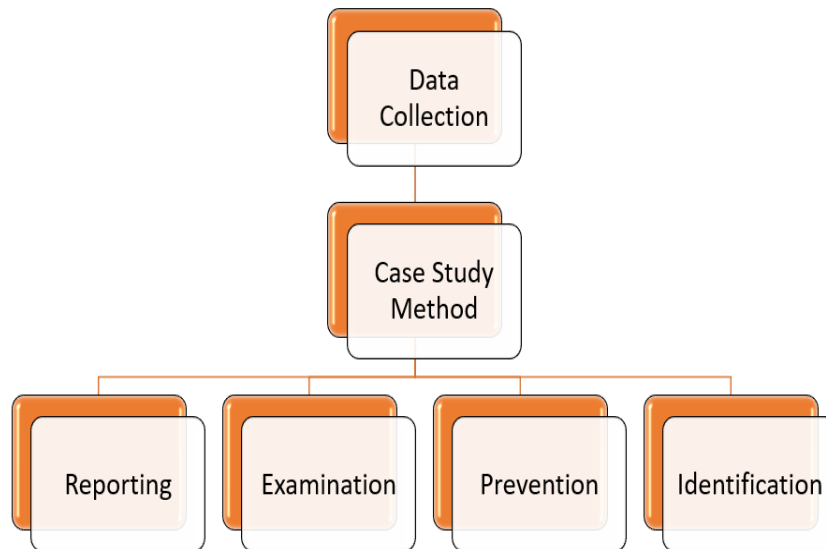


Figure 3.2: Research cycle with NIST Guidelines

3.4.4 Data collection

Data collection is the fourth phase of the research cycle. It is the most important phase in the research process as all the results outcomes are influenced by the data collected by the researcher. The data collection phase has two sub process.

1. Selection of cases and identification of situations
2. Collection and recording of data

To perform these processes, we followed the NIST guidelines which are based

on “Martini Framework”. The following figure shows the steps and details of the framework.

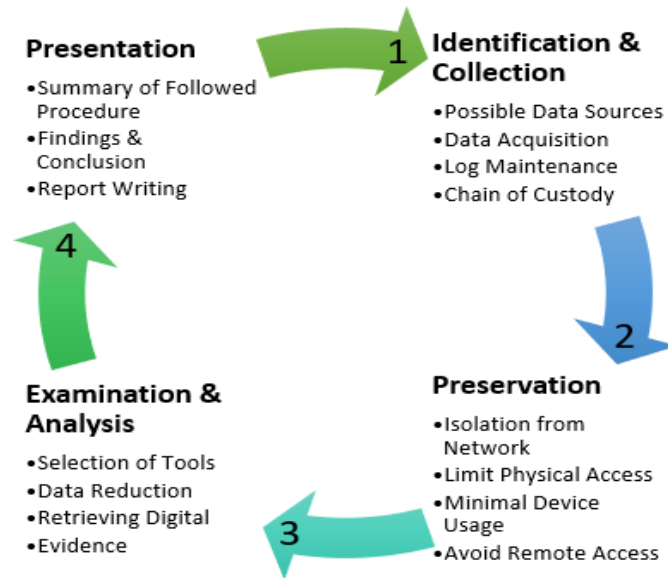


Figure 3.3: Framework Based on NIST Guidelines

3.4.5 Analysis and Interpretation

This is the fifth phase of the research cycle. In this phase we thoroughly analyzed the gathered artifacts of the activities performed on the selected application. We performed the following activities and analysed them with reference to the guidelines provided by NIST.

- Identifying location for performed activities
- Gathering and analyzing artifact of activities
- Cross verifying the results of performed activities.

3.4.6 Presentation and Results/Findings

Presentation of this research work covers all the basic concepts that are required for the target audience to understand this work. The work is presented in a sequence covering all aspect from beginning to the end so that the target audience can understand the problem and get along nicely. To conclude the presentation, it is ought to be arranged in “.docx, .pdf or .pp”. Tables and figures will be part of the final document to comprehensively expalain the work.

3.5 Summary

In this chapter we have covered different research methodologies that have been proposed for research which can be followed by the researchers. As every research method is appropriate for different research scenarios of this thesis, hence all these research methods are followed at different times. In next chapter we will look at the experimental setup designed to perform the analysis.

Chapter 4

Experimental Setup

This chapter explains the experimental setup that has been designed to carry out this network forensic analysis. This chapter also provide justification that why firewall has been used. System configuration is also provided in this chapter.

4.1 Overview

To capture network traffic, network infrastructure has been designed as depicted in figure 4.1 as inspired by [17]. The proposed experimental setup includes a pfSense firewall, an access point, and a PC to display the ongoing traffic of mobile devices passing through the network.

4.1.1 Pfsense Firewall Configuration

The *Signal* app based mobile device is connected to the internet through a wireless access point, and all the internet traffic of the wireless access point

is routed through a PC based pfSense firewall. The role of the firewall is to monitor, sniff, and capture the entire network traffic. Thus the firewall generated trace files are saved for analysis. The configured firewall filters out the *Signal* app based internet traffic in a controlled environment.

4.1.2 Access Point, Wireshark and Mobile device

The access point aids to connect the other devices such as mobile and PC with firewalls. For network traffic analysis, Wireshark software is installed to monitor the encrypted traffic through analysis of the trace files. It is noteworthy that only the payload is encrypted while the IP addresses and ports are in plaintext. This creates an opportunity to determine behaviours based on communication credentials and not on the content that is being communicated.

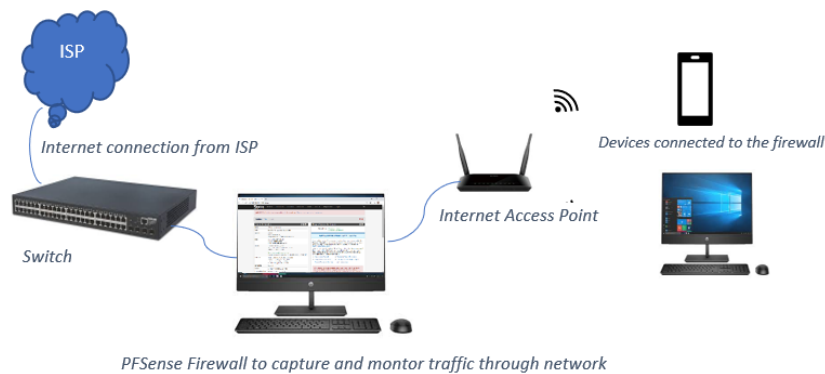


Figure 4.1: Experimental Setup

4.2 Applying firewall rules

To analyze the *Signal* protocol behavior, e.g. how client and server communication works? The proposed strategy sniffed the encrypted traffic between the client and its servers, where it shows a list of server IP addresses and ports. Identification of ports and IP addresses aids in imposing new rules in the network. For example, different servers (based on IP addresses) are dedicated to providing separate services like text and chat are different compared to call based servers. A rule relating to blocking of chat servers which are extensively noticed during the study results in disruption of chat service. An alternate connectivity mechanism would help to continue these services. The details of user activities like opening the *Signal* app, text indicators, text messages, and audio calls are discussed in further detail in the result and analysis section.

4.3 Justification for the need of a Firewall?

Placing a firewall in the investigation network gives control to detect the app behavior in an effective manner. Firewall rules are used to confirm the default app behavior. Furthermore, restrictions can be applied and ulterior app behaviors can be brought to light. Default and alternate traffic patterns could assist the forensics examiners in solving the cases involving the *Signal* or any other app. Moreover, it also helps in observing the client-server connectivity design pattern such as TCP/UDP ports and server ranges, etc. In this scenario, firewall configuration requires defining rules according to net-

work requirements for both WAN and LAN. By default both inbound and outbound traffic would be blocked through the firewall. The LAN rules are defined to configure access to internet resources. As a standard practice there are no rules defined for WAN NIC which means all inbound traffic would be blocked unless configured otherwise. If there is any web server in a network, a WAN rule can be defined to access that server. The ‘Packet Capture’ option in the firewall is used to capture the live traffic of the target device as shown in figure 4.2.

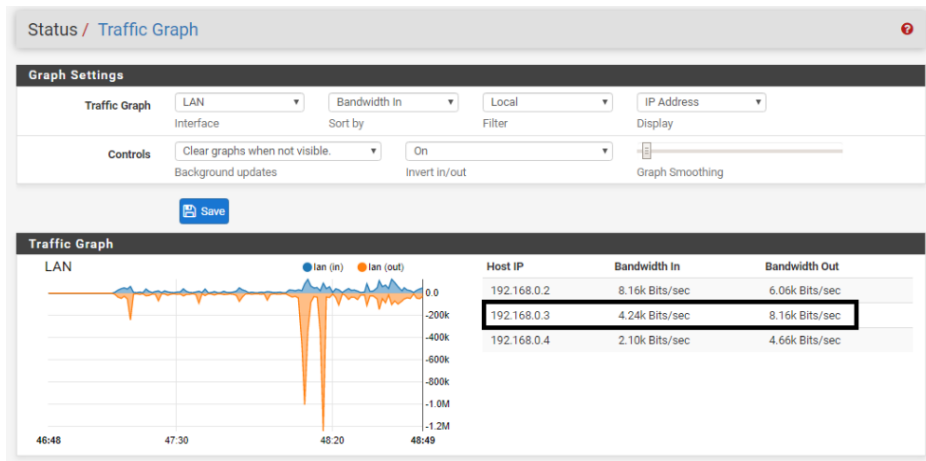


Figure 4.2: IP address of smartphones Traffic passing through pfSense

4.3.1 Setup for the smartphone

Explicit capturing of smartphone’s traffic using *Signal* app through a firewall or any other device in the network is possible with the help of IP addresses as shown in figure 4.3. This helps to reduce the reliance on ‘Port Mirroring’ deployed in the setup explained in paper [17]. The captured live network traffic is saved from LAN/WAN as trace files which are analyzed by Wireshark.

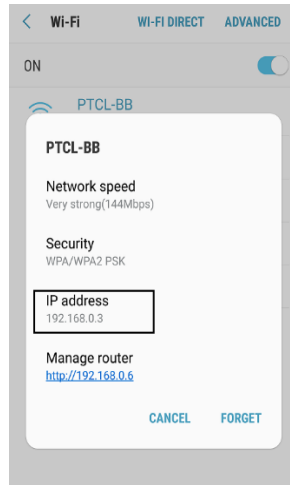


Figure 4.3: IP address of the Smartphone

Noticing IP Addresses: The IP address for WAN is given as 10.2.31.180/24 whereas the IP address of LAN interface is 192.168.0.6/24, thus static having a pool of DHCP LAN IPs range from 192.168.0.1 – 192.168.0.40. The system specifications of PC on which firewall is deployed is a 64-bit Operating system that has an x64-based processor, 4GB RAM, Intel (R) Core (TM) i5-3470 CPU @3.20 GHz. pfSense version 2.4.4-p3 (amd64version) is installed.

4.4 Summary

In this chapter we have covered the experimental setup that have been proposed to carry out the analysis during the research. Moreover justification for using the firewall and its configuration has been provided in this chapter. In the next chapter we will look at the results and analysis performed during the research.

Chapter 5

Results and Analysis

This chapter explains the achieved results and their analysis in the form of bytes size of packets. Resulting IP addresses found in the study are restricted to verify all the activities that have been monitored during the analysis.

5.1 Overview

Signal app uses strong encryption algorithms to encrypt data for communication. In this study, extensive analysis of traffic dumps was done and revealed bytes and payload patterns of different activities. But these patterns did not help in reconstructing any encrypted data. However, through the experimental setup explained in the earlier section, regulatory authorities and law enforcement agencies can determine the behaviors of *Signal* app by observing consistent connectivity patterns.

5.2 Ports and Servers Ranges

Monitoring encrypted traffic of *Signal* app passing through the firewall revealed that a common 443 TCP port is being used throughout the communication. Blocking of 443 port on the firewall would also block other services on the smartphone that utilize the port. Therefore, blocking of 443 port is not a suitable method to check other connectivity patterns of the *Signal* app. After each finding, the firewall rules are updated on a to evaluate new network patterns against different activities/events performed by the target user. Hence, network patterns against activities like calling, texting, and typing are monitored and analyzed repeatedly to firmly conclude.

Table 5.1: Identified Chat Servers

Sr#	Server List 1	Server List 2	Server List 3	Server List 4	Server List 5
1	52.207.41.59	34.225.196.214	34.196.69.69	100.24.0.111	35.169.3.40
2	100.24.0.111	3.228.254.81	3.222.249.138	52.207.41.59	34.196.194.172
3	54.175.47.110	34.196.69.69	100.24.0.111	54.175.47.110	34.225.240.173
4	34.196.69.69	54.175.130.206	3.228.254.81	54.175.149.136	34.225.240.173
5	3.228.254.81	54.175.149.136	54.175.149.136	3.222.249.138	107.23.71.89
6	3.222.249.138	3.222.249.138	52.207.41.59	54.175.130.206	35.169.3.40
7	54.175.130.206	100.24.0.111	34.225.196.214	34.225.196.214	34.196.194.172
8	34.225.196.214	54.175.47.110	54.175.47.110	34.196.69.69	107.23.71.89

Unlike many other XMPP (eXtensible Messaging Presence Protocol) based applications that use specific TCP ports like 5222, 5223, and 5228, *Signal* app uses port 443 for communication, while random UDP open ports are used for voice and video calls. Here the IP addresses of the servers are extensively noticed against many activities. It is observed that IP addresses of Class A are used for text messaging as shown in table 5.1. For call services, *Signal* client app uses Google's servers in combination with the observed IP addresses of servers as shown in table 6.2 in next chapter. This may have

been done for better connectivity or load balancing of traffic.

Similarly, a list of servers is identified against different activities. In case of blocking the servers, IP would result in denial of service discussed in section.

5.3 Validation and verification

It is also important to note that repetitive patterns of packets are identified i.e. certain payloads size against specific events that help to determine the events types, etc. Although privacy and secrecy are not compromised, still size inspection of packets reveal prominent events as it is summarized in table 5.2 at the end of this chapter. For better understanding, the following users are created with a description.

User A: Target user against whose entire set of activities are monitored.

User B: A user with whom User A communicates.

5.4 Identifying Signal's Traffic

To identify the events associated with Signal app, traffic dumps from the target android device are saved for further analysis. The captured traffic is encrypted and cannot be decrypted without the cryptographic key. Despite this multiple activities were performed to uncover traffic patterns against the activities. Hence, the traffic patterns are noticed against the Target User A as sender, caller, and receiver. The following activities are performed.

- (a) Accessing/Opening the *Signal* App

- (b) Target Device (User A Typing) Patterns
- (c) User B Typing Patterns
- (d) Call initiated by User A (Caller)
- (e) Call received by User A (Callee)
- (f) Media Messages
- (g) Video calls

User A is a target device on which multiple activities are performed. These traffic patterns could help us to determine which activities are performed by randomly looking at the trace files. For forensic experts, network administrators and security engineer the emerging patterns would help to examine many cases depicting various scenarios.

5.4.1 Accessing/Opening the Signal App

To identify the *Signal* traffic from other existing traffic over the network requires to find the patterns. Where an extensive number of trace files are captured and monitored during the opening of the *Signal* app. We observed that a standard DNS query is sent to access the *Signal* server from client end i.e. "textsecure-service.whispersystems.org". In response, a list of eight servers with IP addresses is sent back to the client to establish a connection. In most cases, it was noticed that the first two servers mentioned in the list are used for establishing communications. *Signal* app uses two random ports to establish a connection with the server port 443 as shown in figure 5.1.

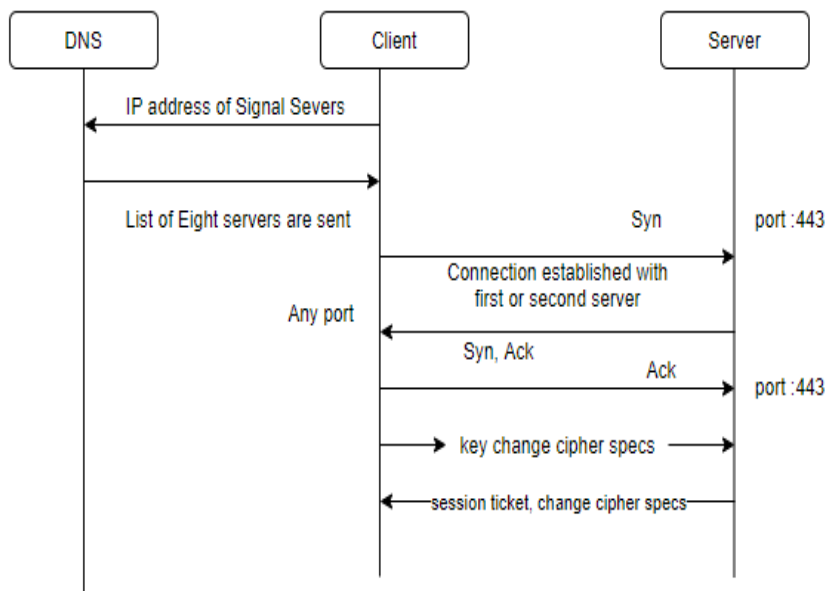


Figure 5.1: Authentication Phase

To identify the opening/accessing of *Signal* app activity, we opened the *Signal* client app without doing any activity, further the app was turned closed. As we observed that before starting any conversation or any activity, a TCP connection is established in three steps *syn*, *syn ack*, *ack* between client and server. After that, a TLS handshake occurs by using TLSv1.2, certificates, and session keys are exchanged between client and server. Next client exchanges key change cipher specs and send encrypted handshake message to the server. In response, the server sends a new session ticket, change cipher specifications, and encrypted handshake message to the client device using TLS.

After an encrypted handshake, during traffic transmission, it is further observed that the *Signal* client sends packets of (449,372,127) bytes having a

payload of size (383,306,61), respectively to the *Signal* servers. In response, server sends packets of (261,261,137) bytes to the *Signal* client having payload of (195,195,71), as showed in figure 5.2, where the *Signal* client IP is 192.168.0.5 and Signal server IP is 54.175.47.110. In most cases, this behavior is found after the negotiation of session keys between the client and the server. This activity is performed multiple times to confirm the patterns of opening the *Signal* app. It is concluded that the above patterns of bytes transmission indicate the opening of the signal application.

No.	Time	Source	Destination	Protocol	Length	Info
33	09:44:46	192.168.0.5	54.175.47.110	TCP	66	47880 → 443 [ACK]
34	09:44:46	54.175.47.110	192.168.0.5	TLSv1.2	1134	Certificate, Serv
35	09:44:47	192.168.0.5	54.175.47.110	TCP	66	47880 → 443 [ACK]
36	09:44:47	192.168.0.5	54.175.47.110	TCP	66	47879 → 443 [ACK]
37	09:44:47	192.168.0.5	54.175.47.110	TCP	66	47879 → 443 [ACK]
38	09:44:47	192.168.0.5	54.175.47.110	TLSv1.2	192	Client Key Exchan
39	09:44:47	192.168.0.5	54.175.47.110	TLSv1.2	192	Client Key Exchan
40	09:44:47	54.175.47.110	192.168.0.5	TLSv1.2	356	New Session Ticke
41	09:44:47	54.175.47.110	192.168.0.5	TLSv1.2	356	New Session Ticke
42	09:44:47	192.168.0.5	54.175.47.110	TLSv1.2	449	Application Data
43	09:44:47	192.168.0.5	54.175.47.110	TLSv1.2	372	Application Data
44	09:44:47	54.175.47.110	192.168.0.5	TLSv1.2	261	Application Data
45	09:44:47	54.175.47.110	192.168.0.5	TLSv1.2	261	Application Data
46	09:44:47	54.175.47.110	192.168.0.5	TLSv1.2	137	Application Data
47	09:44:47	192.168.0.5	54.175.47.110	TCP	66	47880 → 443 [ACK]
48	09:44:47	192.168.0.5	54.175.47.110	TLSv1.2	127	Application Data
49	09:44:47	192.168.0.5	54.175.47.110	TCP	66	47879 → 443 [ACK]

Figure 5.2: Signal Application Opened

5.4.2 Target Device (User A Typing) Pattern

To notify the patterns of the *Signal* app when target User A starts typing a message in a chat window. Certain flow patterns with fixed payload size are

noticed. To be assured, these patterns are observed several times in trace files to deduce results. In most pcap files, the uncovered pattern is shown in figure 5.3. Target User A is highlighted with IP 192.168.137.69.

No.	Time	Source	Destination	Protocol	Length	Info
1	09:58:25	192.168.137.69	3.228.254.81	TLSv1.2	1181	Application Data
2	09:58:25	3.228.254.81	192.168.137.69	TLSv1.2	139	Application Data
3	09:58:26	192.168.137.69	3.228.254.81	TCP	66	44378 → 443 [ACK]
4	09:58:33	192.168.137.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1
5	09:58:35	192.168.137.69	3.228.254.81	TLSv1.2	1182	Application Data
6	09:58:36	3.228.254.81	192.168.137.69	TLSv1.2	140	Application Data
7	09:58:36	192.168.137.69	3.228.254.81	TCP	66	44378 → 443 [ACK]
8	09:58:36	192.168.137.1	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1
9	09:58:39	192.168.137.69	3.228.254.81	TLSv1.2	1181	Application Data
10	09:58:39	3.228.254.81	192.168.137.69	TLSv1.2	139	Application Data
11	09:58:39	192.168.137.69	3.228.254.81	TCP	66	44378 → 443 [ACK]

Figure 5.3: User A typing a message

It is noticed that when user A with IP 192.168.137.69 starts typing in a chat window 1181,1182 and 1183 bytes of data packets with 1115 and 1116 payload size are sent to the *Signal* server. In response, the server sends 139 or 140 bytes of data packets with 73,74 payload sizes back to the *Signal* client. Signal client in response sends an empty packet of 66 bytes with 0 payload size to the server as an acknowledgment of the previous packets. *Packet Capture* is turned on to capture network traffic as soon as the user opens the *Signal* app to type a message. When the user stops typing and closes the app without sending it to the receiver, the captured trace file was saved for

detailed analysis. These packet patterns are noticed with the *Signal* server during typing of any text in the chat thread as the message is not yet sent to the receiver.

5.4.3 User B Typing Patterns

Analysis has shown that typing traffic patterns of Target user A in the previous section. Similarly, in this section, we will analyze the traffic patterns of User B once it starts typing in his/her chat window. When user B starts typing a message in his/her chat window, which will be sent to the target User A, following noticeable patterns are observed. It is always noticed that 729 bytes of packets having 663 bytes of payload are sent from the server to the client at the device of target User A. Client at target User A also sends 122 bytes of the packet in response having 56 bytes of payload. The target client device A also sends a response with packets 105 bytes. In sequence the server keeps sending alerts to the client of 101, 97 bytes data packet with 35 and 31 payload size as shown in the figure 5.4. This pattern was observed and confirmed repeatedly through multiple iterations.

5.4.4 Call initiated by User A (Caller)

Once the user A dials a call, the IP address of the receiver is also captured. While this scenario differs in case of sending text messages, the IP address of the receiver would never be known even if the users are on the same network. Hence, binding between both users is explicitly seen. It is observed that the *Signal* app connects to at least three servers to finally connect with

No.	Time	Source	Destination	Protocol	Length	TCP Seg	Info
1	11:39:57	13.248.212.111	192.168.137.43	TLSv1.2	729	663	Application Data
2	11:39:57	13.248.212.111	192.168.137.43	TCP	729	663	TCP Retransmission
3	11:39:58	192.168.137.43	13.248.212.111	TCP	66	0	4145 → 443 [ACK]
4	11:39:58	192.168.137.43	13.248.212.111	TCP	78	0	TCP Dup ACK 3#1
5	11:39:58	192.168.137.43	13.248.212.111	TLSv1.2	122	56	Application Data
6	11:39:58	192.168.137.43	13.248.212.111	TLSv1.2	105	39	Application Data
7	11:39:58	13.248.212.111	192.168.137.43	TCP	66	0	443 → 41415 [ACK]
8	11:39:58	192.168.137.43	13.248.212.111	TLSv1.2	105	39	Application Data
9	11:39:58	13.248.212.111	192.168.137.43	TCP	66	0	443 → 41414 [ACK]
10	11:39:58	13.248.212.111	192.168.137.43	TCP	66	0	443 → 41415 [ACK]
11	11:39:58	13.248.212.111	192.168.137.43	TLSv1.2	101	35	Application Data
12	11:39:58	13.248.212.111	192.168.137.43	TLSv1.2	97	31	Encrypted Alert
13	11:39:58	13.248.212.111	192.168.137.43	TCP	66	0	443 → 41414 [FIN,
14	11:39:58	192.168.137.43	13.248.212.111	TCP	66	0	41414 → 443 [ACK]
15	11:39:58	192.168.137.43	13.248.212.111	TCP	66	0	41414 → 443 [ACK]
16	11:39:58	192.168.137.43	13.248.212.111	TCP	66	0	41414 → 443 [ACK]
17	11:39:58	13.248.212.111	192.168.137.43	TLSv1.2	101	35	Application Data
18	11:39:58	13.248.212.111	192.168.137.43	TLSv1.2	97	31	Encrypted Alert
19	11:39:58	13.248.212.111	192.168.137.43	TCP	66	0	443 → 41415 [FIN,
20	11:39:58	192.168.137.43	13.248.212.111	TCP	66	0	41415 → 443 [ACK]

Figure 5.4: User B is typing a message

the receiver or callee. When a user opens the app, it is connected to a *Signal* server. Next, the user opens a chat window of the receiver to place a voice call. Further, a user dials a call, the client app makes a DNS query to the STUN server¹ of the google (stun1.l.google.com) and TURN server² (turn1.whispersystems.org). In response to these DNS queries, the STUN server provides a google server like (64.233.161.127) and the DNS turn query provides a server IP of whispersystems.org/signal server like (52.47.185.9) as shown in the figure 5.5.

Stun protocol is used for UDP calls to connect the electronic devices behind the NAT. This resolves issues related to devices that are deployed between different NAT. Whereas Turn protocol is an extension to STUN used

¹STUN: Simple/session Traversal of UDP through NAT

²TURN: Traversal Using Relays around NAT

No.	Time	Source	Destination	Protocol	Length	TCP Seg Info
33	16:51:27	192.168.137.206	192.168.137.1	DNS	78	Standard query 0xaff2 A stun1.l.google.com
34	16:51:27	192.168.137.206	192.168.137.1	DNS	84	Standard query 0xc8b7 A turn1.whispersystems.org
35	16:51:27	192.168.137.206	192.168.137.1	DNS	84	Standard query 0xa377 A turn1.whispersystems.org
36	16:51:27	192.168.137.1	192.168.137.206	DNS	94	Standard query response 0xaff2 A stun1.l.google.com A 64.233.161.127
37	16:51:27	192.168.137.206	64.233.161.127	STUN	62	Binding Request
38	16:51:27	192.168.137.1	192.168.137.206	DNS	129	Standard query response 0xc8b7 A turn1.whispersystems.org CNAME turn-eu-west-3.whispersystems.org A 52.47.185.9
39	16:51:27	192.168.137.206	52.47.185.9	STUN	62	binding request
40	16:51:27	192.168.137.206	52.47.185.9	STUN	70	Allocate Request UDP
41	16:51:27	192.168.137.206	52.47.185.9	STUN	70	Allocate Request UDP
42	16:51:27	192.168.137.1	192.168.137.206	DNS	129	Standard query response 0xa377 A turn1.whispersystems.org CNAME turn-eu-west-3.whispersystems.org A 52.47.185.9
43	16:51:27	192.168.137.206	52.47.185.9	STUN	62	Binding Request
44	16:51:27	192.168.137.206	13.248.212.111	TCP	74	0 41506 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1400 SACK_PERM=1 TSval=2529434 TSecr=0 NS=256
45	16:51:27	13.248.212.111	192.168.137.206	TCP	74	0 443 → 41506 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1452 SACK_PERM=1 TSval=1075877044 TSecr=2529434 NS=128
46	16:51:27	192.168.137.206	13.248.212.111	TCP	66	0 41506 → 443 [ACK] Seq=1 Ack=1 Win=84224 Len=0 TSval=2529469 TSecr=1075877044
47	16:51:27	192.168.137.206	13.248.212.111	TLSv1.2	583	517 Client Hello
48	16:51:27	64.233.161.127	192.168.137.206	STUN	74	Binding Success Response XOR-MAPPED-ADDRESS: 39.41.208.228:61938
49	16:51:28	192.168.137.206	52.47.185.9	STUN	62	Binding Request
50	16:51:28	192.168.137.206	52.47.185.9	STUN	70	Allocate Request UDP
51	16:51:28	192.168.137.206	52.47.185.9	STUN	70	Allocate Request UDP
52	16:51:28	192.168.137.206	52.47.185.9	STUN	62	Binding Request

Figure 5.5: STUN and TURN servers

to relay the messages between two devices after a connection is established. It can also be said that binding is done by the STUN server, while further communication is relayed between two devices and performed through the TURN server. Patterns were observed while the audio call was established, hence 1388 bytes of reassembled PDUs are sent to the *Signal* server to indicate audio call activity. When binding request between the users is successful, it is observed that the private IP address of the other user is also visible. In the figure 5.6, the IP address of the receiver is 192.168.10.10, and the IP address of the initiating User A is 192.168.137.206.

The port numbers used by the TURN servers are 80 and 3478. In most cases, it was noticed that the public IP of the user is mapped with the TURN server. When the call is successfully established between the users, multiple UDP packets of Length 22 are found that were transmitted between both users as shown in figure 5.7.

No.	Time	Source	Destination	Protocol	Length	TCP Set Info
121	16:51:33	192.168.137.206	13.248.212.111	TCP	66	0 41503 → 443 [ACK] Seq=1 Ack=4344 Win=394 Len=0 TSval=2532427 TSecr=1075791416
122	16:51:33	192.168.137.206	13.248.212.111	TCP	78	0 [TCP Dup ACK 121#1] 41503 → 443 [ACK] Seq=1 Ack=4344 Win=394 Len=0 TSval=2532427 TSecr=1075791431 SLE=4165 SRE=
123	16:51:33	192.168.137.206	13.248.212.111	TLSv1.2	122	56 Application Data
124	16:51:33	13.248.212.111	192.168.137.206	TCP	66	0 443 → 41503 [ACK] Seq=4344 Ack=57 Win=2002 Len=0 TSval=1075791444 TSecr=2532445
125	16:51:34	192.168.137.206	52.47.185.9	STUN	166	CreatePermission Request XOR-PEER-ADDRESS: 192.168.10.10:37721 user: 1584618687:218262994 realm: whispersystems
126	16:51:34	192.168.137.206	52.47.185.9	STUN	166	CreatePermission Request XOR-PEER-ADDRESS: 192.168.10.10:37721 user: 1584618687:218262994 realm: whispersystems
127	16:51:34	192.168.137.206	192.168.10.10	STUN	138	Binding Request user: Ie5I:4uoy
128	16:51:34	192.168.10.10	192.168.137.206	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.10.2:61938
129	16:51:34	192.168.10.10	192.168.137.206	STUN	138	Binding Request user: 4uoy:Ie5I
130	16:51:34	192.168.137.206	192.168.10.10	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.10.10:37721
131	16:51:34	192.168.137.206	192.168.10.10	STUN	146	Binding Request user: Ie5I:4uoy
132	16:51:34	192.168.10.10	192.168.137.206	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.10.2:61938

Figure 5.6: Receiver's IP address

No.	Source	Destination	Protocol	Length	TCP Set Info
68	13.248.212.111	192.168.137.206	TCP	1454	1388 [TCP Out-Of-Order] 443 → 41506 [ACK] Seq=518 Ack=1389 Win=86784 Len=0 TSval=2529921 TSecr=1075877881
69	192.168.137.206	13.248.212.111	TCP	66	0 41506 → 443 [ACK] Seq=518 Ack=1389 Win=86784 Len=0 TSval=2529921 TSecr=1075877881
70	192.168.137.206	52.47.185.9	STUN	162	Allocate Request UDP user: 1584618687:218262994 realm: whispersystems.org with nonce
71	192.168.137.206	13.248.212.111	TCP	66	0 41506 → 443 [ACK] Seq=518 Ack=2451 Win=89600 Len=0 TSval=2529921 TSecr=1075877881
72	192.168.137.206	13.248.212.111	TCP	78	0 [TCP Dup ACK 71#1] 41506 → 443 [ACK] Seq=518 Ack=2451 Win=89600 Len=0 TSval=2529921 TSecr=1075877881 SLE=1389 SRE=2451
73	192.168.137.206	13.248.212.111	TCP	78	0 [TCP Dup ACK 71#2] 41506 → 443 [ACK] Seq=518 Ack=2451 Win=89600 Len=0 TSval=2529921 TSecr=1075877881 SLE=1389 SRE=1389
74	192.168.137.206	13.248.212.111	TLSv1.2	192	126 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
75	192.168.137.206	13.248.212.111	TLSv1.2	1454	1388 Application Data, Application Data, Application Data
76	192.168.137.206	13.248.212.111	TCP	1454	1388 11506 → 443 [ACK] Seq=2032 Ack=2451 Win=89600 Len=1388 TSval=2529932 TSecr=1075877884 [TCP segment of a reassembled PDU]
77	192.168.137.206	13.248.212.111	TCP	1454	1388 11506 → 443 [ACK] Seq=3420 Ack=2451 Win=89600 Len=1388 TSval=2529932 TSecr=1075877884 [TCP segment of a reassembled PDU]
78	192.168.137.206	13.248.212.111	TCP	1454	1388 11506 → 443 [ACK] Seq=4808 Ack=2451 Win=89600 Len=1388 TSval=2529932 TSecr=1075877884 [TCP segment of a reassembled PDU]
79	13.248.212.111	192.168.137.206	TCP	78	0 [TCP Dup ACK 59#1] 443 → 41506 [ACK] Seq=2451 Ack=518 Win=256512 Len=0 TSval=1075877142 TSecr=2529607 SLE=1 SRE=518
80	13.248.212.111	192.168.137.206	TCP	78	0 [TCP Dup ACK 59#2] 443 → 41506 [ACK] Seq=2451 Ack=518 Win=256512 Len=0 TSval=1075877143 TSecr=2529743 SLE=1 SRE=518
81	13.248.212.111	192.168.137.206	TCP	66	0 443 → 41506 [ACK] Seq=2451 Ack=644 Win=256512 Len=0 TSval=1075877144 TSecr=2529928
82	192.168.137.206	13.248.212.111	TLSv1.2	464	398 Application Data
175	52.47.185.9	192.168.137.206	STUN	102	CreatePermission Success Response
176	52.47.185.9	192.168.137.206	STUN	102	CreatePermission Success Response
177	52.47.185.9	192.168.137.206	STUN	102	CreatePermission Success Response
178	52.47.185.9	192.168.137.206	STUN	102	CreatePermission Success Response
179	52.47.185.9	192.168.137.206	STUN	102	CreatePermission Success Response
180	192.168.10.10	192.168.137.206	STUN	138	Binding Request user: 4uoy:Ie5I
181	192.168.137.206	192.168.10.10	STUN	106	Binding Success Response XOR-MAPPED-ADDRESS: 192.168.10.10:37721
182	192.168.10.10	192.168.137.206	UDP	64	37721 → 48179 Len=22
183	192.168.137.206	192.168.10.10	UDP	64	48179 → 37721 Len=22
184	192.168.137.206	192.168.10.10	UDP	64	48179 → 37721 Len=22
185	192.168.10.10	192.168.137.206	UDP	64	37721 → 48179 Len=22
186	192.168.137.206	192.168.10.10	UDP	64	48179 → 37721 Len=22
187	192.168.137.206	64.233.161.127	STUN	62	Binding Request
188	64.233.161.127	192.168.137.206	STUN	74	Binding Success Response XOR-MAPPED-ADDRESS: 39.41.208.228:61938

Figure 5.7: Patterns: User A calls to user B

When the call ends some ICMP packets are sent from caller to the server that shows the destination host is unreachable. Traffic patterns related to the indication of ending a call is shown in figure 5.8.

5.4.5 Call Received by User A (Callee)

When another user dials a call to the target device A, certain patterns are observed. Initially, the server sends a TCP segment of reassembled PDU to the Client with packets length 1454 bytes and payload size 1388 bytes. In response, the client sends 66 bytes of packets to the server. Later UDP

No.	Source	Destination	Protocol	Length	TCP Segm	Info
197	192.168.137.206	192.168.10.10	ICMP	109		Destination unreachable (Port unreachable)
198	192.168.137.206	13.248.212.111	TLSv1.2	1052	986	Application Data
199	52.47.185.9	192.168.137.206	STUN	110		Refresh Success Response lifetime: 0
200	192.168.137.206	52.47.185.9	ICMP	138		Destination unreachable (Port unreachable)
201	52.47.185.9	192.168.137.206	STUN	110		Refresh Success Response lifetime: 0
202	192.168.137.206	52.47.185.9	ICMP	138		Destination unreachable (Port unreachable)
203	13.248.212.111	192.168.137.206	TCP	66	0 443 → 41506	[ACK] Seq=3140 Ack=9457 Win=253312
204	13.248.212.111	192.168.137.206	TLSv1.2	174	108	Application Data

Figure 5.8: Patterns: Call Terminated

packets are observed flowing between the sender and the receiver. It was also noticed that the IP address of the caller is also captured during this activity. The UDP packets of 64 bytes with a payload size of 22 bytes are observed between the receiver and the caller. As shown in figure 5.9 and figure 5.10, Server's IP is 76.223.92.165, IP of User A is 192.168.137.252 and IP of User B is 192.168.10.3.

No.	Source	Destination	Protocol	Length	TCP Segment	Info
87	76.223.92.165	192.168.137.252	TCP	66	0 443 → 47959	[ACK]
88	192.168.137.252	76.223.92.165	TCP	74	0 47961 → 443	[SYN]
89	76.223.92.165	192.168.137.252	TLSv1.2	104	38	Application Data
90	76.223.92.165	192.168.137.252	TCP	1454	1388	443 → 47959 [ACK]
91	76.223.92.165	192.168.137.252	TCP	1454	1388	443 → 47959 [ACK]
92	76.223.92.165	192.168.137.252	TCP	1454	1388	443 → 47959 [ACK]
93	76.223.92.165	192.168.137.252	TCP	1454	1388	443 → 47959 [ACK]
94	76.223.92.165	192.168.137.252	TLSv1.2	408	342	Application Data,
95	192.168.137.252	76.223.92.165	TCP	78	0	[TCP Dup ACK 83#1]
96	192.168.137.252	76.223.92.165	TCP	66	0 47959 → 443	[ACK]
97	192.168.137.252	76.223.92.165	TCP	66	0 47959 → 443	[ACK]
98	192.168.137.252	76.223.92.165	TCP	66	0 47959 → 443	[ACK]

Figure 5.9: Target User A gets a call from user B

5.4.6 Media Messages

Media messages including sending pictures, videos, and stickers. These messages are sent from the android device to the other user. Traced files of media messages are captured and saved for the detailed inspection. We noticed that the bytes patterns are the same for all these activities. Hence, distinguishing

No.	Source	Destination	Protocol	Length	TCP Segment	Info
400	192.168.137.252	192.168.10.3	UDP	64	45475 → 39725	Len=22
401	192.168.10.3	192.168.137.252	UDP	64	39725 → 45475	Len=22
402	192.168.10.3	192.168.137.252	UDP	64	39725 → 45475	Len=22
403	192.168.137.252	192.168.10.3	UDP	64	45475 → 39725	Len=22
404	192.168.137.252	157.175.72.151	STUN	62		Binding Request
405	192.168.137.252	192.168.10.3	STUN	138		Binding Request user:
406	192.168.10.3	192.168.137.252	STUN	106		Binding Success Respo
407	157.175.72.151	192.168.137.252	STUN	126		Binding Success Respo
408	192.168.10.3	192.168.137.252	UDP	64	39725 → 45475	Len=22

Figure 5.10: UDP packets flow of a call

among these messages is difficult. The patterns include multiple groups of reassembled PDU that are seen delivered from client to server. The size of each PDU is 1454 with a payload size of 1388 bytes. It was also noticed that the client uses only one port to send messages to the server port 443. After sending the messages, the server sends an equal acknowledgment to the PDU list of size 108 bytes with a payload size of 42 bytes as shown in figure 5.11. All types of media messages mentioned above, we cannot distinguish which type of message is delivered from client to server based on bytes patterns. Moreover, the receiver IP is also not possible to capture. TCP segment of a reassembled PDU: When the payload of the message is large enough to be sent in one packet. TCP forms chunks of large data streams into small pieces called segments and applies TCP header on it. The IP header is then applied over it to be sent over the network.

No.	Source	Destination	Protocol	Length	TCP Segment	Info
52	192.168.137.45	13.35.183.48	TLSv1.2	184	118	Application Data
53	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=119 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]
54	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=1507 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]
55	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=2895 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]
56	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=4283 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]
57	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=5671 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]
58	192.168.137.45	13.35.183.48	TLSv1.2	1454	1388	Application Data [TCP segment of a reassembled PDU]
59	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=8447 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]
60	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=9835 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]
61	192.168.137.45	13.35.183.48	TCP	1454	1388	40451 → 443 [ACK] Seq=11223 Ack=1 Win=341 Len=1388 TSval=317050981 TSecr=313665589 [TCP segment of a reassembled PDU]

Figure 5.11: Media Messages

5.4.7 Video Calls

During the video call, Similar patterns of audio calls were observed as shown in figure 5.12. The only difference found in the length of UDP packets. UDP packet lengths varying from 800 bytes up to 1250 bytes.

No.	Time	Source	Destination	Protocol	Length	TCP Seg	Info
481	21:51:02	192.168.10.3	192.168.137.45	UDP	152		38492 → 46852 Len=110
482	21:51:02	192.168.137.45	192.168.10.3	UDP	152		46852 → 38492 Len=110
483	21:51:02	192.168.137.45	192.168.10.3	UDP	1214		46852 → 38492 Len=1172
484	21:51:02	192.168.10.3	192.168.137.45	UDP	152		38492 → 46852 Len=110
485	21:51:02	192.168.137.45	192.168.10.3	UDP	1214		46852 → 38492 Len=1172
486	21:51:02	192.168.10.3	192.168.137.45	UDP	152		38492 → 46852 Len=110
487	21:51:02	192.168.137.45	192.168.10.3	UDP	152		46852 → 38492 Len=110
488	21:51:02	192.168.137.45	192.168.10.3	UDP	1214		46852 → 38492 Len=1172
489	21:51:02	192.168.10.3	192.168.137.45	UDP	80		38492 → 46852 Len=38
490	21:51:02	192.168.137.45	192.168.10.3	UDP	152		46852 → 38492 Len=110
491	21:51:02	192.168.137.45	192.168.10.3	UDP	1214		46852 → 38492 Len=1172
492	21:51:02	192.168.10.3	192.168.137.45	UDP	152		38492 → 46852 Len=110
493	21:51:02	192.168.137.45	192.168.10.3	UDP	1215		46852 → 38492 Len=1173
494	21:51:03	192.168.137.45	192.168.10.3	UDP	152		46852 → 38492 Len=110

Figure 5.12: Video calls UDPatterns

As we have learned the behavior of all major activities, now we can filter out the application traffic by looking at these patterns. From a generic network traffic dump, we can identify the parties without knowing anything about parties beforehand. By looking at the patterns of protocol, we can filter

the *Signal* packets first and then further inspect them to find information about the user activities, type of communication, etc.

The summary of the analysis performed during this research is shown in table 5.2. The table summarizes all activities performed and the results found against them. During the research client and server behaviour in terms of payloads patterns is noticed.

Table 5.2: Traffic Characteristics of Signal Application on Android Device

Activities/ Events	Client/ Server	Observed Bytes Patterns	Payload Size
Opening the Signal Application	Client	Client sends encrypted data packets of 449, 372 and 127 bytes to the server	Actual payload size of these packets are 338,306,61 bytes respectively
	Server	Server acknowledges the client data with 261,261 and 137 bytes	Payload size of these packets are 195,195,71 bytes respectively
Target User A typing in Chat window	Client	Client sends encrypted data packets of 1181, 1182 and 1183 bytes to the server	Payload size of these packets is 1115,1116 and 1117 bytes respectively
	Server	Server responds to the client requests with 139 and 140 bytes	Payload size of these packets is 73 and 74 bytes respectively
User B typing in Chat window	Server	Server sends encrypted data packets of 729, 101 and 97 bytes to the Client at the target device	Payload size of these packets is 663,35 and 31 bytes respectively
	Client	Client at the target device responds to the server's requests with 122, 105 and 105 bytes	Payload size of these packets is 56,39 and 39 bytes respectively
User A as Caller	Client	Client sends TCP segment of a reassembled PDU of size 1454 bytes to server. UDP packets length 64	Payload size of TCP packets is 1388 bytes and UDP payload size is 22 bytes
	Server	Server responds the above TCP segments with 66 bytes.	Payload size is 0 bytes
Target User A as Receiver	Server	Server sends TCP segment of reassembled PDU to the Client with packets length 1454 bytes.UDP packets length 64	TCP payload size 1388 bytes and UDP payload size is 22 bytes
	Client	Client responds to the above TCP segments sent by the server with 66 bytes.	Payload size is 0 bytes
Media Messages by Target User A	Client	Client device sends Media messages like picture, video and stickers. TCP segment of a reassembled PDU of 1454 bytes are noticed.	Payload size of the packets is 1388 bytes.
	Server	After ack with 66 bytes, server sends 108 bytes of data to client.	Payload size of 108 bytes is 42 bytes
Video Call	Client-Server	UDP packet lengths varying from 800 bytes up to 1250 bytes are exchanged between client and server	Payload size varies
Call Termination	Client	Client sends 109 and 138 bytes of ICMP packets to the receiver and server respectively.	

5.5 Summary

In this chapter analysis and results achieved during the research is discussed.

In the following chapter validation and verification of the achieved results is provided.

Chapter 6

Validation and Verification

This chapter discusses the validation and verification of the analysis/results found against the user activities during the experimentation. During the study, major IP addresses of the chat servers are noticed explicitly while performing activities mentioned in previous chapter. In this chapter, the observed IP addresses were verified by applying the firewall rules to block them.

6.1 Blocking List of Servers

While performing all the above-mentioned activities, certain results are deduced. For example, during multiple activities, we observed a group of *Signal* servers that are found in response of the DNS query. While monitoring network packets during *Signal* app opening and connection times, it was noted that the app always connects to one of the servers mentioned in the DNS query i.e. "textsecure-serrvice.whispersystems.org". During the experimentation a group of eight type A servers were discovered in most DNS queries as

shown in figure 6.1. It has also been observed that an application establishes a connection with one of these servers using two random ports. The details of all those servers can be found in packets details in the Wireshark section as shown in figure 6.2.

No.	Source	Destination	Protocol	Length	Info
1	192.168.137.1	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
2	192.168.137.1	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
3	192.168.137.1	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
4	192.168.137.196	192.168.137.72	DNS	97	Standard query 0x5c72 A textsecure-service.whispersystems.org
5	192.168.137.196	192.168.137.72	DNS	97	Standard query 0x4e74 A textsecure-service.whispersystems.org
6	192.168.137.72	192.168.137.196	DNS	225	Standard query response 0x5c72 A textsecure-service.whispersystems.org A 52.207.41.59 A 100.24.0.111 A 54.175.47.110 A 34.196.69.69 A
7	192.168.137.196	52.207.41.59	TCP	74	40641 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1400 SACK_PERM=1 TSval=229071609 TSecr=0 NS=256
8	192.168.137.72	192.168.137.196	DNS	225	Standard query response 0x4e74 A textsecure-service.whispersystems.org A 54.175.47.110 A 34.225.196.214 A 3.228.254.81 A 34.196.69.69

Figure 6.1: DNS query textsecure-service.whispersystems.org

```

Authority RRs: 0
Additional RRs: 0
Queries
  textsecure-service.whispersystems.org: type A, class IN
    Name: textsecure-service.whispersystems.org
    [Name Length: 37]
    [Label Count: 3]
    Type: A (Host Address) (1)
    Class: IN (0x0001)
Answers
  > textsecure-service.whispersystems.org: type A, class IN, addr 52.207.41.59
  > textsecure-service.whispersystems.org: type A, class IN, addr 100.24.0.111
  > textsecure-service.whispersystems.org: type A, class IN, addr 54.175.47.110
  > textsecure-service.whispersystems.org: type A, class IN, addr 34.196.69.69
  > textsecure-service.whispersystems.org: type A, class IN, addr 3.228.254.81
  > textsecure-service.whispersystems.org: type A, class IN, addr 3.222.249.138
  > textsecure-service.whispersystems.org: type A, class IN, addr 54.175.130.206
  > textsecure-service.whispersystems.org: type A, class IN, addr 34.225.196.214

```

Figure 6.2: Type A, Server Addresses

The *Signal* app attempts to connect either with the first server using one of the two ports or with the first two servers using any of the two ports mentioned in the list of the servers. It is also noticed that one port is used to establish a connection for indicating typing patterns while the other port is used to send the actual data in the context of text messages. There were almost 10 servers found that were used to connect the client with the server. Chat servers found during the study are already shown in the table 5.1 of

previous chapter. It is also worth mentioning that Servers addresses are common in each list. However, the order of these addresses may be changed.

To confirm the results servers found in the study were blocked. Relevant rules were applied on a firewall to block them to see the behavior of the client smartphone app. With the help of aliases, we put all the chat servers shown in table 5.2 of previous chapter in a firewall rule and blocked them. Similarly, other activities mentioned in the previous section could also be verified. In our experiments, the firewall always gives us an edge to control as and monitor the ongoing activities within the network.

6.2 Blocking Chat Servers

The servers that were discovered in the study that are repeated in each list mentioned in table 5.1. In the current network environment that implements the pfSense Software Firewall, blocking these servers could show more servers except these in the list. Blocking these servers would let us know the connectivity pattern of the client application. To look at this behavior, an alias of these servers is created and used to block traffic on LAN and WAN through rules. In response to it, the client app of *Signal* was not able to connect to these servers and the messages were not sending as depicted in figure 6.3.

Signal servers use port 443 to connect to the client which cannot be blocked as other apps used 443 for communication purposes. 443 is an SSL communication port and is highly resistant to eavesdropping and interception. Remote servers providing services using this port are trustworthy and verified without any doubt. The web servers that accept and establish secure

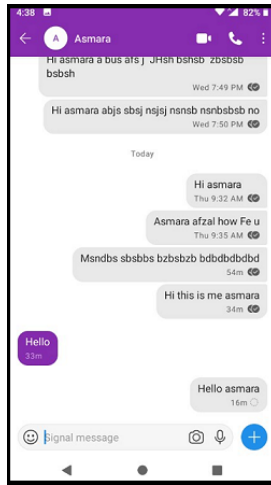


Figure 6.3: Message sending Failed

connections listen to this port for connections from web browsers desiring strong communication security. Hence, blocking these servers would help to stop the *Signal* app services. It cannot be confirmed if this will work in scenarios other than those targeted by this work.

Application behavior on a mobile device is as follows. In the figure 6.3, it took almost 16 minutes to deliver the text message, but it failed. Meanwhile, the app tried to connect all the above-mentioned chat servers. Wireshark based analysis shows the details of these servers shown in figure 6.4. When the client app tried to connect to these servers to deliver a message to another user, the mentioned servers did not respond or a 0 is sent in response to the client requests.

It was noted that all services were working fine when there was no restriction on the IP addresses. Firewall rules that we used to block/delay the chat activities are mentioned in table 6.1. With the help of the rules and list of servers, chat service was disrupted. The device of Target User A tried to

Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
3.222.249.138	443	39	2886	0	0	39	
3.228.254.81	443	39	2886	0	0	39	
34.196.69.69	443	54	3996	0	0	54	
34.225.196.214	443	49	3626	0	0	49	
52.207.41.59	443	39	2886	0	0	39	
54.175.47.110	443	34	2516	0	0	34	
54.175.130.206	443	39	2886	0	0	39	
54.175.149.136	443	49	3626	0	0	49	
100.24.0.111	443	64	4736	0	0	64	
172.217.19.163	443	6	520	0	0	6	
192.168.0.6	853	87	13 k	41	7918	46	

Figure 6.4: Response from blocked servers

send messages to other servers but failed to do so. Figure 6.5 shows that the client application tried its best to connect the main servers to send messages. However, the connection is denied with each of them as shown in black color.

Table 6.1: Firewall Rules

Step	Protocol	Source Port	Destination Port	Action	Observation
1	Tcp	Any	Any	Default allow LAN to any rule	All services work smoothly
2	Tcp	Any	Any	Block the entire list of IP addresses of chat servers	Message sending failed

6.3 List of Call Servers

Many trace files are captured and saved while calls are made between users. As discussed, in the previous section, the IP address of the receiver involved in the call is also captured during call activity, unlike messages activity. During a call that is established between two users, it was noticed that at least three servers are involved. The initial Server authenticates the Client and in the DNS query response, two more IP addresses of servers are sent

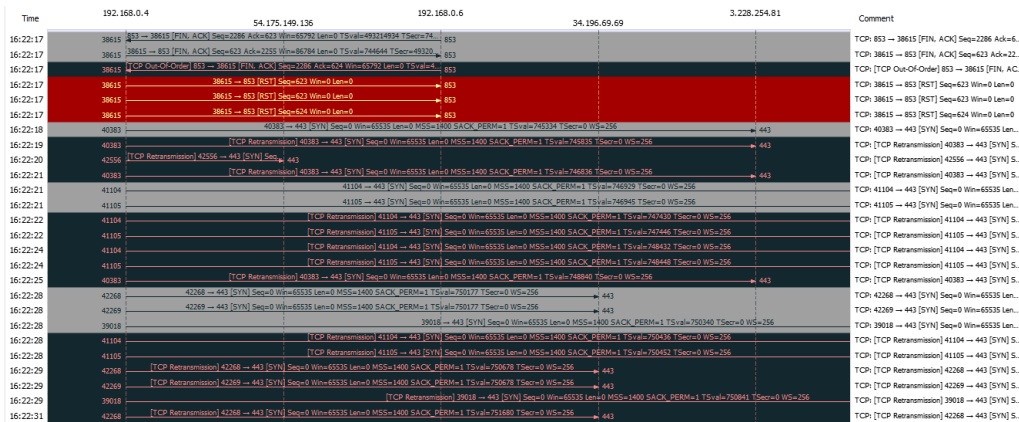


Figure 6.5: Wireshark-based Graph of blocked servers

Table 6.2: Call Servers and Receivers IP Found – Needs to be Blocked

Trace Files	Client's IP	Initial Server	Google's Stun Server	Signal's TURN Server	Receiver's Private IP	Receiver's Public IP
1	192.168.137.34	13.248.212.111	64.233.161.127	52.47.185.9	192.168.0.5	
2	192.168.137.206	13.248.212.111	64.233.161.127	52.47.185.9	192.168.10.10	
3	192.168.137.206	13.248.212.111	64.233.161.127	52.47.185.9	192.168.10.10	
4	192.168.137.25	76.233.92.165	64.233.161.127	52.47.185.9	192.168.0.6	182.182.252.29
5	192.168.137.25	13.248.212.111 76.233.92.165	64.233.161.127	52.47.185.9	192.168.0.6	
6	192.168.137.25	13.248.212.111 76.233.92.165 74.125.133.188	64.233.161.127	52.47.185.9	192.168.0.3	
7	192.168.137.43	13.248.212.111	64.233.161.127	52.47.185.9	192.168.0.6	182.182.252.29

to the client for further communication. The client connects with the first server of Google that helps to find the receiver's IP address. While server 2 helps to relay the communication between the Caller and receiver. And finally, the client is connected to the desired receiver as shown in the table 6.2.

6.4 Crime Scene Reconstruction

The details or results found during this study can help in investigating the cases involving live traffic monitoring and capturing based on *Signal* app. Forensics examiner can deduce the results of on-going calls, messages, and typing patterns. The communicating parties can also be identified with the IP addresses while calling each other. Forensic examiner needs to get access to the target network with the help of the specific organization. Traffic dumps of the ongoing traffic are required to be captured at a certain time of suspect's activities. With the help of results deduced in the section Results and Analysis, specific traffic can be filtered out among the ongoing traffic. Specific activities with the help of fixed patterns could be distinguished. Further information on the users with captured IP addresses could also be verified from ISPs.

6.5 A Case Study

The following case study is presented below to better understand the the proposed approach. To support the device forensics, network forensics patterns could help in proving evidences. We can correlate the evidences collected from device and networks to infer the results. We analyzed real time application's behavior which can be helpful in some way for the investigator. In lab, we had controlled environment to study and analyze the behavior of the application's ports and IP ranges. These can be further used in key word searching.

Assumptions:

1. Controlled environment, admin has control over network and firewall.
2. Logs are being maintained according to the specific time duration.
3. After an incident, investigators can get the firewall logs to analyze the performed activities

Use Case: Sensitive information in an organization got leaked. It got noticed at April 5, 2020. The organization does not allow any digital gadgets or smart phones due to security purposes. So, there is a notion the data got leaked from within the organization using the organization network. To trace the activity, firewall logs can help investigating the activities. Assuming the log policy is implemented in a way that we can get information about devices, source IP, destination IP, payload sizes, services, protocols and timestamps. We can use the study of the behavior analysis of the application to interpret the logs. Or Signal app analysis can support in deducing the activities and involved users. The organization can also get more accurate information from ISPs as well.

6.6 Case study:Data leakage using Media messages

Confirming and Identifying the source of data leakage using Signal app through behaviour analysis.

Timeline:

1. Photographs leaked and received by users between 21:23 PM to 21:25

PM.

2. Traffic dumps for duration of interest: Examine the organization’s traffic for same duration.

No.	Arrival Time	Source	Destination	TCP Segme	Protocol	Length	Info
1	Apr 5, 2020 21:23:59.089130000..	192.168.137.45	192.168.137.1		DNS	97	Standard query 0xfdc3 A textsecure-service.whispersystems.org
2	Apr 5, 2020 21:23:59.089132000..	192.168.137.45	192.168.137.1		DNS	97	Standard query 0xfbc2 A textsecure-service.whispersystems.org
3	Apr 5, 2020 21:23:59.106846000..	192.168.137.1	192.168.137.45		DNS	113	Standard query response 0xfdc3 A textsecure-service.whispersystems.org A 76.223.92.1

Figure 6.6: DNS Query and Response

3. Identify and analyse the traffic for particular activity.

No.	Arrival Time	Source	Destination	TCP Segme	Protocol	Length	Info
91	Apr 5, 2020 21:24:14.401940000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=1957 Ack=153 Win=84224 Len=1388 TSval=316401116 TSecr=31582358
92	Apr 5, 2020 21:24:14.401957000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=3345 Ack=153 Win=84224 Len=1388 TSval=316401116 TSecr=31582358
93	Apr 5, 2020 21:24:14.402280000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=4733 Ack=153 Win=84224 Len=1388 TSval=316401116 TSecr=31582358
94	Apr 5, 2020 21:24:14.402380000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=6121 Ack=153 Win=84224 Len=1388 TSval=316401116 TSecr=31582358
95	Apr 5, 2020 21:24:14.427631000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=7509 Ack=153 Win=84224 Len=1388 TSval=316401116 TSecr=31582358
96	Apr 5, 2020 21:24:14.427633000..	192.168.137.45	13.35.183.42	1388	TLSv1.2	1454	Application Data [TCP segment of a reassembled PDU]
97	Apr 5, 2020 21:24:14.427974000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=10285 Ack=153 Win=84224 Len=1388 TSval=316401116 TSecr=31582358
98	Apr 5, 2020 21:24:14.428194000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=11673 Ack=153 Win=84224 Len=1388 TSval=316401116 TSecr=31582358
99	Apr 5, 2020 21:24:14.455628000..	13.35.183.42	192.168.137.45	0	TCP	66	443 → 44818 [ACK] Seq=153 Ack=569 Win=30288 Len=0 TSval=315823594 TSecr=316401100
100	Apr 5, 2020 21:24:14.460791000..	13.35.183.42	192.168.137.45	69	TLSv1.2	135	Application Data
101	Apr 5, 2020 21:24:14.519471000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=13061 Ack=153 Win=84224 Len=1388 TSval=316401169 TSecr=31582358
102	Apr 5, 2020 21:24:14.519764000..	192.168.137.45	13.35.183.42	1388	TCP	1454	44818 → 443 [ACK] Seq=14449 Ack=153 Win=84224 Len=1388 TSval=316401169 TSecr=31582358

Figure 6.7: Media messages patterns are detected

4. Now, to check the host to which this source IP belongs, the investigator can check the DHCP lease in organization’s firewall which shows the complete list of IP addresses

Date	Source IP	Destination IP	Timestamps	Session	Protocol	Packet type	Packet length
5-04-2020	192.269.137.45	192.269.137.1	21:23:59 PM	C→S	DNS	DNS	97
			21:23:59 PM	S→C	DNS	DNS (Response)	113
	13.35.183.42	192.269.137.45	21:24:14 PM	C→S	TCP	TCP Segments- 1454 bytes payload size	1388
			21:24:14 PM	S→C	TCP	ACK	66

Figure 6.8: Temporal Analysis

6.7 Summary

In this chapter, we have summarized important results and applied the validations to verify our findings. List of chat server are blocked in the firewall rules to check out the application behaviour. In the next chapter conclusion and future work is discussed.

Chapter 7

Conclusion & Future Work

Chapter 7 concludes the presented thesis and highlights potential future research directions. It describes different research prospects of our research and identifies open research problems that still need to be solved by the research community.

7.1 Conclusion

In this research, we analyzed the encrypted network traffic of the *Signal* app. The methodology we followed in this research is demonstrated by using *Signal* app as a case study. Without knowledge of communication protocol and security architecture, we observed possible behavior from encrypted network traffic of *Signal* app which can be exploited by an investigator to investigate a case involving *Signal* app. A new idea of using firewall in such studies was demonstrated which supported our methodology to reveal obscured call connectivity scenarios of *Signal* app. Further, proposed results on *Signal*

traffic pattern can facilitate to correctly identify the events of *Signal* chats, voice and video calls and also revealed IP addresses of involved parties, and helped to list down the involved servers etc. It is worthful to note down that proposed strategy is equally applicable for *Signal* and as well as for other existing social media apps. In future, it can be employed to investigate with other apps after slight fine-tuning in IP ranges, ports of associated devices.

7.2 Limitation & Future Work

The major limitation of the proposed work is the lack of device forensic analysis. Even though, we extensively studied the encrypted network traffic of the *Signal* app and drew the correct and significant detection of the app's behavior in the context of supportive evidence for forensic investigation. However, device forensic analysis helps to provide a complete solution in terms of investigation; therefore device forensic would be targeted in the future. The second limitation consists of version dependency. Like other existing apps forensics approaches are dependent on the specific version of the application. Hence, the proposed work is also version dependent. In the case of the upgraded app like extensive modification of the structure, byte, or patterns may change and conflict results. In the future, we will target a GUI based solution for other existing IM based apps by providing slight tuning features in terms of ports and associated servers IPs, etc. to generalize the proposed methodology.

7.3 Summary

This chapter has presented the conclusion of the thesis. Furthermore, it describes potential future directions in which this thesis can be extended for further research work.

Bibliography

- [1] H. Arshad, A. Jantan, G. K. Hoon, and I. O. Abiodun, “Formal knowledge model for online social network forensics,” *Comput. Secur.*, vol. 89, p. 101675, Feb. 2020, doi: 10.1016/j.cose.2019.101675.
- [2] “Signal-Home.”<https://signal.org/> (accessed Apr.18,2020)
- [3] T. Perrin, “The XEdDSA and VEdDSA Signature Schemes,” *Signal*, p. 14, 2016, Accessed: Apr. 18, 2020. [Online]. Available: <https://signal.org/docs/specifications/xeddsa/>.
- [4] “Signal »Specifications »The X3DH Key Agreement Protocol.” <https://signal.org/docs/specifications/x3dh/> (accessed Apr. 18, 2020).
- [5] “Signal »Specifications »The Double Ratchet Algorithm.” <https://signal.org/docs/specifications/doubleratchet/> (accessed Apr. 18, 2020).
- [6] M. Marlinspike and T. Perrin, “The Sesame Algorithm: Session Management for Asynchronous Message Encryption,” *Signal*, p. 17, 2017, Accessed: Apr. 18, 2020. [Online]. Available: <https://signal.org/docs/specifications/sesame/>.

- [7] L. F. Sikos, "Packet analysis for network forensics: A comprehensive survey," *Forensic Sci. Int. Digit. Investig.*, vol. 32, p. 200892, Mar. 2020, doi: 10.1016/j.fsidi.2019.200892.
- [8] R. Montasari, R. Hill, V. Carpenter, and F. Montaseri, "Digital Forensic Investigation of Social Media, Acquisition and Analysis of Digital Evidence," *Int. J. Strateg. Eng.*, vol. 2, no. 1, pp. 52–60, Nov. 2018, doi: 10.4018/ijose.2019010105.
- [9] S. Schroder, M. Huber, D. Wind, and C. Rottermanner, "When SIGNAL hits the Fan: On the Usability and Security of State-of-the-Art Secure Mobile Messaging," Sep. 2017, doi: 10.14722/eurosec.2016.23012.
- [10] H. Zhang, L. Chen, and Q. Liu, "Digital Forensic Analysis of Instant Messaging Applications on Android Smartphones," in *2018 International Conference on Computing, Networking and Communications, ICNC 2018*, Jun. 2018, pp. 647–651, doi: 10.1109/ICCNC.2018.8390330.
- [11] F. A. Awan, "Forensic examination of social networking applications on smartphones," in *Proceedings - 2015 Conference on Information Assurance and Cyber Security, CIACS 2015*, Jan. 2016, pp. 36–43, doi: 10.1109/CIACS.2015.7395564.
- [12] N. Al Mutawa, I. Baggili, and A. Marrington, "Forensic analysis of social networking applications on mobile devices," in *Proceedings of the Digital Forensic Research Conference, DFRWS 2012 USA*, Aug. 2012, vol. 9, pp. S24–S33, doi: 10.1016/j.diin.2012.05.007.

- [13] Knox, Shawn, et al. "What's really 'Happning'? A forensic analysis of Android and iOS Happn dating apps." *Computers Security*, 2020: 101833. doi: 10.1016/j.cose.2020.101833
- [14] C. Anglano, M. Canonico, and M. Guazzone, "Forensic analysis of Telegram Messenger on Android smartphones," *Digit. Investig.*, vol. 23, pp. 31–49, Dec. 2017, doi: 10.1016/j.diin.2017.09.002.
- [15] D. Walnycky, I. Baggili, A. Marrington, J. Moore, and F. Breiting, "Network and device forensic analysis of android social-messaging applications," in *Proceedings of the Digital Forensic Research Conference, DFRWS 2015 USA*, Aug. 2015, vol. 14, pp. S77–S84, doi: 10.1016/j.diin.2015.05.009.
- [16] M. N. Yusoff, A. Dehghantanha, and R. Mahmood, "Network Traffic Forensics on Firefox Mobile OS: Facebook, Twitter, and Telegram as Case Studies," in *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*, Elsevier Inc., 2017, pp. 63–78.
- [17] M. A. K. Sudozai, S. Saleem, W. J. Buchanan, N. Habib, and H. Zia, "Forensics study of IMO call and chat app," *Digit. Investig.*, vol. 25, pp. 5–23, Jun. 2018, doi: 10.1016/j.diin.2018.04.006.
- [18] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Can't you hear me knocking: Identification of user actions on android apps via traffic analysis," in *CODASPY 2015 - Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, Mar. 2015, pp. 297–304, doi: 10.1145/2699026.2699119.

- [19] H. Arshad, E. Omlara, I. O. Abiodun, and A. Aminu, “A semi-automated forensic investigation model for online social networks,” *Comput. Secur.*, vol. 97, Oct. 2020, doi: 10.1016/j.cose.2020.101946.
- [20] N. Clarke, F. Li, and S. Furnell, “A novel privacy preserving user identification approach for network traffic,” *Comput. Secur.*, vol. 70, pp. 335–350, Sep. 2017, doi: 10.1016/j.cose.2017.06.012.
- [21] R. Y. Patil and S. R. Devane, “Network Forensic Investigation Protocol to Identify True Origin of Cyber Crime,” *J. King Saud Univ. - Comput. Inf. Sci.*, Dec. 2019, doi: 10.1016/j.jksuci.2019.11.016.
- [22] X. Lin, T. Chen, T. Zhu, K. Yang, and F. Wei, “Automated forensic analysis of mobile applications on android devices,” in *Proceedings of the Digital Forensic Research Conference, DFRWS 2018 USA*, Jul. 2018, vol. 26, pp. S59–S66, doi: 10.1016/j.diin.2018.04.012.
- [23] C. Anglano, M. Canonico, and M. Guazzone, “The Android Forensics Automator (AnForA): A tool for the Automated Forensic Analysis of Android Applications,” *Comput. Secur.*, vol. 88, Jan. 2020, doi: 10.1016/j.cose.2019.101650.
- [24] H. Arshad, A. Jantan, G. K. Hoon, and A. S. Butt, “A multi-layered semantic framework for integrated forensic acquisition on social media,” *Digit. Investig.*, vol. 29, pp. 147–158, Jun. 2019, doi: 10.1016/j.diin.2019.04.002.
- [25] G. B. Satria, P. T. Daely, and S. Y. Shin, “Android forensics analysis: Private chat on social messenger,” in *International Conference on Ubiquitous Computing and Communications*, Oct. 2019, pp. 1–6, doi: 10.1109/ICCC47720.2019.9007002.

- uitous and Future Networks, ICUFN, Aug. 2016, vol. 2016-Augus, pp. 430–435, doi: 10.1109/ICUFN.2016.7537064.
- [26] J. I. James and P. Gladyshev, “Challenges with Automation in Digital Forensic Investigations,” Mar. 2013, Accessed: Nov. 07, 2020. [Online]. Available: <http://arxiv.org/abs/1303.4498>.
- [27] C. Ntantogian, D. Apostolopoulos, G. Marinakis, and C. Xenakis, “Evaluating the privacy of Android mobile applications under forensic analysis,” *Comput. Secur.*, vol. 42, pp. 66–76, May 2014, doi: 10.1016/j.cose.2014.01.004.
- [28] V. V. Rao and A. S. N. Chakravarthy, “Forensic analysis of android mobile devices,” 2016, doi: 10.1109/ICRAIE.2016.7939540.
- [29] A. Ababneh, M. A. Awwad, and M. I. Al-Saleh, “IMO forensics in Android and windows systems,” in 2017 8th International Conference on Information, Intelligence, Systems and Applications, IISA 2017, Mar. 2018, vol. 2018-Janua, pp. 1–6, doi: 10.1109/IISA.2017.8316377.
- [30] F.-C. Tsai, E.-C. Chang, and D.-Y. Kao, “WhatsApp network forensics: Discovering the communication payloads behind cybercriminals,” Apr. 2018, pp. 1–1, doi: 10.23919/icact.2018.8323881.
- [31] K. Rathi, U. Karabiyik, T. Aderibigbe, and H. Chi, “Forensic analysis of encrypted instant messaging applications on Android,” 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding, vol. 2018-January, pp. 1–6, 2018, doi: 10.1109/ISDFS.2018.8355344.

- [32] S. Wu, Y. Zhang, X. Wang, X. Xiong, and L. Du, "Forensic analysis of WeChat on Android smartphones," *Digit. Investig.*, vol. 21, pp. 3–10, 2017, doi: 10.1016/j.diin.2016.11.002.
- [33] F. Karpisek, I. Baggili, and F. Breiting, "WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages," *Digit. Investig.*, vol. 15, no. October, pp. 110–118, 2015, doi: 10.1016/j.diin.2015.09.002.
- [34] C. Anglano, "Forensic analysis of whats app messenger on Android smartphones," *Digit. Investig.*, vol. 11, no. 3, pp. 201–213, 2014, doi: 10.1016/j.diin.2014.04.003.
- [35] M. A. K. Sudozai, N. Habib, S. Saleem, and A. A. Khan, "Signatures of Viber Security Traffic," *J. Digit. Forensics, Secur. Law*, 2017, doi: 10.15394/jdfsl.2017.1477.
- [36] M. A. K. Sudozai and S. Saleem, "Profiling of secure chat and calling apps from encrypted traffic," in *Proceedings of 2018 15th International Bhurban Conference on Applied Sciences and Technology, IB-CAST 2018*, Mar. 2018, vol. 2018-Janua, pp. 502–508, doi: 10.1109/IB-CAST.2018.8312271.
- [37] D. Barradas, T. Brito, D. Duarte, N. Santos, and L. Rodrigues, "Forensic analysis of communication records ofweb-based messaging applications from physical memory," in *ICETE 2017 - Proceedings of the 14th International Joint Conference on e-Business and Telecommunications*, 2017, vol. 4, pp. 43–54, doi: 10.5220/0006396100430054.

- [38] D. Quick and K. K. R. Choo, "Pervasive social networking forensics: Intelligence and evidence from mobile device extracts," *J. Netw. Comput. Appl.*, vol. 86, pp. 24–33, May 2017, doi: 10.1016/j.jnca.2016.11.018.
- [39] S. Rajasekar, P. Philominathan, and V. Chinnathambi, "Research Methodology", 2013. [Online]. Available: <http://arxiv.org/pdf/physics/0601009.pdf>. [Accessed: 08-Oct-2019].
- [40] W. C. Booth, G. G. Colomb, and J. M. Williams, "The Craft of Research." [Online]. Available: http://sir.spbu.ru/en/programs/master/master_program_in_international_relations/digital_library/Book_Research_seminar_by_Booth.pdf. [Accessed: 08-Oct-2019].
- [41] C. Woody, "Chapter 3: Research Methodology," 2001. [Online]. Available: https://shodhganga.inflibnet.ac.in/bitstream/10603/2026/16/16_chapter_3.pdf. [Accessed: 09-Oct-2019].