

# **NETWORK TRAFFIC ANALYSIS OF TWITTER**



By

**ALIA UMRANI**

**273881-MS (IS)-11-2018**

Supervisor

**DR. YOUSRA JAVED**

**DEPARTMENT OF COMPUTING**

A thesis submitted in partial fulfillment of the requirements for the degree of

Masters in Information Security (MSIS)

In

**School of Electrical Engineering and Computer Science**

**National University of Sciences and Technology (NUST)**

**Islamabad, Pakistan**

**(August 2020)**

## Approval

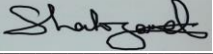
It is certified that the contents and form of the thesis entitled "Network Traffic Analysis of Twitter" submitted by Alia Umrani have been found satisfactory for the requirement of the degree

Advisor : Dr. Yousra Javed

Signature: 

Date: 30-Jul-2020

Committee Member 1: Dr. Shahzad Saleem

Signature: 

Date: 30-Jul-2020

Committee Member 2: Dr. Dr Hasan Tahir

Signature: 

Date: 30-Jul-2020

Committee Member 3: Mehdi Hussain

Signature: 

Date: 30-Jul-2020

## **Certificate of Originality**

I hereby declare that this submission titled “Network Traffic Analysis of Twitter” 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: Alia Umrani

Student Signature:  \_\_\_\_\_

## THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled “Network Traffic Analysis of Twitter” written by Alia Umrani, (Registration No 00000273881), 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 fulfilment 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. Yousra Javed

Date: 30-Jul-2020

Signature (HOD) : \_\_\_\_\_

Date: \_\_\_\_\_

Signature (Dean/Principal): \_\_\_\_\_

Date: \_\_\_\_\_

## **Dedication**

*This dissertation is dedicated to my parents and family members for their unconditional support and prayers throughout the period and respected teachers for their guidance.*

## **Acknowledgement**

I am grateful to Almighty Allah for the blessings throughout this journey. With His utmost support I am able to achieve this goal.

I would like to thank my father, mother, brother and other family members for their motivation and support which helped me in overcoming the hurdles while pursuing my MS degree. I would also express gratitude to my supervisor Dr. Yousra Javed for her guidance and valuable suggestions which improved this work in every manner and GEC members Dr. Shahzad Saleem, Dr. Hasan Tahir and Dr. Mehdi Hussain for their valuable time and contributions.

Despite all the assistance provided by my supervisor and committee members, I take the responsibility for any errors and omissions which may unwittingly remain.

# Table of Contents

INTRODUCTION .....	1
1.1. Introduction.....	1
1.2 Historical Background .....	1
1.2.1 Digital Forensics .....	1
1.2.2. Categories of Digital Forensics.....	2
1.2.3 Network Forensics and its Challenges: .....	4
1.2.4 Overview of Twitter application .....	6
1.3 Motivation.....	7
1.4 Problem Statement .....	7
1.5 Research Questions.....	8
1.6 Thesis Outline .....	8
1.7 Summary .....	8
LITERATURE REVIEW .....	9
2.1 Introduction.....	9
2.2 Methodologies of Network Forensics .....	9
2.3 Network forensics analysis of social media applications.....	10
2.4 Network forensics analysis and security tools .....	13
2.5 Summary.....	14
RESEARCH METHODOLOGY .....	15
3.1 Introduction.....	15
3.2 The Research Method .....	15
3.3 The proposed Framework .....	20
3.4 Summary .....	21
EXPERIMENTAL SETUP.....	22
4.1 Introduction.....	22
4.2 Tools and Technologies .....	22
4.3 Device Configurations and Specifications .....	22
4.4 Environmental Setup.....	23
4.4.1 Setup with firewall deployment .....	23
4.4.1.1 The firewall.....	23
4.4.1.2 The WAN Switch.....	25
4.4.1.3 Wireless Access Point/ Wi-Fi router.....	25

4.4.1.4	Configurations.....	26
4.4.2	Setup without firewall deployment .....	27
4.5	Smartphone Setup .....	28
4.6	Summary .....	29
<b>ANALYSIS OF APPLICATION AND RESULTS .....</b>		<b>30</b>
5.1	Introduction.....	30
5.2	User activities.....	30
5.3	Fixed packet length observation .....	30
5.4	Port range identification.....	31
5.5	IP range identification.....	32
5.5.1	Login Server.....	32
5.5.2	Opening an image .....	38
5.5.3	Text Message .....	42
5.5.4	Media as a message.....	42
5.6	Traffic behaviour analysis.....	44
5.6.1	Login .....	44
5.6.2	Logout .....	48
5.6.3	Opening an image .....	50
5.6.4	Sending a text message .....	52
5.6.5	Sending media as a message .....	53
5.6.6	Playing a video.....	55
5.7	Results.....	56
5.8	Application of the research .....	57
5.9	Summary .....	60
<b>CONCLUSION AND FUTURE SCOPE.....</b>		<b>61</b>
6.1	Introduction.....	61
6.2	Conclusion .....	61
6.3	Future Scope .....	62
<b>REFERENCES .....</b>		<b>63</b>



## List of Tables

Table 1. Network Forensics Methodologies .....	10
Table 2. Related work Limitations.....	12
Table 3. Traffic Classification Tools .....	13
Table 4. Case-I Description .....	18
Table 5. Case-II Description .....	18
Table 6. Case-III Description.....	19
Table 7. Configurations .....	26
Table 8. User Activities Performed on Twitter.....	30
Table 9. Packet length without firewall deployment .....	31
Table 10. Packet length with firewall deployment .....	31
Table 11. Server Range for Accessing Twitter .....	37
Table 12. Parallel connections .....	38
Table 13. Server Range for Sending a text in DMs.....	43
Table 14. Server Range for Sending Media in DMs.....	43
Table 15. Patterns for Logout Activity .....	50
Table 16. Patterns for Opening an Image .....	51
Table 17. Patterns for Sending a Text Message.....	53
Table 18. Patterns for Sending a Media.....	56
Table 19. Patterns for Playing a Video .....	56
Table 20. Results .....	57

# List of Figures

Figure 1. Digital Forensics Model .....	2
Figure 2. Categories of Digital Forensics .....	3
Figure 3. The OSCAR Methodology .....	5
Figure 4. The Research Process .....	16
Figure 5. Case Study Method.....	17
Figure 6. NIST guide for Digital Forensics Process .....	17
Figure 7. Analysis Framework.....	21
Figure 8. Environmental Setup with Firewall.....	23
Figure 9. Firewall Dashboard.....	24
Figure 10. WAN Interface Settings of Firewall.....	25
Figure 11. LAN Interface Settings of Firewall .....	26
Figure 12. Environmental Setup to Analyse Traffic Behaviour .....	28
Figure 13. Identification of Port Range .....	32
Figure 14. Observed Range-I for Login.....	32
Figure 15. Flow Graph for Range-I IP Addresses.....	33
Figure 16. Alias Created for Range-I IP Addresses.....	33
Figure 17. LAN Rule to Block Range-I IP Addresses.....	33
Figure 18. Error Packets after Blocking Range-I.....	34
Figure 19. Observed Range-II for Login .....	34
Figure 20. Failed Connection with Range-II and Observed Range-III.....	35
Figure 21. Failed Connection with Range-III .....	35
Figure 22. Error Packets for Blocked IP Addresses.....	35
Figure 23. Observed Range-V .....	35
Figure 24. Error Packets for Blocked IP Addresses.....	36
Figure 25. Server Infrastructure for Login.....	37
Figure 26. Twitter Infrastructure with Parallel Connection .....	38
Figure 27. Connection Established with 93.184.220.70 .....	39
Figure 28. Connection Established with 192.229.220.133 .....	39
Figure 29. Failed Connection with 93.184.220.70 and 192.229.220.133 .....	39
Figure 30. Connection Established with 192.220.233.50 .....	40
Figure 31. Failed Connections for Opening an Image.....	40
Figure 32. Images After Blocking and Unblocking the Image Server.....	41
Figure 33. Conversation with Blocked IP Addresses.....	41
Figure 34. Infrastructure of Chat Server .....	42
Figure 35. DNS Query for Twitter.com.....	44
Figure 36. TLS Handshake .....	45
Figure 37. TLS Handshake- Client Hello Packet.....	45
Figure 38. Certificate, Server Key Exchange and Server Hello Done Packet .....	46
Figure 39. Client Key Exchange Packet .....	46
Figure 40. Packet Containing New Session Ticket.....	47
Figure 41. Encrypted Data Patterns After Successful Authentication .....	48
Figure 42. Patterns for Entering Wrong Password .....	49
Figure 43. Connection Redirection to Alternate IPs after Entering Wrong Password.....	49

Figure 44. Traffic Patterns for Logout Activity .....	49
Figure 45. Flow Graph for Logout Activity.....	50
Figure 46. DNS Queries for Media Server .....	50
Figure 47. Query Sent to api-stream.twitter.com.....	50
Figure 49. Fixed Traffic Patterns While Image is Opened .....	51
Figure 48. Connection Established with Twitter to Open an Image .....	51
Figure 50. Fixed patterns on TCP port 55807 (Observation-I).....	52
Figure 51. Fixed patterns on TCP port 55807 (observation-II).....	52
Figure 52. Fixed patterns on TCP port 49868 .....	52
Figure 53. Traffic Flow Graph for Attaching and Sending a Media in Messages .....	54
Figure 54. Connection Established with upload.twitter.com .....	54
Figure 55. Patterns While Uploading a Photo in Messages .....	55
Figure 56. Patterns While Sending a Photo in Messages.....	55
Figure 57. Fixed Patterns for Playing a Video.....	56
Figure 59. Observation of Timeframe for Stream of Packets .....	58
Figure 58. Timestamps Observation for Connection Establishment .....	58

## **Abstract**

With the rapid technological advancement, the usage of social media applications has increased resulting in major security concerns. From leisure activities to business management, a social media application is frequently used, attracting the attention of criminals to perform undesirable activities. During such activities certain remnants often reside within the network. However, the increased security in the application development and HTTPS based client server architecture makes the network investigation complex. The extensive analysis of sessions of encrypted traffic has the potential of identifying and classifying important artefacts related to users, and their activities. This has attracted researcher's attention in studying secure social media applications in forensics and information security domains.

In this thesis, we carry out network forensic analysis of Twitter, a famous social media application which applies encryption to protect information over the network. The analysis is conducted by extracting hidden patterns of the application, information of involved parties and related activities. Focusing on byte level analysis, we identify fixed patterns against user activities from client server sessions. This analysis in forensic research is termed as behaviour analysis of secure applications.

The objective of this research is to forensically profile the Twitter application through network traffic analysis. Our methodology is based on understanding of traffic classification and behaviour analysis techniques, traffic interception and identification of fixed patterns to correctly identify the user activities. Moreover, firewall is used to explore the hidden design flexibilities and other connectivity options used by Twitter.

Our analysis shows that we can correctly identify the flow of Twitter traffic, user related information, and fixed patterns to classify the user activities on Twitter. Our methodology of Network Traffic Analysis of Twitter can be of great help during criminal investigations.

### **Keywords:**

Android, Behaviour analysis, Digital Forensics, Encryption, Firewall, HTTPS, Network forensics, Network monitoring, pfsense, Port based analysis, Secure social media, Traffic analysis, Twitter, Wireshark

# Chapter 1

## INTRODUCTION

### 1.1. Introduction

This chapter discusses the initial concepts and historical background to build understanding and roadmap of the entire work. It further defines the basic concepts of digital forensics, forensic categories, motivation, and extent of this work. The structure of this chapter is as follows:

- **Section 1.2** Historical Background
- **Section 1.3** Motivation
- **Section 1.4** Problem Statement
- **Section 1.5** Research Questions
- **Section 1.6** Thesis Outline
- **Section 1.7** Summary

### 1.2 Historical Background

The frequently used terms in this work are “Forensics”, “Digital Forensics” and “Network Forensic Analysis”. This section describes them in detail so that readers with little or no knowledge can develop understanding of this work.

#### 1.2.1 Digital Forensics

Forensics is a Latin word which refers to the “Court of Law”. The concept of computer forensics was raised in response to the seriousness of digital crimes. It is defined as applying computer science as an art and science in a legal process [1]. The evidence acquired, stored and conveyed through a digital source in legal proceedings and criminal investigation is known as a Digital Forensics [2].

Important terms and methodologies in digital forensics are Identification, Analysis, Interpretation, Preservation, Validation, Documentation and Presentation of the information known as digital evidence [3]. Integrity and validity of a digital evidence needs to be ensured during the entire process of an investigation. Investigation of digital evidence is reported as a post-event response in a forensics process [4]. Mostly an acquired digital evidence is in raw format which makes an investigation complex, therefore, many digital forensics tools are available which translate the data into human readable format and make investigation easier.

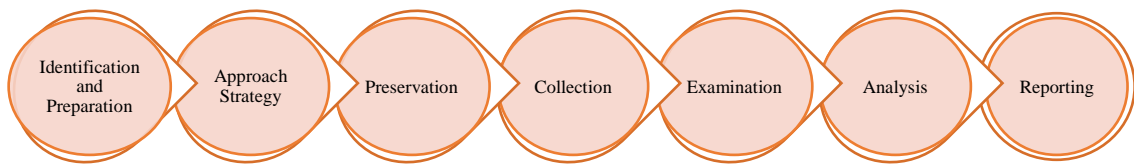
Further, to ensure the reliability of digital forensics the following requirements must be fulfilled:

- i. Presented scientific theory is proved beforehand
- ii. The error rate of applied method is defined
- iii. The entire process must be reviewed by the committee beforehand

iv. The approval of process by the committee

European Network of Forensics Science Institutes (ENFSI), Forensic Information Technology (FIT) and various other working groups are formed to extend the above mentioned criteria into digital forensics [5]. To extend this, in [6] two more factors are defined i.e. the rights of the involved people must be preserved and ensured the integrity of digital evidence.

In the above context, implementation of digital forensics in crime investigations will help in prevention of malicious attacks and in case of occurrence of such event, it will help in finding the guilty, providing justice and preventing such events. To start with the investigation, a standard methodology is studied which ensures the proper identification and preparation of the event, approach strategy, preservation of the evidence, collection of evidence, examination, analysis and reporting. Digital forensics method is defined in Figure 1 below.

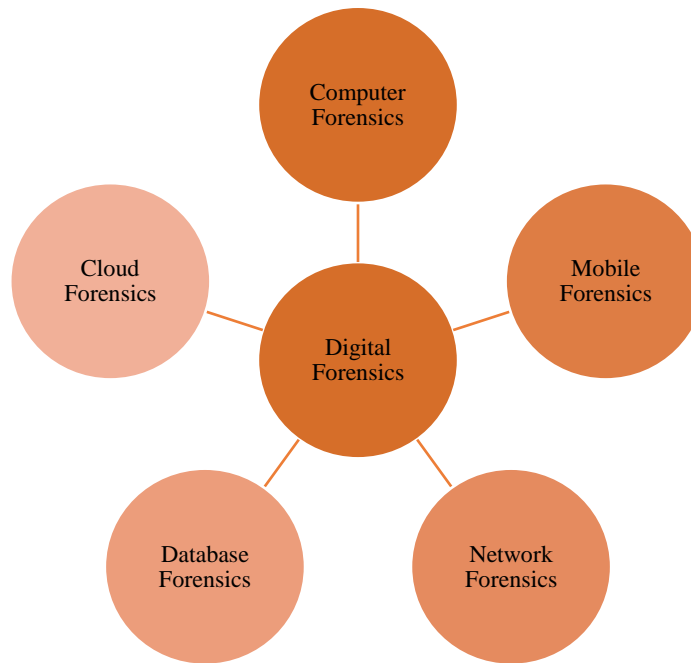


*Figure 1. Digital Forensics Model*

### **1.2.2. Categories of Digital Forensics**

Digital Forensic Investigation (DFI) is used in court of law for legal proceedings with the steps discussed in the previous section. However, digital forensics is not limited to acquiring data from computer devices. Criminal activities are carried on different mobile devices and tablets. Networks are another source of launching an attack. Therefore, digital forensics has its scope in multiple branches to perform forensic investigation. In this section we will discuss the branches of digital forensics. Digital forensics is divided into five main categories including Mobile Forensics, Network Forensics, Database Forensics, Cloud Forensics and Computer Forensics [7]. Categories are described briefly and shown in Figure 2 below.

- **Computer Forensics:** Computer forensics refers to acquiring, extracting and examining an evidence retrieved from computer sources such as hard drives, RAM, Flash Drives, Registries, logs and internet sources. For computer forensics, there are many tools used to extract the evidence such as Encase, Helix and FTK Imager. These tools have the ability to work with FAT and NTFS file systems. Wiping the storage area and taking the MD5 image of an evidence before starting the forensic investigation are necessary steps in computer forensics.



*Figure 2. Categories of Digital Forensics*

- **Mobile Forensics:** Mobile Forensics focuses on investigation of mobile devices to acquire evidence. The sources of evidence are SMS, Gallery, phone directories, call histories and mobile memory. Along with this, cameras, communication systems, multimedia and Global Positioning System (GPS) makes mobiles more vulnerable to crimes [8]. Due to the advancement in computing, mobile devices are smaller in size and carry enough data which makes them a prime source of information in criminal proceedings [9].
- **Cloud Forensics:** In cloud forensics, major source of information is cloud environment and cloud storage. It is a subset of Network Forensics and an application of digital forensics in cloud computing [10]. Cloud forensics follows the steps of network forensics and cloud computing techniques.
- **Database Forensics:** Database forensics refers to the investigation of content of a database, its metadata and cached information in RAM. Database forensics and database recovery are not the same concepts. Database forensics refers to the reconstruction of metadata by investigating the failed database.
- **Network Forensics:** Network forensics is another branch of computer forensics which deals with the investigation of network traffic. It includes capturing traffic for malicious activities, its analysis and reporting for further actions. Traffic monitoring has two main perspectives, first, monitoring the captured traffic or real time traffic for forensics investigation and second, traffic monitoring for intrusion detection and prevention purposes. For forensic investigation, packet level analysis is used in order to analyse the traffic and find possible artefacts against a criminal activity. This research focuses on

network forensics of a secure social media application. In the next section, we will discuss network forensics and its challenges in detail.

### **1.2.3 Network Forensics and its Challenges:**

Network Forensics is one of the branches of digital forensics which deals with monitoring and extracting evidence retrieved from network traffic. In order to understand the network traffic; it is necessary to analyse the network traffic flow [11]. Network analysis can be performed in two ways. One is the real time monitoring which requires intense human involvement and hardware requirement second is capturing traffic and analysing it offline. The later method involves is the reconstruction of traffic which is mainly called “Network Forensics”.

A forensic network is built to monitor the traffic for vulnerabilities, failed configuration, malicious activities and other security violations. Network Traffic Analysis is not only performed for vulnerabilities but for retrieving of information. For example, if an organization’s backup server is failed, the backed up data can be retrieved from recorded network traffic. Further, application level faults and traffic rate can also be monitored during network traffic analysis. Network monitoring tools play an important part in the analysis of traffic. Such tools are called Network Forensic Analysis Tools (NFATs). NFATs must be capable of performing three tasks, capture traffic, analyse traffic and provide useful information from the analysis. Before an investigation, it must be ensured that the tool is able of capturing and processing high volume of traffic. Further, in order to analyse the traffic, network forensic investigators must have some understanding of the tools.

Many open source tools are used for network forensics where the majority of them lack basic functionalities and therefore are replaced by proprietary commercial tools. These tools are time and cost efficient and require less human intervention [12]. In hardware and operating systems, FreeBSD, OpenBSD and Linux are more reliable and suitable for network forensics. There are six types of network forensics methods: Distributed System mechanism, Soft Computing method, Honeypot Technique, Attack Graph, Formal and Aggregation method [13].

Data retrieved from the network is referred to as network based evidence. Network based evidence can be in different forms including Full content data, Session data, Alert data and Statistical data [14]. The types of network based evidence are defined below:

- i. **Full content data:** It is the recording and storing of entire network data and captured files without filters. Wireshark is the most common tool used to capture this data.



- ii. **Session data:** Session data usually includes the flow of conversation between two or more parties. This aggregated evidence helps investigators in analysing who communicated with whom and how many entities in a particular period of time.
- iii. **Alert data:** In some analysis, the analyst pre-defines some characteristics on the network such as IP addresses for a particular server. In the occurrence of those characteristics, an alert is generated which is known as alert data. For this type of investigation, Network Investigation and Detection System (NIDS) is used. This type of evidence is more prone to false alerts which need to be dealt with extensive care.
- iv. **Statistical data:** Last one is statistical data, consisting of terms such as number of bytes in a packet, IP addresses, Timestamps and so on. This type of network based evidence is captured by Wireshark usually.

Several network forensic methodologies are proposed in different places. In [15], a generic framework is proposed consisting of eight steps including Preparation and Authorization, Detection, Collection of Network Traces, Preservation and Protection, Examination, Analysis, Investigation and Presentation. In this study, we have defined the suitable Network Forensics Investigative Methodology from the book “Network Forensics: Tracking Hackers through Cyberspace” by Shaerri Davidoff and Jonathan Ham [16]. This methodology is called OSCAR which stands for Obtain Information, Strategize, Collect evidence, Analyse and Report. Figure 3 shows the OSCAR methodology.

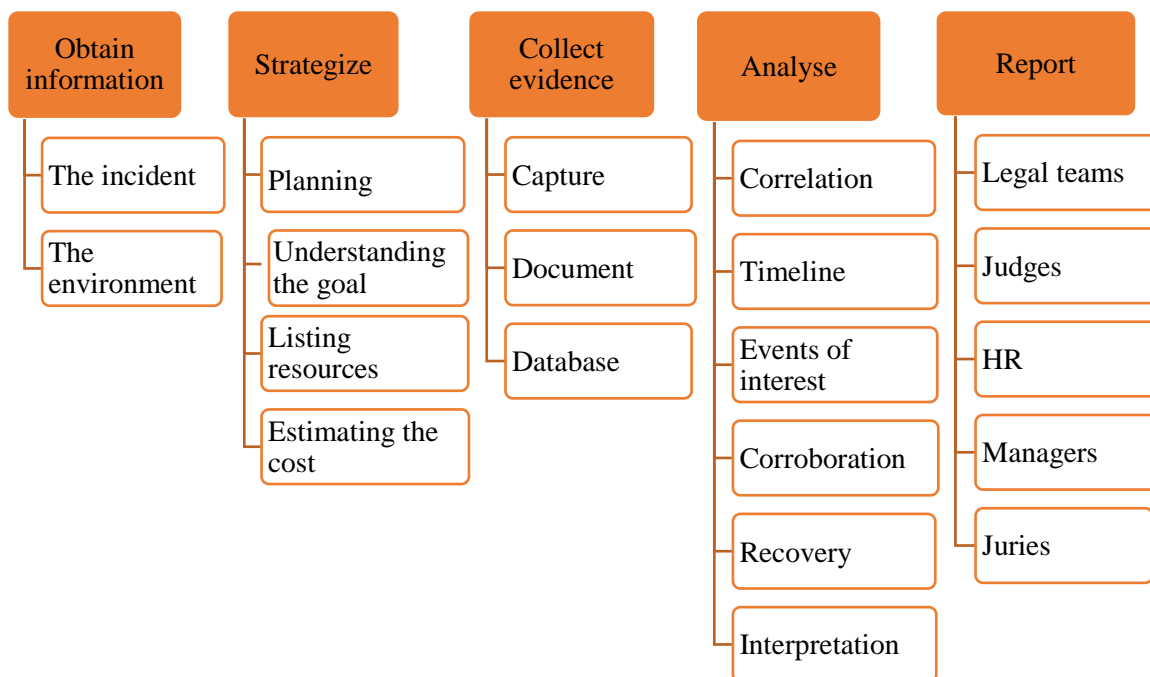


Figure 3. The OSCAR Methodology

- **Challenges in Network Forensics**

Network Forensics needs a significant amount of effort with accurate results. The challenges faced by network forensic investigators are described in this section. While carrying out the network forensics, acquisition is the major task in the collection and detection phase. Further, a huge information resides in different places, such as firewall logs, database servers and wireless access points. Aggregating, locating and gaining legal access to the specific and targeted location is much more difficult in the initial phase.

The randomization in traffic makes it difficult for an investigator to characterize, differentiate and correlate the malicious and non-malicious traffic. Storage and privacy are also major concerns where the network related evidence requires huge storage and protection from different attacks. Finally, during an investigation process, meeting the criteria in the court of law, creating a long chain of evidence and locating a criminal geographically demands extra effort. Some of the major challenges are listed below:

- Acquisition
- Aggregating and locating an evidence
- Gaining legal access
- Characterization, differentiation and correlation of an evidence
- Storage
- Privacy
- Meeting the criteria in court of law
- Locating criminal geographically

#### **1.2.4 Overview of Twitter application**

Twitter is an American social networking and microblogging online service which has 25+ offices worldwide [17]. Users can interact with each other by sending messages, updating, liking and retweeting the tweets, following, unfollowing and sharing media. On this platform, the registered users can follow people with similar interests [18]. It is mostly used by academic students, politicians, musicians and the general public for promoting work and news updates.

There are around 330 million monthly active users on Twitter worldwide [19] [20] and 145 million daily active users. 500 million tweets are created every day [21] and around 23% of internet users use Twitter. According to a survey 120 million users access Twitter through desktop and a mobile application out of which 31.1% use it on Android [22] [23]. Statistics show that it is a major platform of carrying out criminal activities. Twitter Inc. uses Google services, Amazon Web Services and MCI communication services for load balancing on highly distributed infrastructure.

### **1.3 Motivation**

Online communication through social networking has given cyber criminals an open platform for committing criminal activities. The increased use of social media applications on smartphones, has also increased the rate of cyber incidents. However, forensic investigation of such applications has become difficult due to the intense security in the design and development of features and architecture of such applications. Therefore, there needs to be a mechanism which provides a complete process of the analysis.

Twitter is a secure social media application which is used worldwide by millions of people. The crime rate has increased by 780% in 2008 and around 650 people were charged for the crime [24]. In 2016, a user leaked the sensitive data of 41 million Twitter accounts including email addresses and credentials [25]. In April 2020, Curiel et. al correlated the frequency of tweets with actual murder cases over a specific region and a period of time. The correlation resulted in high accuracy showing the highest crime rate on Twitter application. The study further states that 15 out of 1000 tweets are related to hate and abusive crimes [26]. In February 2020, Facebook's Twitter account was hacked where the hackers tweeted about the flaws in Twitter's security [69]. In July 2020, attackers hacked Twitter accounts of different celebrities to promote a scam [70]. Another report states that, some attackers stole the credentials of Twitter employees and accessed the internal support using those credentials in July 2020. Through internal support system, the attackers also accessed 130 different Twitter accounts and Tweeted from 45 accounts, accessed the inbox of 36 accounts and downloaded the Twitter data from 7 accounts [71]. With this rise in Twitter-based crimes, it becomes necessary to carry out the network traffic analysis of Twitter application. However, the distributed architecture and encryption of information over the network makes this task challenging. Nevertheless, an extensive behaviour analysis technique can be used to retrieve the important artefacts which can help a forensic investigator. Thus, the motivation behind this study is that in spite of high usage and crime rate there is no work done on the encrypted traffic analysis of Twitter in detail.

### **1.4 Problem Statement**

Twitter provides additional security features with confidentiality and HTTPS based client server architecture. The contents of the application are encrypted thus hiding the information over the network. Moreover, the distributed architecture has added the complexity feature for forensic investigators during criminal proceedings.

The objective of this study is to perform an extensive Twitter traffic analysis to classify the encrypted traffic and identify patterns that can assist investigators. Further, the hidden design flexibilities and other connectivity options are discovered using a firewall.

## 1.5 Research Questions

This research focuses on extensive profiling of secure social media applications. For the interest of forensics examiners, researchers and readers we will answer the following questions in this study:

- How will the traffic be identified for a particular application on the network?
- For the interest of a forensics investigator, what kind of remnants will be retrieved from the encrypted network traffic?
- How will the encrypted traffic be classified for different user actions?
- How will the user actions be identified after classification of traffic?
- From the encrypted traffic, can the contents be decrypted?
- Will a network forensic investigator be able to learn the distributed network client server architecture of Twitter application and to what extent?

## 1.6 Thesis Outline

For better understanding and simplicity, the thesis is organized into seven chapters. In this chapter, the concepts, previous work and related terms are described in details. Continuing from Introduction chapter 2 is the “*Literature Review*”. In chapter 2, we have discussed the previous work done by researchers worldwide and have chained the entire work with this study.

Chapter 3 describes the “*Research Methodology*” which consists of the process and steps we have adopted to implement our research work. Further, in this section we have described complete detail of the proposed framework.

Chapter 4 discusses the “*Experimental Setup*” established to carry out the research. In this section, complete details of tools and technologies, workstations and devices along with configurations are defined.

Chapter 5 consists of “*Analysis of Application and Results*” which defines the complete forensic analysis of application fulfilling the research objectives and answering research questions.

Chapter 6 defines the “*Conclusion and Future Scope*” where the entire work is concluded and the future directions are discussed.

Chapter 7 contains the “*References*” in which the bibliographic source of the entire work is specified in IEEE format.

## 1.7 Summary

In this chapter, we have discussed the background, complete process and categories of digital forensics in detail. The discussion was further continued for Network forensics and its challenges. After that, the motivation and problem statement were described. In the end, the thesis outline was defined.

# Chapter 2

## LITERATURE REVIEW

### 2.1 Introduction

This chapter contains the latest theoretical and experimental work related to this thesis. The literature proves to be helpful throughout this research. Further sections are as follows:

- **Section 2.2** Discussion on previous methodologies of Network Forensics
- **Section 2.3** Work done on Network Forensics of social media applications
- **Section 2.4** Literature review on Network Analysis and security tools
- **Section 2.5** Summary

### 2.2 Methodologies of Network Forensics

In [27] M.A.K Sudozai and S. Saleem proposed a generic framework for network profiling of a secure chat and calling application. It states that during criminal investigations, if the architecture and communication protocol of an application is unknown and the contents are encrypted, behaviour analysis technique can be helpful. Traffic classification is based on port and behaviour analysis techniques. For port analysis, flexibilities in TCP and UDP ports are used for chat and calling activities. Behaviour analysis is carried out by repeating the activities and identifying the byte patterns, their frequency, payload size, request and acknowledgment between client and server. In this methodology, a firewall is used to find possible alternate connectivity options of secure applications. This generic framework was applied on Android and iOS platforms.

Ndatinya et al. [28] proposed a methodology for identification of attacks launched on a network using packet analysis technique. According to this study, Network analysis can be proactive and reactive. In the proactive approach, the network is pre-analysed and in reactive approach the network is diagnosed after an intrusion activity. Further, Wireshark is used for packet analysis. The analysed attacks include covert network channel, drive-by downloads, ICMP based attacks and DDOS attacks using Wireshark filters and port scanning.

Petr et al. [29] surveyed the traffic classification and analysis methods for encrypted traffic. The study proves that for protocol structure and behaviour analysis it is important to initially understand the protocol and then the information provided by the protocol. The information provided by the protocol is studied from two phases including unencrypted initialisation phase and encrypted data transport phase. The studied protocols are IPsec, TLS, SSH, BitTorrent and Skype protocol. Moreover, for traffic analysis, payload based and feature based classification techniques are proposed. For payload based technique, classification

tools are used. Feature classification technique is based on supervised machine learning methods, semi-supervised machine learning methods, basic statistical methods and hybrid methods. The above methodologies are summarized in Table 1 below:

*Table 1. Network Forensics Methodologies*

Techniques		References		
		Sudozai etl al	Ndatinya et. al	Petr et. al
Packet Based Analysis	Byte Level Analysis	✓	✓	✓
	Payload Based Analysis	✓	✓	✓
Port Based Analysis		✓	✓	×
Feature Based Analysis	Supervised Machine Learning Methods	×	×	✓
	Semi-Supervised Machine Learning Methods	×	×	✓
	Basic Statistical Methods	×	×	✓
	Hybrid Methods	×	×	✓

### 2.3 Network forensics analysis of social media applications

In 2018 Sudozai et al. [30] proposed a network forensics study of IMO application on Android and iOS platforms. Port based analysis and fixed pattern analysis techniques were used to observe the user related activities. Moreover, to observe the design flexibilities of this application, a firewall was used. The results showed that using these techniques some important artefacts can be found from encrypted traffic which can be helpful during criminal investigations.

In 2017, another forensic analysis technique was proposed by Muehlstein et al. [31] to identify the operating system, web browser and application traffic over the network. Supervised machine learning approach was used to analyse the encrypted traffic. Results showed how the confidentiality of users can be exploited by the intruders from the secure network. Moreover, the baseline features (TTL value, incoming and outgoing packets sizes etc.) and new features (window size, SSL cipher and compression methods etc.) are used for traffic behaviour identification. Similarly, in [32] a data mining approach is proposed to determine the vulnerability residing within the Viber communication patterns. According to this research, Viber voice and IM application has a dynamic and distributed server architecture and has an unknown

protocol structure. However, the identification of the server is done on the probability which can result in inaccurate server identification. Further, there is no proper statistical evaluation of user activities.

In 2015 Daniel et al. [33] performed network and device forensics analysis of 20 social media applications. This research focused on finding the artefacts such as messages exchanged, videos, images, passwords and voice notes from unencrypted traffic. It also highlights the confidentiality and privacy concerns of users due to unencrypted network traffic. Studied applications are Tinder, Wickr, Snapchat, BBM, WhatsApp, Viber, Instagram, Okcupid, ooVoo, Tango, Kik, Nimbuzz, MeetMe, MessageMe, TextMe, Grindr, HeyWire, Hike, textPlus and Facebook Messenger. However, out of 20 applications, network forensic analysis was performed only for those applications which had unencrypted traffic over the network.

In 2015, Karpisek et al. [34] proposed network forensics of WhatsApp providing complete understanding of WhatsApp signalling messages. This study gives detailed review of the tools and methodologies to capture and decrypt the traffic. Using these methodologies, some artefacts such as passwords, phone numbers, duration of calls and timestamps were obtained. Wireshark, password extractor and WhatsApp dissector were used to capture and decrypt the credentials and protocol messages. However, the new version of WhatsApp has introduced end-to-end encryption. Therefore, this methodology is only applicable to older versions of WhatsApp.

Conti et al. [35] performed the traffic analysis of Gmail, Facebook and Twitter on Android OS. Using a supervised machine learning approach, user actions were identified from the encrypted traffic. The traffic flow was exhausted and a labelled dataset was created for different user actions using domain filtering and packet filtering. Further, hierarchical clustering method was used to detect the clusters with different sets of flows. The study highlights that the privacy of users can be violated even from encrypted traffic. However, the attributes used to analyse the flow of encrypted traffic is limited to identify and differentiate the user actions.

Yusoff et al. [36] proposed network forensic study of Facebook, Twitter and Telegram on Firefox. Virtual cloud environment was used for traffic capturing and analysis. The user actions analysed were installation of apps, message activates, tweet update, retweet and so on. For capturing traffic, Wireshark was used. Network Miner and Microsoft Network Monitor were used to analyse the traffic. Some important artefacts such as passwords were captured on Telegram application as it processes unencrypted traffic. For Facebook and Twitter, originating IP address, originating countries and certificate providers were found. However, along with the high memory requirement, the criteria to analyse the traffic such as IP addresses

and identification of certificate providers were limited. Further, this work lacks the synchronization between the performed user activities and traffic behaviour analysis against those activities.

Molnár et al. [37] attempted to identify and analyze the Skype traffic using different algorithms. These algorithms are based on dynamic flow and characteristics of packets. In this study, the skype hosts and connection initiation were identified by detecting the port, IP addresses and timestamps. The found parameters in the first step were exploited to analyse the flow of calls on skype network. This study is only focused on the identification of skype traffic and does not define a generic method to identify the user activities. Table 2 defines the limitations of the discussed related work.

*Table 2. Related work Limitations*

<b>Description</b>	<b>Limitations</b>
Daniel et. al	<ul style="list-style-type: none"> <li>• Unencrypted traffic analysis</li> <li>• No network analysis for Snapchat, Tinder, Wickr and BBM</li> </ul>
Conti et. al	<ul style="list-style-type: none"> <li>• Attributes to analyse the flow of encrypted network traffic is limited to identify and differentiate the user actions</li> </ul>
Marik et. al	<ul style="list-style-type: none"> <li>• The identification of server is done on probability, therefore proper selection of server is needed</li> <li>• No proper statistical evaluation for user activities</li> <li>• The accuracy evaluation is still needed</li> </ul>
Yusoff et. al	<ul style="list-style-type: none"> <li>• Criteria to analyse the encrypted network traffic is limited</li> <li>• The traffic is not categorised according to the specific user activities</li> <li>• High memory requirement</li> </ul>
Muehlstein et. al	<ul style="list-style-type: none"> <li>• Due to the randomness in application’s traffic, supervised machine learning technique will not result in maximum accuracy until complete and detailed samples are given with maximum features</li> </ul>
Molnár et. al	<ul style="list-style-type: none"> <li>• Algorithm focuses only on the identification of the Skype traffic</li> <li>• No generic method to analyse the user activities</li> </ul>

Installation and usage of any social media application on Android devices leaves behind a wealth of data even after uninstallation. Android devices play an important part in network forensics due to synchronization features of online applications with the data saved in Android devices. With the advancement of technology, features in social media applications are upgraded and are used for almost every purpose, increasing the crime rate.



Wu et al. [38] and Mehrotra et al. [39] proposed a forensics study of WeChat and Wickr applications on Android devices. Anglano [40] presented the forensics analysis of WhatsApp messenger in 2014, then in 2016 and 2017 Anglano et al. presented the forensics study of ChatSecure IM application and Telegram app on Android platform [41] [42]. Husain et. al [43] presented the device forensic analysis of Google Talk, AIM and Yahoo!, Messenger on iOS and Windows. In [44], Mutawa et al. performed the detailed forensic investigation of Facebook, Twitter and MySpace on Android, BlackBerry and iPhone smartphones.

## 2.4 Network forensics analysis and security tools

According to [45], on 6<sup>th</sup> August 2009 three social media apps including Twitter, Facebook and Google were blocked by distributed denial of service attacks where Facebook and Google were able to recover but Twitter Inc. was not able to recover from the attack. The reason behind the Twitter failure was the lack of proper tools to detect and counter the attacks. Network forensic analysis tools play an important part in monitoring networks, capturing traffic, analysing it and creating the incident response. The tools are classified into two categories: Network Security Monitoring (NSM) tools and Network Forensic Analysis Tools (NFAT). The classification of tools is described in Table 3 below.

Table 3. Traffic Classification Tools

Network Security Monitoring (NSM) tools					
Packet capture	Statistic	Pattern matching	Manipulation	Fingerprinting	IDS
✓ TCPDump	✓ TCPTrace	✓ Ngrep	✓ SiLK	✓ P0f	✓ Bro
✓ Wireshark	✓ Ntop	✓ TCPXtract	✓ TCPReplay	✓ Nmap	✓ Snort
✓ TCPFlow	✓ TCPStat				
✓ Flow-tools	✓ NetFlow				
✓ NFDumps	✓ TCPDstat				
✓ PADS					
✓ Argus Nessus					
✓ Sebek					

NSM tools do not provide preventive measures. However, there are certain NSM tools which provide alert notification in case of any abnormal activity. In the proposed research, we have used Wireshark which only monitors the traffic but does not notify the mishap over the network [46]. Further, Wireshark has a colouring scheme which shows different colours for different kinds of packets such as RESET, Retransmission and Duplicate packets.

## **2.5 Summary**

In this chapter, we discussed the related work done in the field of network forensics. There are several network forensic methods proposed by digital forensic researchers. In Section 3, network forensic analysis of applications is described and in Section 4, network monitoring tools are defined.

# Chapter 3

## RESEARCH METHODOLOGY

### 3.1 Introduction

Research is a method of following particular steps and tools to achieve the targeted objectives. In this chapter, the process of this research and proposed framework are discussed. The subsections are as under:

- **Section 3.2** The Research Method
- **Section 3.3** Proposed Framework
- **Section 3.4** Summary

### 3.2 The Research Method

Research is the process of collection and analysis of information to get a complete understanding of any subject [47]. It has three major steps including problem identification, following the specific steps for information collection and presentation. Various research methods can be adopted according to the requirement of the study. According to [48], the objective of the research is to find the hidden answers through scientific procedures and practices which gives new insights of the problem.

According to [49], the research is not only the information collection but providing non-existent answers to various questions. In John W. Creswell's book "Educational Research", a complete research process is defined which is considered in this study with required modifications [47]. The research methodology is defined in Figure 4 and defined below.

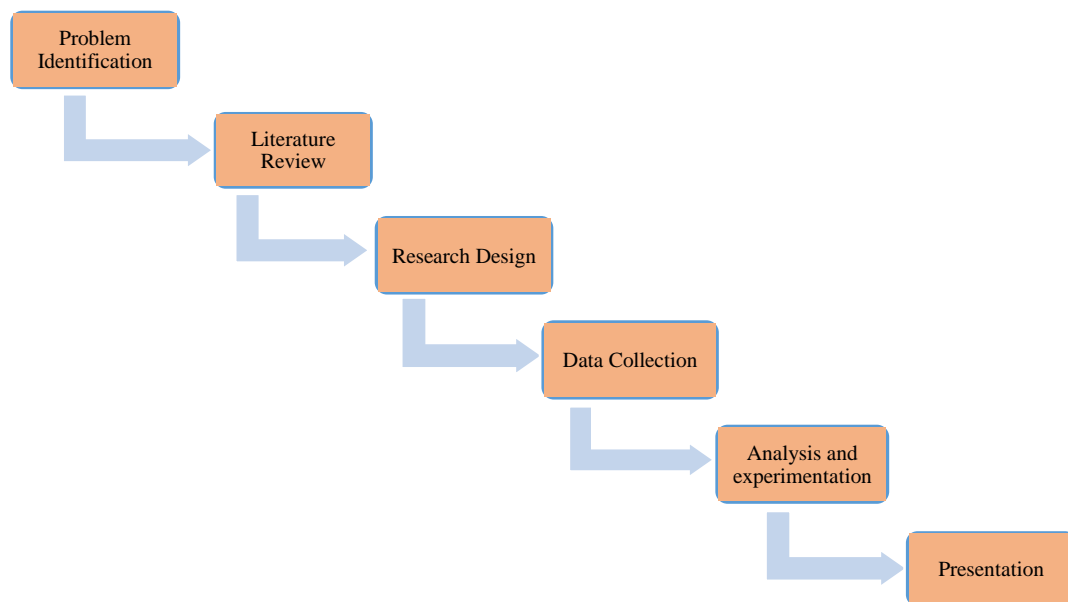
- **Problem Identification**

To start the research, a particular problem needs to be focused. Therefore, the first step is to identify the field of study, finding loopholes and then perform further research. Problem identification exhausts the study to a specific issue which leads to the simplicity of research. This study is mainly focused on network forensics analysis of a secure social media application where the challenge is to retrieve the artefacts from encrypted traffic. This step helped us to concentrate on a particular field and carry out exhaustive research in the field of network forensics.

- **Literature Review**

After the identification of a problem, its background work has to be reviewed in detail. Reviewing the background literature is an important step because it enhances the knowledge, techniques, tools and methodologies which removes the gap between problem and the solution. In this study, we first reviewed the previous work, categorised it and mapped it to our study.

In the first step of literature review, we studied all the methodologies proposed by the researchers from national and international conference papers, journals, articles and blogs. After that, the related tools and technologies were reviewed and the previous work on social media applications was studied. In the end, the entire work was summarized in order to know where the current research lies and how much gap has to be filled to carry the research further.



*Figure 4. The Research Process*

- **Research Design**

Research design contains the entire study including formulating the research questions, defining data collection methods, answering the research questions and showing the results. It is an important step where the study types, time division and proposed framework are described. This research is qualitative in nature which is focused on creating deep understanding of a problem. On the other hand, quantitative research is based on numerical and statistical data [48]. Based on the problem identification, our research is divided into six questions described in Section 1.5. We have adopted the case study approach which is based on the identification and observation of the case and then reconstructing the case. We worked on network forensics analysis of Twitter applications and performed several activities to observe the behaviour. The case study method is described in Figure 5 below.

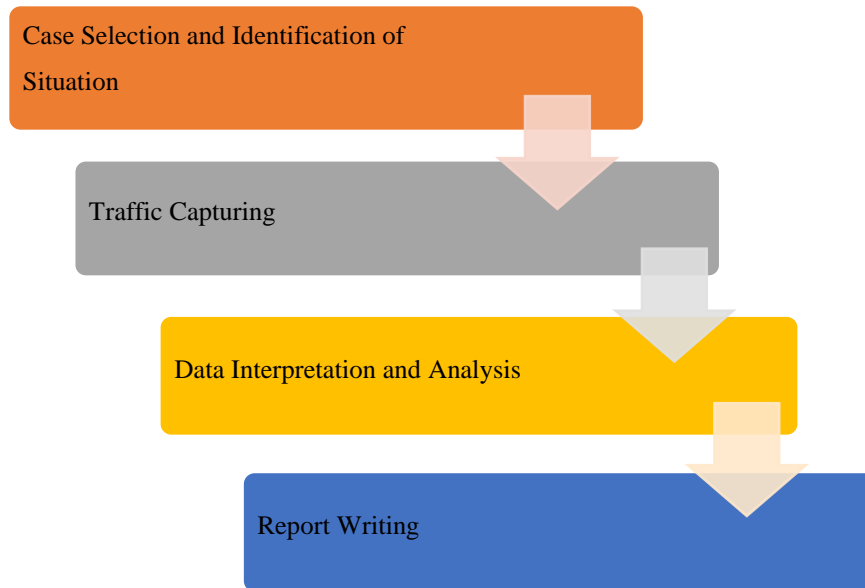


Figure 5. Case Study Method

- **Data Collection**

To answer the research questions, data collection is a necessary step. It is a significant part, because all the technical methods and experimentation is based on collected data. For data collection phase, we have adopted the National Institute of Standards and Technology (NIST) guide to Integrating Forensics Techniques into Incident Response [50]. We followed the NIST framework for the data collection phase consisting of collection, examination, analysis and reporting steps. Figure 6 shows all the steps in the data collection phase.

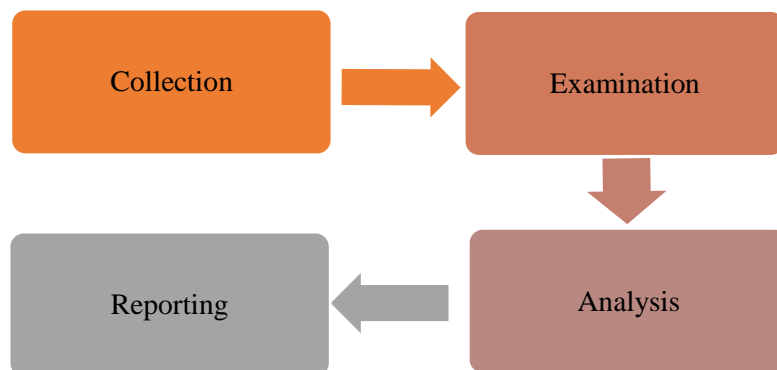


Figure 6. NIST guide for Digital Forensics Process

We integrated the case study method in the NIST guide and created a method for further research. The first two steps of the NIST process were replaced with the first two steps of the case study method including Case Selection & Identification of Situation and Data Capturing. The remaining two steps have similar

purpose in both frameworks. All the steps of the final research process along with the case studies are defined below.

- Case Selection and Identification of Situation:

In this step, we define the particular case and situation as required in this study.

1. Application Installation:

The latest available version of Twitter application was downloaded and installed from Google play store on Android. This application was used for the research in this study. Description of this case is defined in Table 4 below.

*Table 4. Case-I Description*

<b>Application Installation</b>	
<b>Task</b>	Twitter application was installed on Android
<b>Description</b>	The available version of Twitter application was searched on Google play store on Samsung Galaxy J7 and downloaded it.
<b>Purpose</b>	The purpose of installation of this app was to carry out further research by performing different activities and analysing the traffic behaviour on the network.

2. Application Login and Logout:

After downloading and installing the application the traffic was analysed for Login and Logout activity. The description is given in Table 5.

*Table 5. Case-II Description*

<b>Application Login and Logout</b>	
<b>Task</b>	Logged-into the application and Logged-out of the application.
<b>Description</b>	The user logged-in and Logged-out of the application
<b>Purpose</b>	The application was logged-in to perform further activities to analyse the traffic behaviour on the network. After performing activities, the user Logged-out of the application.

3. User activities:

For the analysis of user actions, some specific user activities were performed. Description is given in Table 6.

Table 6. Case-III Description

<b>User activities</b>	
<b>Task</b>	Performing different activities on application
<b>Description</b>	This case involves the user actions including Message activities Video playing Image opening
<b>Purpose</b>	The purpose of this case was to analyse the traffic behaviour of the user actions by packet analysis, port analysis and identification of source and destination IP addresses.

○ Traffic Capturing:

This phase consists of traffic capturing against user actions, traffic identification, traffic filtering and verifying the obtained results. We performed different activities on Twitter app e.g., the application login while the traffic was captured. After capturing traffic, the packet capture file was saved with the activity name. From the file, we identified the source and destination IP addresses and filtered the traffic for those IP addresses for further analysis.

● **Analysis and Experimentation**

This step is constant for all the discussed methodologies. The captured file was examined and analysed in detail to find important artefacts. The analysis process followed for the captured traffic is defined as under:

- Identify source and destination IP addresses
- Identify source and destination ports
- Identify packet size
- Timestamp analysis
- Identify payload size
- Identify fixed patterns and conclude the behaviour from that pattern and analyse the results.

● **Presentation**

This is the final step for all the research methods defined above. In this step, the results were concluded. The finalised results are compiled into a report and a presentation file is created. The report is written in a

way that can be easily understood by the fellow researchers and readers. The summary of the research is then compiled into a presentation which is the end of the research study.

### **3.3 The proposed Framework**

A validated research is one which answers the research questions at the end [51]. A generic framework concludes the results and is the best possible way to answer the research questions. To answer the questions, an analysis framework is proposed and is described in Figure 7 below.

The framework shows complete analysis of this research study and the process is described. The process starts with the identification of IP address range and port range. After the observation of IP address ranges, the steps for behaviour analysis are shown against different activities on Twitter application. After the environmental and technological setup, an application is accessed and an activity is performed. While performing an activity we enabled the traffic capturing tool and captured the traffic for the activity.

After accessing the application, we captured the traffic and observed application server IP addresses and ports. In the next step, we blocked the newly observed IP address and repeated the process. The process for IP range observation was repeated until the app server stopped responding. In the next step, the observed server IP addresses were concluded.

After concluding the entire server ranges, we performed behaviour analysis of traffic. For behaviour analysis, we performed several activities and captured traffic for each. The process for behaviour analysis is non-iterative, the performance of one activity does not depend on another activity. Observed behaviour includes IP address and port identification, packet length, payload size, pattern analysis and flow graph analysis. After the behaviour analysis and IP address identification, the process ends.

Further, we performed an experiment for a month to analyse the behaviour of traffic against different user actions and to observe the server IP address ranges. We repeated the experiment after two months. We observed that server IP addresses were similar in both experiments which proves that the observed IP addresses are reserved to provide particular services.



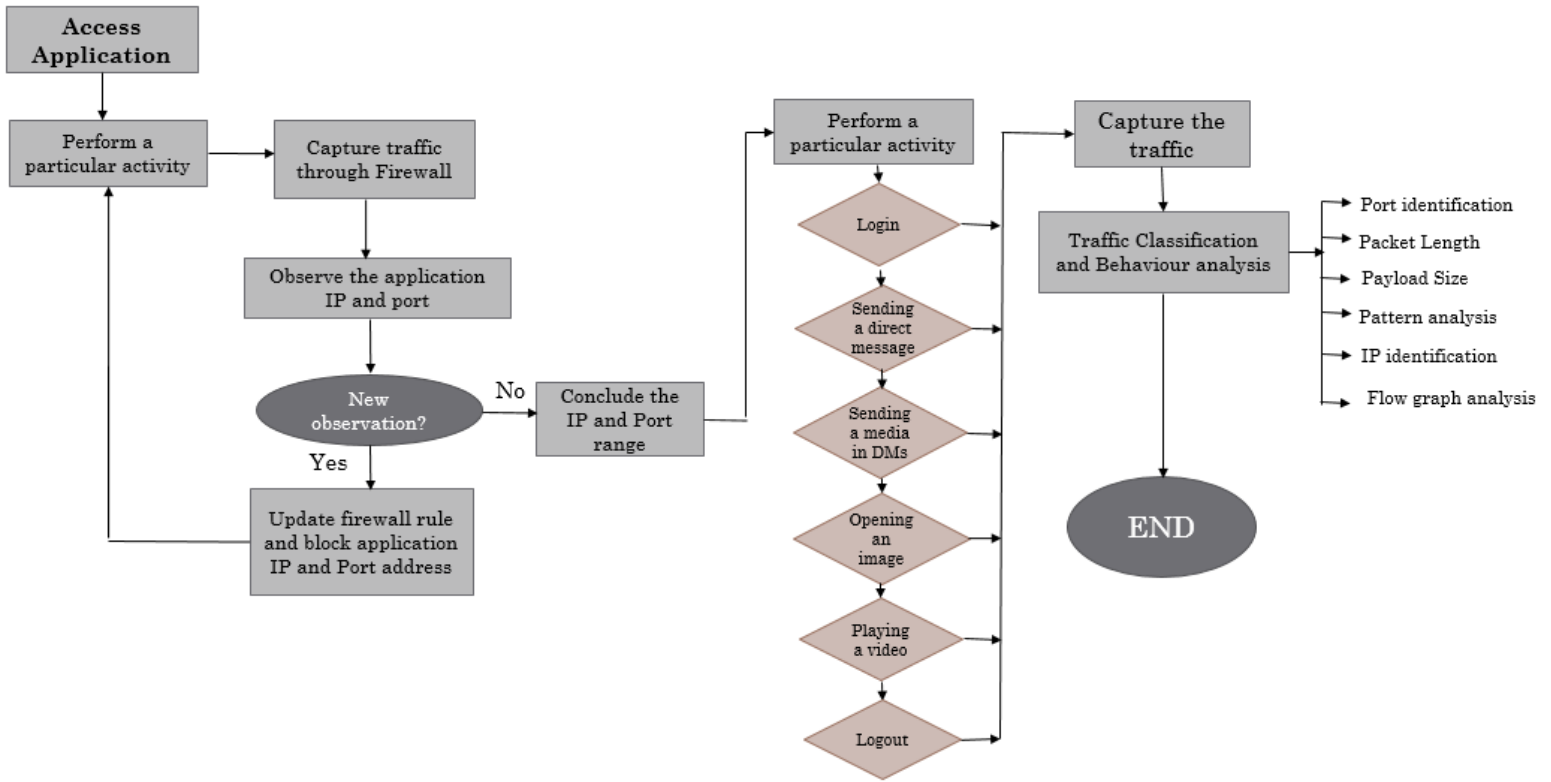


Figure 7. Analysis Framework

### 3.4 Summary

In this chapter, the research methodology from problem statement to report writing has been discussed with reference to the NIST guide. The proposed analysis framework is defined in the end which is mapped to the research process.

# Chapter 4

## EXPERIMENTAL SETUP

### 4.1 Introduction

Experimental setup and its understanding are the foundations of forensics study. In network forensics, a prerequisite of analysis is the deployment of networking devices at appropriate places and with appropriate configurations. In this chapter, an environmental setup with tools, technologies and their versions are defined in detail. The setup includes networking devices, a smartphone device, specifications and connection establishment procedure. Further sections are as under:

- **Section 4.2** Tools and Technologies
- **Section 4.3** Specification and Configuration of Devices
- **Section 4.4** Environmental Setup
- **Section 4.5** Smartphone setup
- **Section 4.6** Summary

### 4.2 Tools and Technologies

Several user friendly tools helped us in carrying out the tasks effectively. These tools are categorised into hardware and software and are defined below.

a. Hardware:

- Firewall
- Cisco Switch
- Wireless Access Point (WAP)
- Samsung smartphone
- Desktop Computer

b. Software:

- Wireshark
- Windows OS
- Android OS
- Twitter application

### 4.3 Device Configurations and Specifications

Versions and specifications of tools, software and devices are defined below:

- Switch (SG300-52 port Gigabit Cisco Layer-3) used for WAN connection

- Firewall (2.4.4-RELEASE-p3 and FreeBSD version 11.2-RELEASE-p3)
- Targeted smartphone (Samsung Galaxy J7)
- Wireless Access Point/Wi-Fi router (PTCL, DSL-2750U)
- Client PC (Windows 10 Pro) used for packet sniffing and analysis purposes
- Wireshark (version-3.2.1) Software used for packet sniffing
- Android Operating Software (version-5.1.1)
- Twitter application (version-8.22.0-release.00)

## 4.4 Environmental Setup

In order to intercept and capture the traffic, we created a network infrastructure with firewall deployment. Environmental setup is divided into two parts. In this first part, we established a network with a firewall to filter the traffic and identify port and IP address range. To capture the traffic for behaviour analysis, we established the network without a firewall. Subsections below explain the entire setup.

### 4.4.1 Setup with firewall deployment

The created setup is shown in Figure 8 below

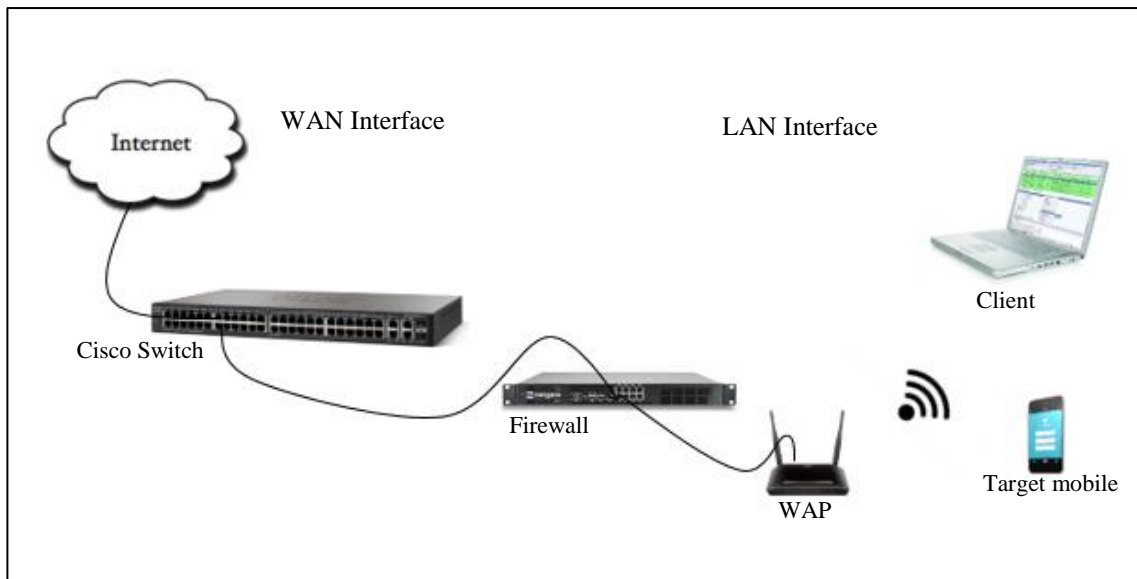


Figure 8. Environmental Setup with Firewall

#### 4.4.1.1 The firewall

Network infrastructure shown in Figure 8 was created to intercept the traffic under a controlled environment. A software based Firewall pfSense was used to filter the traffic. PfSense is a FreeBSD distribution used as a firewall, router and Unified Threat Management and has a user-friendly interface [52] [53] [54]. The figure shows that firewall has two connections: Wireless Area Network (WAN) and a Local Area Network (LAN).

The WAN interface of firewall is connected to the switch and the LAN interface is connected to Wireless Access Point/Wi-Fi router. Packets were captured from the 'Packet Capture' feature available in firewall and the captured .pcap files were downloaded for analysis. The rules on firewall were updated to filter the traffic according to the findings.

- **Usage of Firewall**

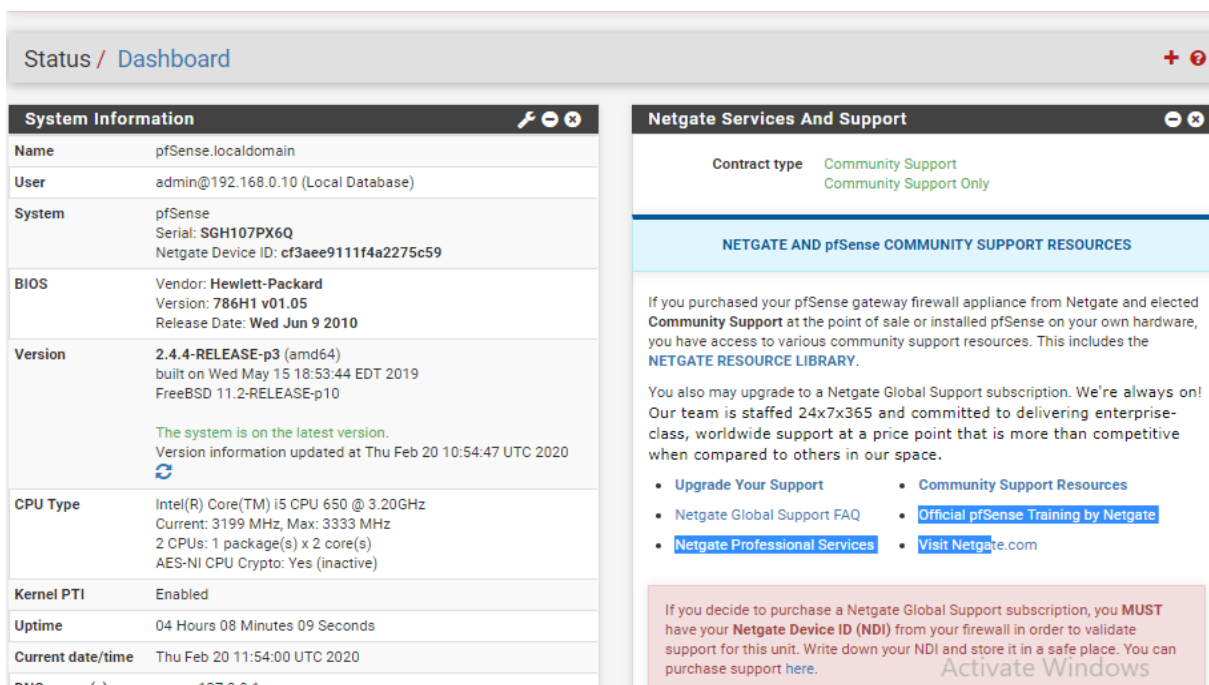


Figure 9. Firewall Dashboard

Application developers design an application with flexibility to ensure the availability of services. In this research, firewall is used to discover the hidden design flexibilities of Twitter application. There are multiple servers distributed to provide alternate connections, in case any server gets blocked.

In this research, the purpose of using firewall is to find out possible alternate connections of Twitter by applying some restrictions on known servers. This analysis will help us in finding out the alternate possible server IP address ranges which Twitter is using for each activity.

Firewall was configured through a web based GUI which was accessed from the client PC connected to the LAN interface. Figure 9 shows the dashboard of the firewall from Web based GUI.

#### 4.4.1.2 The WAN Switch

For internet connectivity we used SG300-52 port Gigabit Cisco layer-3 switch. WAN interface of the firewall is connected to the Switch which provides the internet connectivity. We also applied the firewall rules on the WAN interface to filter the traffic. Figure 10 shows the WAN interface settings of the firewall.

Interfaces / WAN (em0)

**General Configuration**

Enable  Enable interface

Description   
Enter a description (name) for the interface here.

IPv4 Configuration Type

IPv6 Configuration Type

MAC Address   
This field can be used to modify ("spoof") the MAC address of this interface.  
Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank.

MTU   
If this field is blank, the adapter's default MTU will be used. This is typically 1500 bytes but can vary in some circumstances.

MSS   
If a value is entered in this field, then MSS clamping for TCP connections to the value entered above minus 40 (TCP/IP header size) will be in effect.

Speed and Duplex   
Explicitly set speed and duplex mode for this interface.  
WARNING: MUST be set to autoselect (automatically negotiate speed) unless the port this interface connects to has its speed and duplex forced.

**Static IPv4 Configuration**

IPv4 Address  / 8

IPv4 Upstream gateway  [+ Add a new gateway](#)  
If this interface is an Internet connection, select an existing Gateway from the list or add a new one using the "Add" button.  
On local area network interfaces the upstream gateway should be "none". Gateways can be managed by clicking here.

**Reserved Networks**

Block private networks and loopback addresses   
Blocks traffic from IP addresses that are reserved for private networks per RFC 1918 (10/8, 172.16/12, 192.168/16) and unique local addresses per

Figure 10. WAN Interface Settings of Firewall

#### 4.4.1.3 Wireless Access Point/ Wi-Fi router

The LAN interface of the firewall is connected to the Wireless Access Point (WAP). WAP is used to provide internet connection to the target mobile and a PC used within the LAN network. LAN settings of firewall is shown in Figure 11. The WAP settings are given below:

1. Reset the modem
2. Disable the DHCP mode
3. Turn on Wi-Fi on the modem
4. Connect modem with firewall

## 5. Connect PC and target mobile to the Wi-Fi enabled modem

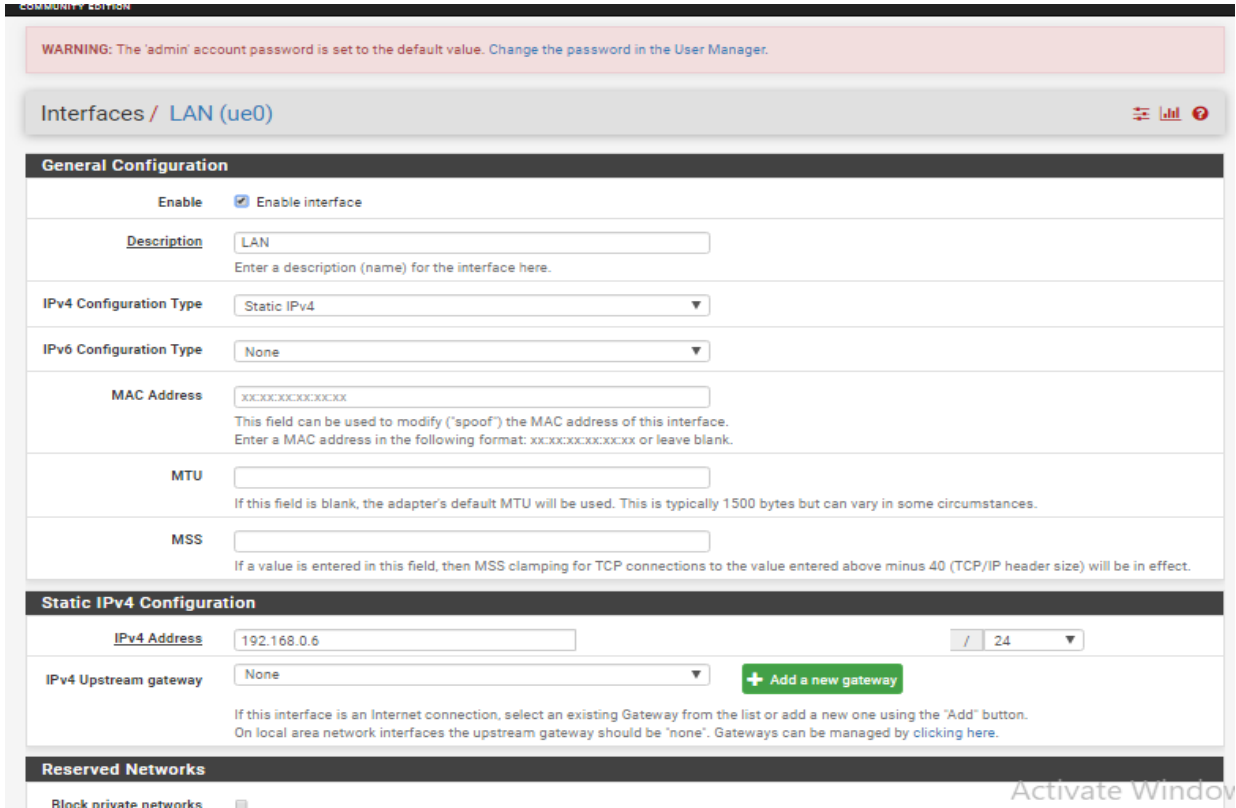


Figure 11. LAN Interface Settings of Firewall

### 4.4.1.4 Configurations

The device configurations are shown in Table 7. The WAN interface was assigned a static IP address 10.2.31.180 and at the LAN interface we enabled the DHCP. The IP address of the target mobile was 192.168.0.2, however, it kept changing. At LAN, the client PC was assigned 192.168.0.1 dynamically. Default Gateway for LAN was 192.168.0.6.

Table 7. Configurations

Device	Configurations
WAN switch	10.2.31.180
Firewall	1 WAN interface, 1 LAN interface
WAP/Wireless Router	192.168.0.6
Target mobile	192.168.0.2
Client PC	192.168.0.1

- **Traffic capturing process**

The goal was to intercept the mobile phone traffic and analyse the traffic behaviour. In the first step, we booted the USB with firewall software and installed it in a Desktop computer. Once the firewall was installed and configured, LAN and WAN interfaces were created and configured. After creating interfaces, we connected the WAN interface with WAN switch and LAN interface with WAP. WAP was connected to two devices, one is the targeted mobile with Twitter application installed and other is the laptop. Laptop was used to access the firewall GUI and capture traffic.

In this setup, traffic was captured from the traffic capture feature of the firewall. For every user activity, we filtered and updated the firewall rules, captured and analysed the traffic. The stepwise process is defined below:

1. Setup firewall
2. Setup LAN and WAN interfaces
3. Configure WAN switch
4. Configure LAN switch
5. Configure WAN and LAN rules on firewall
6. Connect Target mobile to WAP
7. Install and access Twitter application
8. Perform activities and update firewall rules
9. Capture and analyse traffic

#### **4.4.2 Setup without firewall deployment**

For behaviour analysis, an infrastructure was created to intercept the traffic. The target Android mobile was connected to the internet through a hotspot. Laptop was connected to the internet through a layer-3 switch. To capture the traffic from a mobile device, Wireshark was installed on Windows laptop. Wireshark was set to capture the traffic of a mobile phone through a network adapter created for hotspot in Windows 10. The infrastructure is shown in Figure 12 and the process is defined below.

1. Download Wireshark on Windows PC
2. Enable the hotspot on Windows PC
3. Connect Android smartphone with the hotspot
4. Access application and capture the traffic

## 5. Perform different activities and analyse the traffic behaviour

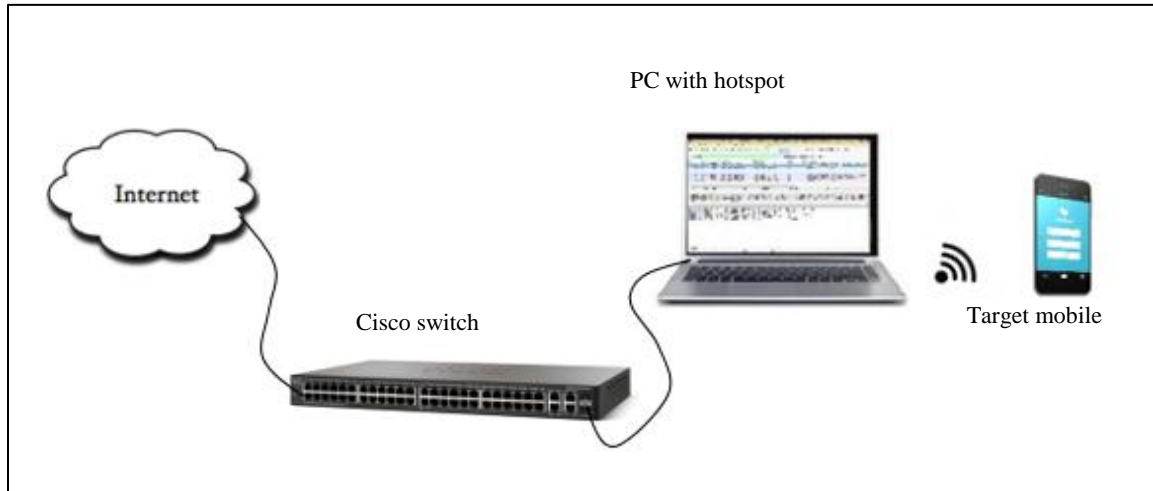


Figure 12. Environmental Setup to Analyse Traffic Behaviour

### 4.5 Smartphone Setup

In this section, application installation, its usage and connection setup are defined.

- **Connection establishment and App installation:**

In the first step, we connected the mobile to WAP, downloaded and installed the Twitter app from Google Play store. After installation, the application was logged in from the created account. Once logged in, we explored the features.

The features of Twitter include the tweets and media uploads, sending direct messages, following and unfollowing the profiles, liking and deleting a tweet, setting a profile photo and updating a bio. Twitter also suggests to link the contacts with a Twitter application.

- **Capturing traffic from mobile phone:**

After installation, the next step is to perform user actions and capture the traffic for analysis. In this step, we enabled the traffic capture feature on the firewall and performed user related activities. Each activity was performed several times and the traffic was analysed for IP range identification.

To analyse the behaviour of traffic, the mobile phone was connected with a laptop as shown in Figure 12. In this scenario, we connected the mobile to hotspot. After that, we opened Wireshark on PC and selected the hotspot interface to capture the mobile traffic. Finally, the packet capture option was enabled and we performed the activities on a mobile phone.



## **4.6 Summary**

In this chapter, the experimental setup for this research was discussed. The setup was divided into two parts, setup with firewall deployment for IP range identification, and without firewall for behaviour analysis. Moreover, device specifications and configurations were defined in detail.

# Chapter 5

## ANALYSIS OF APPLICATION AND RESULTS

### 5.1 Introduction

This chapter presents the network forensic analysis of application and the obtained results. For better understanding figures, graphs and tables are used. Further sections are as follows:

- **Section 5.2** User activities
- **Section 5.3** Fixed packet length observation
- **Section 5.4** Port range identification
- **Section 5.5** IP range identification
- **Section 5.6** Traffic behaviour analysis
- **Section 5.7** Results
- **Section 5.8** Application of the research
- **Section 5.9** Summary

### 5.2 User activities

Considering the application scope, we downloaded and installed Twitter app on Android from Google play store. After the installation, we performed the activities for traffic analysis listed in Table 8.

*Table 8. User Activities Performed on Twitter*

<b>Activity</b>	<b>Description</b>
Login	Using credentials (username and password), a user logs in
Opening an image	User opens any image in media tab
Playing a video	User plays any video in a media tab
Sending a text in Direct Messages (DMs)	User opens a message tab and sends a message to another user
Sending a media in Direct Messages	User opens message tab, attaches a media and sends it
Logout	User logs out from the application

### 5.3 Fixed packet length observation

While performing traffic analysis, it was observed that certain packets always had a fixed length. The length for these packets were observed on the network with and without firewall deployment. Tables below show the fixed packet length of four packets from the client-server session.

Table 9. Packet length without firewall deployment

Session	Packet	Length
Client → Server	SYN	74
Server → Client	SYN, ACK	134
Client → Server	ACK	66
Server → Client	ACK	118

Table 10. Packet length with firewall deployment

Session	Packet	Length
Client → Server	SYN	74
Server → Client	SYN, ACK	74
Client → Server	ACK	66
Server → Client	ACK	66

Table 9 shows that from client to server the [SYN] packet length is fixed to 74 bytes and [ACK] packet has a 66 bytes length, the server to client [SYN, ACK] packet length is 134 bytes and [ACK] packet is 118 bytes. However, when a firewall was installed on the network, length was different for these packets. From Table 10, it can be observed that the client to server [SYN] packet and the server to client [SYN, ACK] packet had always similar fixed lengths of 74 bytes and the [ACK] packets from server to client and client to server had a fixed length of 66 bytes.

#### 5.4 Port range identification

Twitter only uses TCP port 443 (HTTPS) for network traffic. Since twitter.com exchanges the data using TLS therefore, blocking this port will block the entire traffic. However, to find the alternative ports, we performed an experiment and blocked TCP port 443 on our network. After blocking this port, we accessed the application and captured the traffic. It was observed that the client was not able to access the application which shows that it only uses TCP port 443. The results are shown in Figure 50.

Figure 13 shows that, after blocking TCP port 443, the client 192.168.1.4 sent a connection request to api.twitter.com and twitter.com from two different client TCP ports. It is observed that the connection was not successful and a TCP retransmission packet was received. Hence, it is verified that the port range only includes TCP 443 for Twitter application.

Source	Destination	Protocol	Length	Info
192.168.1.4	api.twitter.com	TCP	74	39528 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	api.twitter.com	TCP	74	[TCP Retransmission] 39528 → https(443) [SYN]
192.168.1.4	twitter.com	TCP	74	53478 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	twitter.com	TCP	74	[TCP Retransmission] 53478 → https(443) [SYN]

Figure 13. Identification of Port Range

## 5.5 IP range identification

When a client accesses the application, it establishes multiple connections with different Twitter servers for load balancing. This section discusses the process and identification of those alternate servers. We accessed the application, performed user related activities, captured and filtered the traffic for newly observed IP addresses using a firewall. It was observed that Twitter server redirected its traffic to alternate connections to communicate with Twitter clients. In this study, observed IP addresses for login, media and chat server are valid for Pakistan region. However, the server IP addresses used by Twitter to provide different services can vary in different regions due to distributed architecture.

### 5.5.1 Login Server

We accessed the application initially without any restriction on firewall and identified the server IP addresses. The observed IP addresses were blocked and the application was accessed again. This step was repeated until the client was unable to access the application and the obtained IP address range was concluded.

- **Accessing without restrictions**

Application was accessed using login activity under a controlled environment. Initially, it was observed that every time the client logged in, it established a connection with Twitter IPs in Range-I shown in Table 11. The behaviour of login activity was observed against multiple captured files which showed that the client kept communicating with the same server IP addresses. Connection with Range-I IP addresses is

Source	Destination	Protocol	Length	Info
192.168.0.2	104.244.42.66	TCP	74	39457 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.0.2	104.244.42.65	TCP	74	42252 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.0.2	s.twitter.com	TCP	74	42902 → https(443) [SYN] Seq=0 Win=65535 Len=0

Figure 14. Observed Range-I for Login

shown in Figure 14. IP against s.twitter.com is 104.244.42.131. The flow graph of the above observation is shown in Figure 15.

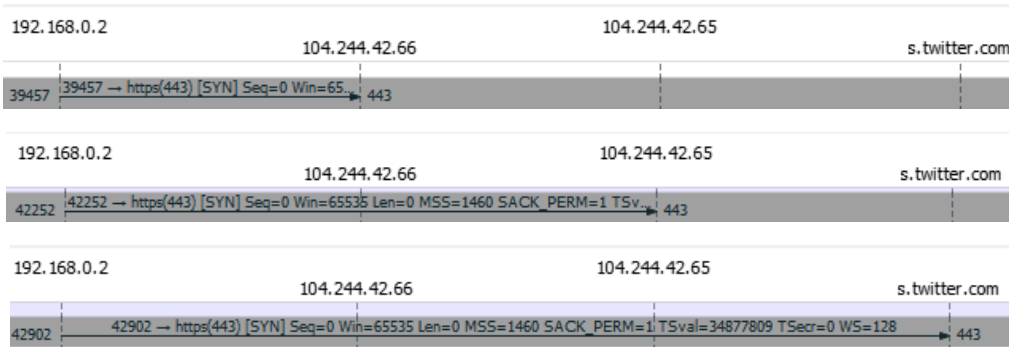


Figure 15. Flow Graph for Range-I IP Addresses

- **Blocking the Range-I IP addresses**

We applied rules on firewall, by creating an alias of hosts containing Range-I IP addresses. A rule was created on LAN and WAN interfaces to block the IP addresses. The rules are as under:

- Rule on LAN interface: For outgoing traffic the rules are defined on LAN interface.
- Rules on WAN interface: For incoming traffic the rules are defined on WAN interface.

For our study, rules were applied on the LAN and WAN interface to block incoming traffic and to allow outgoing traffic to and from defined hosts. Figure 16 shows the alias created to block Range-I IP addresses and Figure 17 shows the rules to block the alias on LAN interface.



Figure 16. Alias Created for Range-I IP Addresses

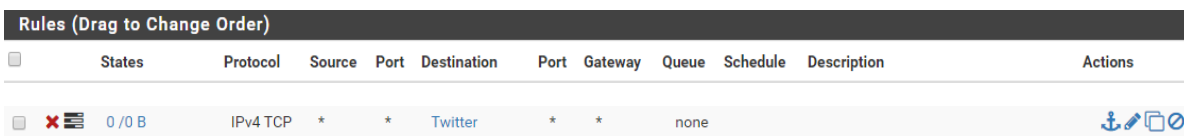


Figure 17. LAN Rule to Block Range-I IP Addresses

After blocking the Range-I, application was accessed again. It was observed that the client 192.168.0.2 tried to establish the connection with 104.244.42.66 and 104.244.42.65. When the client failed to connect to these hosts, it established the connection with Range-II server IP addresses. Trace for failed connection is shown in Figure 18.

Figure shows that 104.244.42.66 and 104.244.42.65 were blocked therefore, the client failed to establish the connection and hence the packets were retransmitted. Trace for Range-II IP addresses is shown in Figure 19. The figure shows that after blocking the traffic from Range-I IP addresses, the client sent the SYN packets to 104.244.42.130, 104.244.42.195 (s.twitter.com), 69.195.186.128 and received SYN, ACK packets.

2.677969	192.168.0.2	104.244.42.66	TCP	74	43920 → https(443) [SYN] Seq=0 Win=65535
2.678216	192.168.0.2	104.244.42.66	TCP	74	55332 → https(443) [SYN] Seq=0 Win=65535
2.678341	192.168.0.2	104.244.42.66	TCP	74	46547 → https(443) [SYN] Seq=0 Win=65535
3.595704	192.168.0.2	104.244.42.66	TCP	74	[TCP Retransmission] 55332 → https(443)
3.596704	192.168.0.2	104.244.42.66	TCP	74	[TCP Retransmission] 43920 → https(443)
3.596830	192.168.0.2	104.244.42.66	TCP	74	[TCP Retransmission] 46547 → https(443)

Time	Source	Destination	Protocol	Length	Info
94.244265	192.168.0.2	twitter.com	TCP	74	42271 → https(443) [SYN] Seq=0 Win=65535 Len=0
94.398013	192.168.0.2	twitter.com	TCP	74	42272 → https(443) [SYN] Seq=0 Win=65535 Len=0
95.197128	192.168.0.2	twitter.com	TCP	74	[TCP Retransmission] 42271 → https(443) [SYN]
95.413001	192.168.0.2	twitter.com	TCP	74	[TCP Retransmission] 42272 → https(443) [SYN]

Figure 18. Error Packets after Blocking Range-I

Time	Source	Destination	Protocol	Length	Info
92.822534	192.168.0.2	104.244.42.130	TCP	74	38979 → https(443) [SYN] Seq=
92.890853	104.244.42.130	192.168.0.2	TCP	74	https(443) → 33327 [SYN, ACK]
195.279191	192.168.0.2	s.twitter.com	TCP	74	36893 → https(443) [SYN] Seq=0
195.434437	s.twitter.com	192.168.0.2	TCP	74	https(443) → 36893 [SYN, ACK]
199.173270	192.168.0.2	69.195.186.128	TCP	74	51456 → https(443) [SYN] Seq=
199.328557	69.195.186.128	192.168.0.2	TCP	74	https(443) → 51456 [SYN, ACK]
205.137311	192.168.0.2	69.195.161.128	TCP	74	34674 → https(443) [SYN]

Figure 19. Observed Range-II for Login

- **Blocking the Range-II**

In the next step, we blocked the Range-II IP addresses and accessed the application. After the analysis, it was observed that the client tried to establish the connection with 104.244.42.65, 104.244.42.195 which failed. However, the client successfully established the connection with server in Range-III. The results for above observations are shown in Figure 20.

- **Blocking the Range-III:**

In this case, Range-III was blocked and the application was login. The analysis showed that the client was still able to communicate and the Range-IV was observed. The client also sent a failed connection request to Range-III IP address. The failed observation is shown in Figure 21.

Time	Source	Destination	Protocol	Length	Info
10.216486	192.168.0.2	twitter.com	TCP	74	42312 → https(443) [SYN] S
11.001850	192.168.0.2	twitter.com	TCP	74	[TCP Retransmission] 42312
17.267640	192.168.0.2	probe.twitter.com	TCP	74	37688 → https(443) [SYN] S
18.260626	192.168.0.2	probe.twitter.com	TCP	74	[TCP Retransmission] 37688

Time	Source	Destination	Protocol	Length	Info
4.253319	192.168.0.2	104.244.42.194	TCP	74	50187 → https(443) [SYN] Seq=
4.406680	104.244.42.194	192.168.0.2	TCP	74	https(443) → 50187 [SYN, ACK]

Figure 20. Failed Connection with Range-II and Observed Range-III

Time	Source	Destination	Protocol	Length	Info
5.149057	192.168.0.2	104.244.42.194	TCP	74	55011 → https(443) [SYN] Seq=
6.257166	192.168.0.2	104.244.42.194	TCP	74	[TCP Retransmission] 55011 →
	192.168.0.2	104.244.42.129	TCP	74	33673 → https(443) [SYN] Seq=0
	104.244.42.129	192.168.0.2	TCP	74	https(443) → 33673 [SYN, ACK]

Figure 21. Failed Connection with Range-III

- **Blocking the Range-IV**

When we accessed the application after blocking Range-IV, the client established connection with a delay of 15 minutes. After capturing and analysing the traffic, two new server IP addresses were observed referred to as Range-V in Table 11. The client also tried to establish the connection with 104.244.42.66, 104.244.42.194, 104.244.42.130 and 104.244.42.129. The results are shown in Figure 22 and 23.

Time	Source	Destination	Protocol	Length	Info
1.286610	192.168.0.3	104.244.42.66	TCP	74	46343 → https(443) [SYN] Seq=0 Win=65535 Len=
2.287846	192.168.0.3	104.244.42.66	TCP	74	[TCP Retransmission] 46343 → https(443) [SYN]
95.290700	192.168.0.2	104.244.42.194	TCP	74	59852 → https(443) [SYN] Seq=0 Win=65535 Len=0
96.280437	192.168.0.2	104.244.42.194	TCP	74	[TCP Retransmission] 59852 → https(443) [SYN]
185.375470	192.168.0.2	104.244.42.130	TCP	74	42237 → https(443) [SYN] Seq=0 Win=65535 Len=0
186.437206	192.168.0.2	104.244.42.130	TCP	74	[TCP Retransmission] 42237 → https(443) [SYN]
1.303361	192.168.0.3	twitter.com	TCP	74	36295 → https(443) [SYN] Seq=0 Win=65535 Len=0
2.288223	192.168.0.3	twitter.com	TCP	74	[TCP Retransmission] 36295 → https(443) [SYN]

Figure 22. Error Packets for Blocked IP Addresses

Time	Source	Destination	Protocol	Length	Info
6.924158	192.168.0.2	probe.twitter.com	TCP	74	53203 → https(443) [SYN] Seq=0
7.079309	probe.twitter.com	192.168.0.2	TCP	74	https(443) → 53203 [SYN, ACK] S
275.486991	192.168.0.2	104.244.42.2	TCP	74	52754 → https(443) [SYN] Seq=0
275.636728	104.244.42.2	192.168.0.2	TCP	74	https(443) → 52754 [SYN, ACK] S

Figure 23. Observed Range-V

- **Blocking the Range-V**

After denying access from Range-V and previously observed IP address ranges, client was unable to login to the application and was redirected to the login page. The results showed that the client kept trying to communicate with 104.244.42.66, 104.244.42.194 and 104.244.42.130 and failed to establish the connection because these IPs were blocked. This concludes the server range identification for login activity. The results for this observation are shown in Figure 24 below. Table 11 below shows all the servers used to access the Twitter application. Figure 25 shows the infrastructure of Twitter app servers.

Time	Source	Destination	Protocol	Length	Info
4.061320	192.168.0.2	104.244.42.66	TCP	74	55026 → https(443) [SYN] Seq=0 Win=65535 Len=0
5.048553	192.168.0.2	104.244.42.66	TCP	74	[TCP Retransmission] 55026 → https(443) [SYN]
192.168.0.2	104.244.42.194	TCP	74	48047 → https(443) [SYN] Seq=0 Win=65535 Len=0	
192.168.0.2	104.244.42.194	TCP	74	[TCP Retransmission] 48047 → https(443) [SYN]	
184.273241	192.168.0.2	104.244.42.130	TCP	74	45207 → https(443) [SYN] Seq=0 Win=65535 Len=0
185.258226	192.168.0.2	104.244.42.130	TCP	74	[TCP Retransmission] 45207 → https(443) [SYN]

Figure 24. Error Packets for Blocked IP Addresses

- **Parallel Connections**

As mentioned earlier that Twitter Inc. has a distributed architecture, that uses many intermediate servers for different purposes. While the client logged in the application, it did not only communicate with Twitter app server but also established parallel connections with different servers belonging to Twitter Inc. These servers are described in parallel connections. From the analysis, it was observed that parallel connections provided by Twitter are 209.237.0.0 to 209.237.255.255, 69.195.0.0 to 69.195.255.255 and 104.244.0.0 to 104.244.4255.255. The list of parallel connections is shown in Table 12 and Figure 26 shows the infrastructure diagram for all parallel connections.



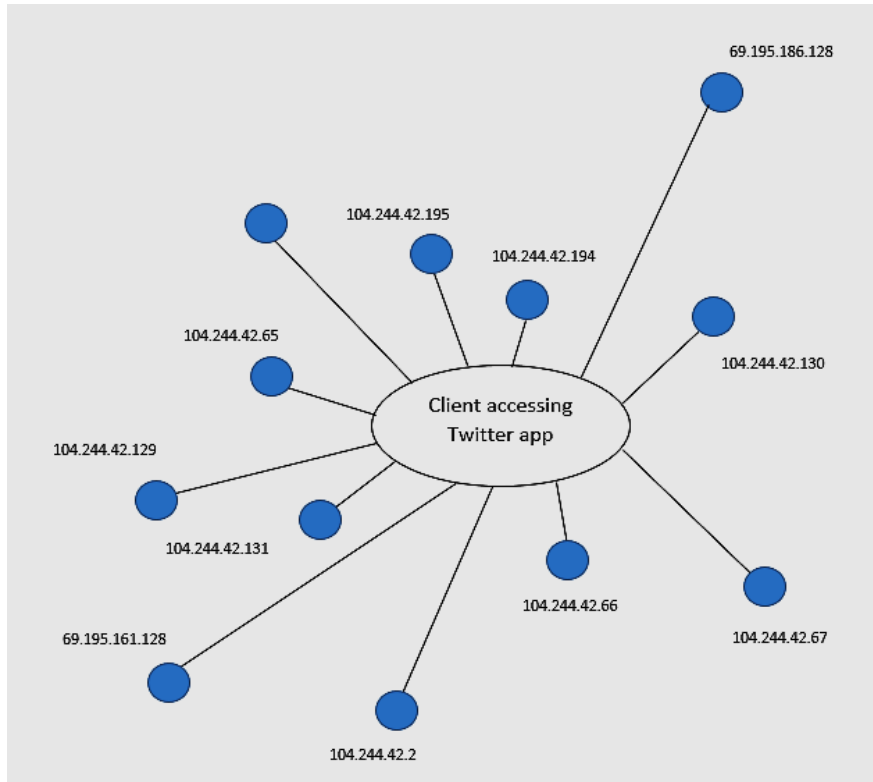


Figure 25. Server Infrastructure for Login

Table 11. Server Range for Accessing Twitter

Activity	Observed range
Accessing without restrictions	Range I: 104.244.42.66, 104.244.42.65, 104.244.42.131
Blocked Range-I	Range II: 104.244.42.130, 104.244.42.195, 69.195.186.128, 69.195.161.128
Blocked Range II	Range III: 104.244.42.194
Blocked Range III	Range IV: 104.244.42.129
Blocked Range IV	Range V: 104.244.42.67, 104.244.42.2
Blocked Range-V	User was unable to log into the Twitter application.

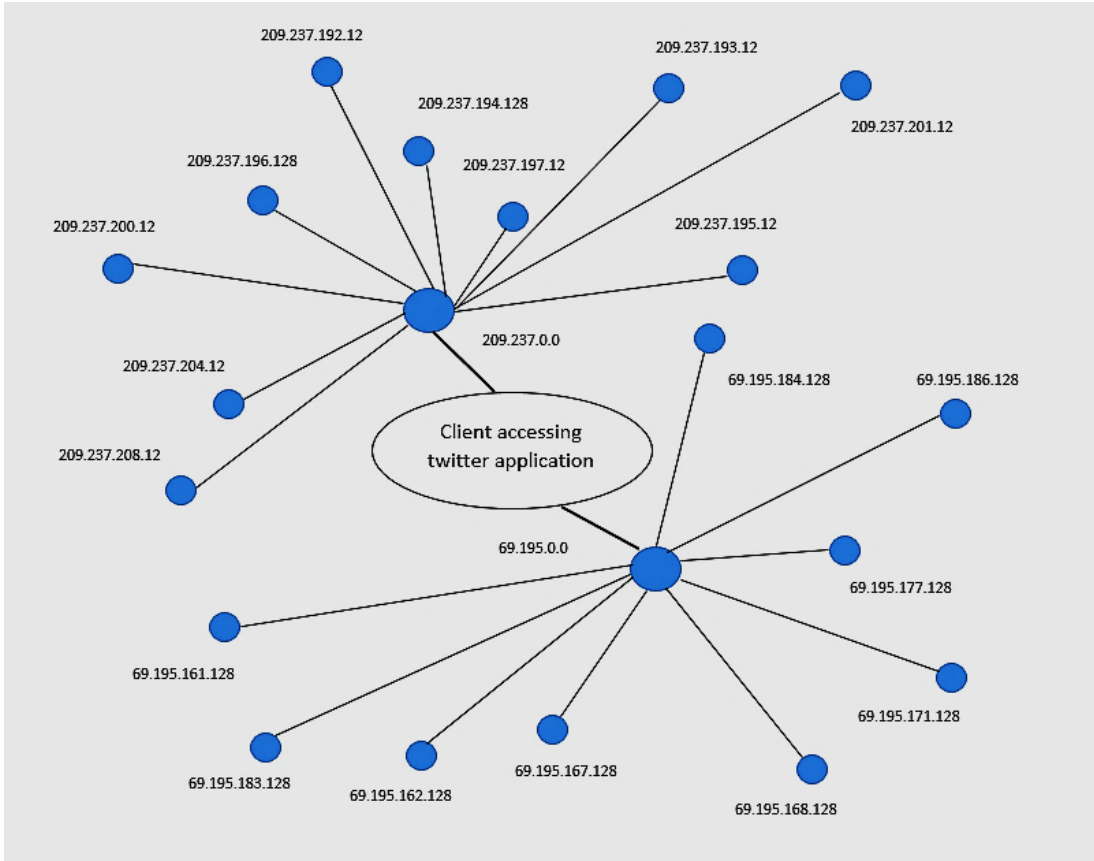


Figure 26. Twitter Infrastructure with Parallel Connection

Table 12. Parallel connections

Parallel Connections				
209.237.192.128	209.237.194.128	209.237.196.128	209.237.200.128	209.237.204.128
209.237.193.128	209.237.195.128	209.237.197.128	209.237.201.128	209.237.208.128
69.195.161.128	69.195.162.128	69.195.167.128	69.195.168.128	69.195.171.128
69.195.177.128	69.195.183.128	69.195.184.128	69.195.186.128	-

### 5.5.2 Opening an image

When a client opened an image, it was observed that it always communicated with 93.184.220.70. Further analysis showed that the client established another connection with 192.229.220.133 which belongs to Verizon Communications. In the next step, we blocked these IP addresses and observed the IP range for image server.

- **Blocked 93.184.220.70:** We blocked the traffic from 93.184.220.70 and captured the traffic to analyse the results. The results show that the client was still able to open an image.

- **Blocked 192.229.220.133:** In the second step, we blocked 192.229.220.133 to find if the image is still accessible. It was observed that the client was still able to open an image. After the analysis, the new IP address was observed to be within the same range which is 192.229.233.50.
- **Blocked 192.229.220.133, 192.229.233.50 and 93.184.220.70:** After we applied restrictions on observed IP ranges for the media server, it was observed that the client was no longer able to open an image. This shows that these IP addresses are reserved for providing image services to Twitter Inc. Figure 27 shows the traces for client request and response with 93.184.220.70.

Source	Destination	Protocol	Length	Info
192.168.1.4	93.184.220.70	TCP	74	43100 → https(443) [SYN] Seq=
93.184.220.70	192.168.1.4	TCP	74	https(443) → 43100 [SYN, ACK]

Figure 27. Connection Established with 93.184.220.70

It shows the results for accessing an image before applying any restriction on the network. When the client opens an image, a SYN packet is sent from the client 192.268.1.4 to the 93.184.220.70. In response, the client receives the SYN, ACK packet showing a successful connection. Figure 28 shows the results after applying restrictions on 93.184.220.70. It shows that the client can now communicate with 192.229.220.133 for accessing an image.

Source	Destination	Protocol	Length	Info
192.168.1.4	192.229.220.133	TCP	74	55552 → https(443) [SYN] Seq=
192.229.220.133	192.168.1.4	TCP	66	https(443) → 55552 [SYN, ACK]

Figure 28. Connection Established with 192.229.220.133

After blocking 93.184.220.70 and 192.229.220.133, packets were retransmitted, and the connection was failed as shown in Figure 29. Now, the client accessed the image with 192.229.233.50 and exchanged packets with it as shown in Figure 30.

Source	Destination	Protocol	Length	Info
192.168.1.4	cs45.wac.edgecastcd...	TCP	74	33263 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	cs45.wac.edgecastcd...	TCP	74	[TCP Retransmission] 33263 → https(443) [SYN]
192.168.1.4	192.229.220.133	TCP	74	59533 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	192.229.220.133	TCP	74	[TCP Retransmission] 59533 → https(443) [SYN]

Figure 29. Failed Connection with 93.184.220.70 and 192.229.220.133

Source	Destination	Protocol	Length	Info
192.168.1.4	192.229.233.50	TCP	74	39765 → https(443) [SYN] Seq=0
192.229.233.50	192.168.1.4	TCP	74	https(443) → 39765 [SYN, ACK]
192.229.233.50	192.168.1.4	TLSv1.2	1514	Application Data
192.229.233.50	192.168.1.4	TCP	1514	https(443) → 39765 [ACK] Seq=6108 Ack=1336 Win=148992 Len=1448
192.229.233.50	192.168.1.4	TCP	1514	https(443) → 39765 [ACK] Seq=7556 Ack=1336 Win=148992 Len=1448
192.229.233.50	192.168.1.4	TCP	1514	https(443) → 39765 [ACK] Seq=9004 Ack=1336 Win=148992 Len=1448
192.229.233.50	192.168.1.4	TCP	1514	https(443) → 39765 [ACK] Seq=10452 Ack=1336 Win=148992 Len=1448
192.229.233.50	192.168.1.4	TCP	1514	https(443) → 39765 [ACK] Seq=11900 Ack=1336 Win=148992 Len=1448
192.229.233.50	192.168.1.4	TCP	1514	https(443) → 39765 [ACK] Seq=13348 Ack=1336 Win=148992 Len=1448
192.229.233.50	192.168.1.4	TCP	1514	https(443) → 39765 [ACK] Seq=14796 Ack=1336 Win=148992 Len=1448

Figure 30. Connection Established with 192.229.233.50

Figure 31 shows the failed connections when the media server was blocked. It concludes that the client tried to establish connection with three servers 93.184.220.70, 192.229.220.133 and 192.229.233.50. In the end, the client was unable to open an image as shown in Figure 32. The figure shows that when the media server is blocked, the images cannot be seen and when it is allowed, the images can clearly be seen.

Source	Destination	Protocol	Length	Info
192.168.1.4	cs45.wac.edgecastcd...	TCP	74	52401 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	cs45.wac.edgecastcd...	TCP	74	[TCP Retransmission] 52401 → https(443) [SYN]
192.168.1.4	cs45.wac.edgecastcd...	TCP	74	[TCP Retransmission] 52401 → https(443) [SYN]
192.168.1.4	192.229.233.50	TCP	74	38895 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	192.229.233.50	TCP	74	[TCP Retransmission] 38895 → https(443) [SYN]
192.168.1.4	192.229.233.50	TCP	74	[TCP Retransmission] 38895 → https(443) [SYN]
192.168.1.4	192.229.233.50	TCP	74	[TCP Retransmission] 38895 → https(443) [SYN]
192.168.1.4	192.229.233.50	TCP	74	56332 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	192.229.233.50	TCP	74	49398 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	192.229.233.50	TCP	74	[TCP Retransmission] 56332 → https(443) [SYN]
192.168.1.4	192.229.233.50	TCP	74	49776 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	192.229.233.50	TCP	74	[TCP Retransmission] 49398 → https(443) [SYN]
192.168.1.4	192.229.233.50	TCP	74	[TCP Retransmission] 49776 → https(443) [SYN]
192.168.1.4	192.229.233.50	TCP	74	[TCP Retransmission] 56332 → https(443) [SYN]
192.168.1.4	192.229.220.133	TCP	74	57092 → https(443) [SYN] Seq=0 Win=65535 Len=0
192.168.1.4	192.229.220.133	TCP	74	[TCP Retransmission] 57092 → https(443) [SYN]
192.168.1.4	192.229.220.133	TCP	74	[TCP Retransmission] 57092 → https(443) [SYN]
192.168.1.4	192.229.220.133	TCP	74	[TCP Retransmission] 57092 → https(443) [SYN]
192.168.1.4	192.229.220.133	TCP	74	[TCP Retransmission] 57092 → https(443) [SYN]

Figure 31. Failed Connections for Opening an Image



Figure 32. Images After Blocking and Unblocking the Image Server

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
192.168.1.4	52401	93.184.220.70	443	5	370	5	370	0	0
192.168.1.4	34169	93.184.220.70	443	5	370	5	370	0	0
192.168.1.4	40065	93.184.220.70	443	5	370	5	370	0	0

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
192.168.1.4	55942	192.229.220.133	443	7	518	7	518	0	0
192.168.1.4	51424	192.229.220.133	443	7	518	7	518	0	0
192.168.1.4	52284	192.229.220.133	443	7	518	7	518	0	0
192.168.1.4	36143	192.229.220.133	443	7	518	7	518	0	0
192.168.1.4	54000	192.229.220.133	443	7	518	7	518	0	0
192.168.1.4	51289	192.229.220.133	443	7	518	7	518	0	0
192.168.1.4	34126	192.229.220.133	443	6	444	6	444	0	0
192.168.1.4	37786	192.229.220.133	443	6	444	6	444	0	0

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A
192.168.1.4	33491	192.229.233.50	443	7	518	7	518	0	0
192.168.1.4	56429	192.229.233.50	443	7	518	7	518	0	0

Figure 33. Conversation with Blocked IP Addresses

Figure 33 shows the conversation statistics of 192.168.1.4 with 93.184.220.70, 192.229.220.133 and 192.229.233.50. Here A represents the client IP, and B represents the server IP. The server uses TCP port 443 and the client port is varying. Figure shows that five packets were sent to 93.184.220.70 of 370 Bytes. Seven packets are sent to 192.229.220.133 and 192.229.233.50 of 518 Bytes. It is also observed that B does not send any packet to A, since the image server is blocked.

### 5.5.3 Text Message

In this activity, a text message was sent to identify the IP address range for chat server. The traffic was captured and blocked for each observed IP address to obtain the possible range.

- **Sending a message without restrictions:** In the first step, we sent a text message without restrictions on the firewall. It was observed that the client established a connection with Range-VI shown in Table 13.
- **Blocked Range-VI:** We blocked the Range-VI IP addresses and noticed that the client was still able to send a message. This observation shows that the client communicated with IP addresses in Range VII.
- **Blocked Range-VII:** Upon blocking the Range-VII, the client was still able to perform the chat activity via Range-VIII server.
- **Blocked Range-VIII:** After blocking this Range, the communication continued via Range-IX server IP addresses.
- **Blocked Range-IX:** When we blocked the Range-IX and previously observed IP addresses, the client was not able to send direct messages. This concludes that the observed servers are used for chat activity. The possible infrastructure for chat servers is shown in Figure 34.



Figure 34. Infrastructure of Chat Server

### 5.5.4 Media as a message

In the section above, we performed a chat activity to send text messages on Twitter. In this activity we sent a media (an image) in direct messages, and captured and blocked the observed IP address range.

Table 13. Server Range for Sending a text in DMs

Activity	Observed Range
Sending a text without restrictions	Range VI: 104.244.42.67, 69.195.160.128
Blocked Range VI	Range VII: 104.244.42.66
Blocked Range VII	Range VIII: 104.244.42.3, 104.244.42.194
Blocked Range VIII	Range IX: 104.244.42.193, 104.244.42.129, 104.244.42.2
Blocking IX	Client was unable to send a text in direct messages

- **Sending a media in direct messages:**

We sent an image in a direct message and analysed the traffic. The initial IP addresses were analysed and restricted to get the IP address range. Initially, when the client attached an image, a connection was established with 104.244.42.148 (ton.twitter.com) and server IP addresses in Range X shown in Table 14.

During this activity, some patterns were observed against a new server IP 13.35.180.119. From further research, it was concluded that this server IP belongs to host server-13-35-180-119.fjr50.r.cloudfront.net and the organization Amazon.com Inc. [55]. Amazon CloudFront is one of the fast Content Delivery Network CDN services which delivers data including videos, images and applications to the customers in a very secure and fast transfer rate. CloudFront works with AWS services to provide load balancing to organization traffic and to avoid Distributed Denial of Service attacks.

- **Blocked Range-X:** In the next step, we blocked the Range-X and sent an image. The observed IP addresses are referred to as Range-XI.
- **Blocked the Range-XI:** In this step, the client sent an image and the Range-XII was observed.
- **Blocked the Range-XII:** After blocking these IP addresses, we figured that the client established connection with IP addresses shown in Range-XIII.
- **Blocked the Range-XIII:** The client was not able to send an image in direct messages over the network. The list of IPs for sending media in direct messages is shown in Table 14 below.

Table 14. Server Range for Sending Media in DMs

Activity	Observed range
Sending a media without restrictions	Range-X: 104.244.42.148, 104.244.42.194, 104.244.42.195, 69.195.168.128, 69.195.167.128, 13.35.180.119
Blocked Range-X	Range-XI: 143.204.106.90, 104.244.42.2, 104.244.42.131

Blocked Range-XI	Range-XII: 104.244.42.84, 143.204.106.11, 143.204.106.28
Blocked Range-XII	Range-XIII: 104.244.42.66, 104.244.42.203, 104.244.42.75 (upload.twitter.com)
Blocked Range-XIII	No connection with chat server

## 5.6 Traffic behaviour analysis

According to the study, encrypted traffic can provide an amount of significant artefacts related to user if extensively analysed. After the identification of server IP ranges, the challenging part of this research was to analyse the encrypted traffic. In this section, we accessed the application, performed activities, and captured and analysed the traffic behaviour. Each activity was repeated 10 times and traffic was analysed from each trace file. The results show that application traffic was correctly identified and the user actions were further classified for behaviour analysis.

### 5.6.1 Login

We logged into the Twitter on Android Galaxy J7 and captured the traffic using Wireshark. The first step was to identify the application traffic on the network which can be done through different methods. One method is to perform the same activity multiple times and capture the synced traffic to identify the particular application. In the initial steps, we accessed the application multiple times and observed similar patterns. From the research, it was concluded that obtained IP addresses belong to Twitter. Initially, the captured traffic had packets for notifications on android phone and application updates etc. We applied the Wireshark filters and filtered the Twitter traffic for analysis. User login behaviour is defined in the following sub sections.

- **Connection establishment**

During the sign-in process, it was observed that a DNS query is sent to a local server for api.twitter.com. DNS server responds with available servers against api.twitter.com. Figure 35 shows the DNS query and response for api.twitter.com. The client uses one of these servers to access Twitter applications for login.

192.168.137.230	192.168.137.1	DNS	75 Standard query 0x7b8f A api.twitter.com
192.168.137.1	192.168.137.230	DNS	154 Standard query response 0x7b8f A api.twitter.com A 104.244.42.66 A 104.244.42.194 A 104.244.42.130 A 104.244.42.2

Figure 35. DNS Query for Twitter.com

- **TLS Handshake**

Figure 36 describes the TLS Handshake process between client and server. Client 192.168.137.230 initiates a connection with 104.244.42.66 by sending a [SYN] packet on TCP port 58329. The server responds with [SYN, ACK] and the Handshake process is started.



For TLS Handshake, the client sends *Client Hello message* initiating the session with a server. Server responds with *Server Hello message* and sends *Certificate* and *Server Key Exchange*. Client responds with a *Client Key Exchange*, *Change Cipher Spec* and *Encrypted Handshake Message*. Server now sends the *New Session Ticket*, *Change Cipher Spec* and *Encrypted Handshake Message*

192.168.137.230	104.244.42.66	TCP	74	58329 → https(443) [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=14357758 TSeq=0
104.244.42.66	192.168.137.230	TCP	134	https(443) → 58329 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=222294212 TSeq=0
192.168.137.230	104.244.42.66	TCP	66	58329 → https(443) [ACK] Seq=1 Ack=1 Win=87680 Len=0 TSval=14357758 TSeq=0
192.168.137.230	104.244.42.66	TLSv1.2	228	Client Hello
104.244.42.66	192.168.137.230	TCP	118	https(443) → 58329 [ACK] Seq=1 Ack=163 Win=30208 Len=0 TSval=222294212 TSeq=0
104.244.42.66	192.168.137.230	TLSv1.2	3014	Server Hello
104.244.42.66	192.168.137.230	TCP	3014	https(443) → 58329 [ACK] Seq=1449 Ack=163 Win=30208 Len=1448 TSval=222294212 TSeq=0
104.244.42.66	192.168.137.230	TLSv1.2	880	Certificate, Server Key Exchange, Server Hello Done
192.168.137.230	104.244.42.66	TCP	66	58329 → https(443) [ACK] Seq=163 Ack=1449 Win=90496 Len=0 TSval=14357758 TSeq=0
192.168.137.230	104.244.42.66	TCP	66	58329 → https(443) [ACK] Seq=163 Ack=2897 Win=93440 Len=0 TSval=14357758 TSeq=0
192.168.137.230	104.244.42.66	TCP	66	58329 → https(443) [ACK] Seq=163 Ack=3278 Win=96384 Len=0 TSval=14357758 TSeq=0
192.168.137.230	104.244.42.66	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
104.244.42.66	192.168.137.230	TLSv1.2	602	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

Figure 36. TLS Handshake

TLS Handshake is the process of setting up a secure connection between a client and a server. In this protocol, the client initiates the session by sending a client hello message to server [56]. It includes multiple parameters such as TLS protocol version used by client, session identifier, a random number to create an encryption key, Cipher suite supported by client, Cipher Suite length and Compression method. As shown in Figure 37.

```

▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 157
  ▼ Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 153
    Version: TLS 1.2 (0x0303)
    > Random: c013af69b38c108ec7281d7100d25bf09689cde6893bb0cb...
    Session ID Length: 0
    Cipher Suites Length: 24
    > Cipher Suites (12 suites)
    Compression Methods Length: 1
    > Compression Methods (1 method)
    Extensions Length: 88

```

Figure 37. TLS Handshake- Client Hello Packet

Server responds with server hello, certificate and server key exchange. A certificate is sent by the server to authenticate itself which contains its public key. The new session ticket is an optional message and this key is used by the client to encrypt the Client key exchange later. In the end, a server hello done message is sent to show that the server is done and is waiting for the client's response, as shown in Figure 38.

```

  Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 2859
    Handshake Protocol: Certificate
      Handshake Type: Certificate (11)
      Length: 2855
      Certificates Length: 2852
      Certificates (2852 bytes)
  Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 333
    Handshake Protocol: Server Key Exchange
      Handshake Type: Server Key Exchange (12)
      Length: 329
      EC Diffie-Hellman Server Params
  TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 4
    Handshake Protocol: Server Hello Done
      Handshake Type: Server Hello Done (14)
      Length: 0

```

Figure 38. Certificate, Server Key Exchange and Server Hello Done Packet

Client will now respond to the server with client key exchange, change cipher spec and encrypted handshake message containing the hash of all previous messages. It also has a label that the client has completed the negotiation as shown in Figure 39.

```

  Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 70
    Handshake Protocol: Client Key Exchange
      Handshake Type: Client Key Exchange (16)
      Length: 66
      EC Diffie-Hellman Client Params
  TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    Content Type: Change Cipher Spec (20)
    Version: TLS 1.2 (0x0303)
    Length: 1
    Change Cipher Spec Message
  TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 40
    Handshake Protocol: Encrypted Handshake Message

```

Figure 39. Client Key Exchange Packet

The server sends the packet containing the new session ticket containing complete session state, change cipher spec and encrypted message handshake as shown in Figure 40.

After the client successfully performs TLS Handshake, the authentication process will start. Authentication is based on HTTPS where the client randomly selects a session key and encrypts it with server's Public key. When the encrypted data is received by the server it decrypts it with its private key and authenticates the client.

```

Transport Layer Security
  TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 186
  Handshake Protocol: New Session Ticket
    Handshake Type: New Session Ticket (4)
    Length: 182
  TLS Session Ticket
    Session Ticket Lifetime Hint: 129600 seconds (1 day, 12 hours)
    Session Ticket Length: 176
    Session Ticket: 167832042aa360a329b0129c6b683eace375667e9d086bd...
  TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    Content Type: Change Cipher Spec (20)
    Version: TLS 1.2 (0x0303)
    Length: 1
    Change Cipher Spec Message
  TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 40
    Handshake Protocol: Encrypted Handshake Message
  
```

Figure 40. Packet Containing New Session Ticket

The attack: If the attacker gets access to servers' private key and uses it to decrypt the session key, then the attacker can decrypt the entire session.

Twitter Security: Twitter has enabled Forward Secrecy on twitter.com, api.twitter.com and mobile.twitter.com for the traffic over the network. It uses the EC Diffie-Hellman Cipher suites. Using this, the client and server choose a random session key without even sending the secret key over the network. The client then encrypts the information with this session key. In this case, even if the attacker gains access to servers' private key, he/she will not be able to find the session key to decrypt the entire session [57]. Therefore, HTTPS provides secure connection for sensitive information exchange such as usernames, passwords and cookies.

After TLS Handshake, the user logs into the application via HTTPS based authentication. Login server is responsible for authentication which stores the user information and uses it to verify the end user. The user enters the credentials (username and password) via login form which allows the user to login. Server receives the login request and compares the hash of the password against the username already saved in the database. If it matches, the server responds to the client with OK status and the user is logged in. In this analysis, the data is encrypted over the network during login activity. However, behaviour analysis of fixed patterns created during the authentication process will provide significant artefacts. Figure 41 shows the encrypted application data during the authentication process over HTTPS.

The figure shows the fixed patterns created during the authentication process. In packet number 49, client sends encrypted information having 1246 bytes, immediately after the Handshake process.

Against each packet, the server sends a TCP acknowledgment ACK with 0 bytes. The payload size of the ACK message is always 0 because it does not contain any information. Since the packets are encrypted, it is not possible to retrieve the information exchanged between the client and server.

48	0.511910	104.244.42.66	192.168.137.230	TLSv1.2	602 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
49	0.517044	192.168.137.230	104.244.42.66	TLSv1.2	1246 Application Data
58	0.715816	104.244.42.66	192.168.137.230	TCP	118 https(443) → 43011 [ACK] Seq=3520 Ack=1469 Win=33024 Len=0 TSval=246
61	0.785570	104.244.42.66	192.168.137.230	TLSv1.2	1714 Application Data, Application Data
68	0.826009	192.168.137.230	104.244.42.66	TCP	66 43011 → https(443) [ACK] Seq=1469 Ack=4318 Win=102144 Len=0 TSval=14
235	12.659996	192.168.137.230	104.244.42.66	TLSv1.2	1432 Application Data
238	12.811250	104.244.42.66	192.168.137.230	TCP	118 https(443) → 43011 [ACK] Seq=4318 Ack=2835 Win=35840 Len=0 TSval=246
247	12.950089	104.244.42.66	192.168.137.230	TLSv1.2	3014 Application Data
248	12.950275	104.244.42.66	192.168.137.230	TLSv1.2	2252 Application Data
249	12.952315	192.168.137.230	104.244.42.66	TCP	66 43011 → https(443) [ACK] Seq=2835 Ack=5766 Win=105088 Len=0 TSval=14
250	12.952791	192.168.137.230	104.244.42.66	TCP	66 43011 → https(443) [ACK] Seq=2835 Ack=6833 Win=107904 Len=0 TSval=14

Figure 41. Encrypted Data Patterns After Successful Authentication

- **Entering wrong password**

In the next step, we entered the wrong password for login and observed the traffic behaviour shown in Figure 42. From the figure, it can be observed that the client 192.168.137.104 established connection with 104.244.42.2 and sent SYN messages on TCP port 51336. After the TLS Handshake, the user is directed to the login page to enter the login credentials. In this case, we entered the wrong password and observed that the server sent the TCP Retransmission packet shown in packet 77. It was also observed that after receiving the error packet, the user was redirected to the login page.

Further analysis shows that after receiving TCP retransmission message, the client establishes connection with 104.244.42.67 and 104.244.42.129 as shown in Figure 43.

In this study, the traffic is encrypted therefore, the credential information cannot be retrieved. Nevertheless, we can identify the patterns in terms of packet length and timestamps.

## 5.6.2 Logout

Whenever a user logs out, it ends the HTTPS session with the server. While logging out, the client established the connection with three server IPs including 104.244.42.2, 104.244.42.193 and 104.244.42.65. During logout, it was observed that the client ended the session with 104.244.42.65 which shows that this server was used for login.

No.	Time	Source	Destination	Protocol	Length	Ethernet	Info
7	0.187348	192.168.137.104	104.244.42.2	TCP	74	✓	51336 → https(443) [SYN] Seq=
12	0.340573	104.244.42.2	192.168.137.104	TCP	134	✓	https(443) → 51336 [SYN, ACK]
13	0.341764	192.168.137.104	104.244.42.2	TCP	66	✓	51336 → https(443) [ACK] Seq=
16	0.489881	192.168.137.104	104.244.42.2	TLSv1.2	228	✓	Client Hello
20	0.643071	104.244.42.2	192.168.137.104	TCP	118	✓	https(443) → 51336 [ACK] Seq=
22	0.644600	104.244.42.2	192.168.137.104	TLSv1.2	3014	✓	Server Hello
23	0.646663	104.244.42.2	192.168.137.104	TCP	3014	✓	https(443) → 51336 [ACK] Seq=
24	0.647998	192.168.137.104	104.244.42.2	TCP	66	✓	51336 → https(443) [ACK] Seq=
25	0.648748	104.244.42.2	192.168.137.104	TLSv1.2	880	✓	Certificate, Server Key Excha
26	0.650010	192.168.137.104	104.244.42.2	TCP	66	✓	51336 → https(443) [ACK] Seq=
27	0.650227	192.168.137.104	104.244.42.2	TCP	66	✓	51336 → https(443) [ACK] Seq=
49	0.755383	192.168.137.104	104.244.42.2	TLSv1.2	192	✓	Client Key Exchange, Change C
52	0.910418	104.244.42.2	192.168.137.104	TLSv1.2	602	✓	New Session Ticket, Change C
55	0.912496	192.168.137.104	104.244.42.2	TCP	66	✓	51336 → https(443) [ACK] Seq=
59	0.933670	192.168.137.104	104.244.42.2	TLSv1.2	1212	✓	Application Data
67	1.125507	104.244.42.2	192.168.137.104	TCP	118	✓	https(443) → 51336 [ACK] Seq=
73	1.257322	104.244.42.2	192.168.137.104	TCP	3014	✓	https(443) → 51336 [ACK] Seq=
74	1.257722	104.244.42.2	192.168.137.104	TLSv1.2	808	✓	Application Data
77	1.546171	104.244.42.2	192.168.137.104	TCP	808	✓	[TCP Retransmission] https(443)
79	1.549922	192.168.137.104	104.244.42.2	TCP	66	✓	51336 → https(443) [ACK] Seq=

Figure 42. Patterns for Entering Wrong Password

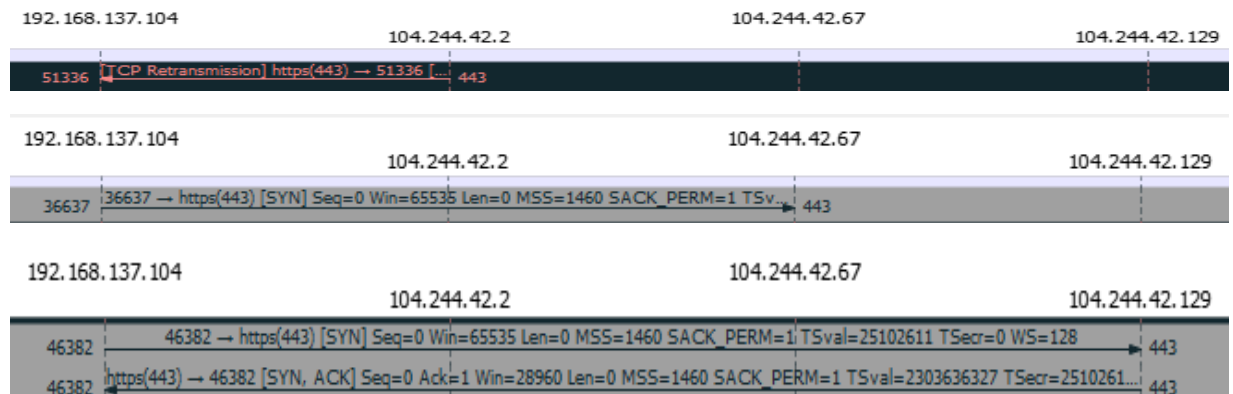


Figure 43. Connection Redirection to Alternate IPs after Entering Wrong Password

Figure 44 shows that the client 192.168.137.230 sent FIN, ACK message to 104.244.42.65 on two client ports, 60072 and 60071 to end the session. Server responds with ‘Encrypted Alert’ message and [FIN, ACK] packet on port 60071. It also sends an ACK message on port 60072. The flow graph of this activity is shown in Figure 45. Patterns for logout activity are shown in Table 15. Further, it must be noted that there is a difference in traffic patterns for logout activity and session termination by closing the application window. If a client is terminating the session by closing the application window, the connection will be disturbed and error packets will be received.

No.	Time	Source	Destination	Protocol	Length	Ethernet	Info
9	0.429585	192.168.137.230	104.244.42.65	TCP	66	✓	60072 → https(443) [FIN, ACK]
10	0.430100	192.168.137.230	104.244.42.65	TCP	66	✓	60071 → https(443) [FIN, ACK]
15	0.579672	104.244.42.65	192.168.137.230	TLSv1.2	180	✓	Encrypted Alert
16	0.579882	104.244.42.65	192.168.137.230	TCP	118	✓	https(443) → 60071 [FIN, ACK]
17	0.581686	104.244.42.65	192.168.137.230	TCP	118	✓	https(443) → 60072 [ACK] Seq=1

Figure 44. Traffic Patterns for Logout Activity

### 5.6.3 Opening an image

When a client opened a media tab, DNS queries were sent to local server for pbs.twimg.com and video.twimg.com as shown in Figure 46.

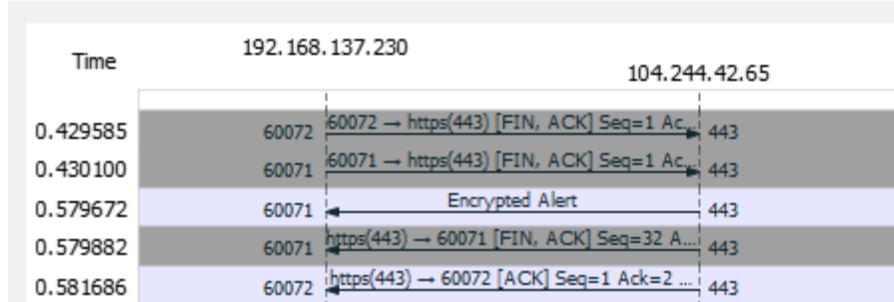


Figure 45. Flow Graph for Logout Activity

Table 15. Patterns for Logout Activity

Client IP	Server IP	Session	Packet Pattern	Packet Length
192.168.137.230	104.244.42.65	Client → Server	[FIN,ACK]	66
		Server → Client	Encrypted Alert	180
		Server → Client	[FIN, ACK]	118

3.439...	192.168.137.61	192.168.137.1	DNS	73 Standard query 0xb306 A pbs.twimg.com
3.441...	192.168.137.61	192.168.137.1	DNS	75 Standard query 0x3584 A video.twimg.com
3.482...	192.168.137.1	192.168.137.61	DNS	227 Standard query response 0xb306 A pbs.twimg.com CNAME cs
3.483...	192.168.137.1	192.168.137.61	DNS	229 Standard query response 0x3584 A video.twimg.com CNAME

Figure 46. DNS Queries for Media Server

Figure 46 shows that when a client accesses the media, it sends a query to pbs.twimg.com which is Twitter’s image hosting domain. Because the media includes images as well as videos, it also sends a query to video.twimg.com which is Twitter’s video hosting server. Through this domain, a client can upload and play a video [58].

It also sends a query to api-stream.twitter.com as shown in Figure 47. api-stream.twitter.com is accessed when a user wants real-time data. For example, if a user wants to receive tweets related to the keyword ‘hockey’, he/she will register the keyword hockey. As soon as Twitter receives the tweet related to hockey this tweet will be delivered to the user by streaming API. Its quality depends upon the criteria user sets such as username, keyword, location, places etc. [59] [60].

323 7.287...	192.168.137.61	192.168.137.1	DNS	82 Standard query 0x267e A api-stream.twitter.com
326 7.329...	192.168.137.1	192.168.137.61	DNS	118 Standard query response 0x267e A api-stream.twitter.com CNAME probe.twitter.com A 104.244.42.195

Figure 47. Query Sent to api-stream.twitter.com



192.168.137.61	93.184.220.70	TCP	74	43245 → https(443) [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
93.184.220.70	192.168.137.61	TCP	134	https(443) → 43245 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 S
192.168.137.61	93.184.220.70	TCP	66	43245 → https(443) [ACK] Seq=1 Ack=1 Win=87680 Len=0 TSval=35366207
192.168.137.61	93.184.220.70	TLSv1.2	583	Client Hello
93.184.220.70	192.168.137.61	TCP	118	https(443) → 43245 [ACK] Seq=1 Ack=518 Win=145920 Len=0 TSval=259314
93.184.220.70	192.168.137.61	TLSv1.2	3014	Server Hello
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=1449 Ack=518 Win=145920 Len=1448 TSval=
93.184.220.70	192.168.137.61	TLSv1.2	1950	Certificate, Certificate Status, Server Key Exchange, Server Hello D
192.168.137.61	93.184.220.70	TCP	66	43245 → https(443) [ACK] Seq=518 Ack=1449 Win=90496 Len=0 TSval=3536
192.168.137.61	93.184.220.70	TCP	66	43245 → https(443) [ACK] Seq=518 Ack=2897 Win=93440 Len=0 TSval=3536
192.168.137.61	93.184.220.70	TCP	66	43245 → https(443) [ACK] Seq=518 Ack=3813 Win=96384 Len=0 TSval=3536
192.168.137.61	93.184.220.70	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
93.184.220.70	192.168.137.61	TLSv1.2	602	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
192.168.137.61	93.184.220.70	TLSv1.2	937	Application Data
93.184.220.70	192.168.137.61	TLSv1.2	3014	Application Data

Figure 49. Connection Established with Twitter to Open an Image

93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=5503 Ack=1515 Win=147968 Len=1448 TSval=2593148705 TS
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=6951 Ack=1515 Win=147968 Len=1448 TSval=2593148705 TS
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=8399 Ack=1515 Win=147968 Len=1448 TSval=2593148705 TS
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=9847 Ack=1515 Win=147968 Len=1448 TSval=2593148705 TS
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=11295 Ack=1515 Win=147968 Len=1448 TSval=2593148705 T
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=12743 Ack=1515 Win=147968 Len=1448 TSval=2593148705 T
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=14191 Ack=1515 Win=147968 Len=1448 TSval=2593148705 T
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=15639 Ack=1515 Win=147968 Len=1448 TSval=2593148705 T
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=17087 Ack=1515 Win=147968 Len=1448 TSval=2593148705 T
93.184.220.70	192.168.137.61	TCP	3014	https(443) → 43245 [ACK] Seq=18535 Ack=1515 Win=147968 Len=1448 TSval=2593148705 T

Figure 48. Fixed Traffic Patterns While Image is Opened

From Figure 48, it was observed that when a client opens an image, a connection is established with 93.184.220.70 which is an IP address of pbs.twimg.com server. The figure shows that 192.168.137.61 performs TLS Handshake with pbs.twimg.com on TCP port 43245. After the handshake, the client sends encrypted data. While the image remains open, the server continuously sends TCP segments having a fixed packet length of 3014 bytes as shown in Figure 49. It was also observed that the packets had a fixed payload of 1448 bytes. The identified patterns for opening an image (length of 3014 bytes) is shown in Table 16. According to [61], 93.184.220.70 is bound for domain resolution of pbs.twimg.com. The autonomous number for 93.184.220.70 is AS15133 MCI Communications Services Inc. d/b/a Verizon Business. MCI Communication services provides the wireless long distance Information Technology IT and Telecommunication services [62].

Table 16. Patterns for Opening an Image

Client IP	Server IP	Session	Identified Patterns	Packet Length
192.168.137.61	93.184.220.70	Client → Sever	[SYN]	74
		Server → Client	[SYN, ACK]	134
		Server → Client	TCP segments with 1448 bytes payload	3014
		Client → Server	[ACK]	66

### 5.6.4 Sending a text message

When a client sent a text message, certain fixed patterns were created. We analysed the patterns offline from multiple captured files. The results of this activity are shown in Figure 50, 51 and 52.

192.168.137.125	104.244.42.3	TCP	1514 55807 → https(443) [ACK] Seq=1 Ack=
192.168.137.125	104.244.42.3	TLSv1.2	198 Application Data
104.244.42.3	192.168.137.125	TCP	118 https(443) → 55807 [ACK] Seq=1 Ack=
104.244.42.3	192.168.137.125	TCP	118 https(443) → 55807 [ACK] Seq=1 Ack=
104.244.42.3	192.168.137.125	TLSv1.2	1696 Application Data, Application Data
192.168.137.125	104.244.42.3	TCP	66 55807 → https(443) [ACK] Seq=1581 ,
192.168.137.125	104.244.42.3	TCP	1514 55807 → https(443) [ACK] Seq=1581 ,
192.168.137.125	104.244.42.3	TLSv1.2	264 Application Data
104.244.42.3	192.168.137.125	TCP	118 https(443) → 55807 [ACK] Seq=790 A
104.244.42.3	192.168.137.125	TCP	118 https(443) → 55807 [ACK] Seq=790 A
104.244.42.3	192.168.137.125	TLSv1.2	1696 Application Data, Application Data
192.168.137.125	104.244.42.3	TCP	66 55807 → https(443) [ACK] Seq=3227 ,

Figure 50. Fixed patterns on TCP port 55807 (Observation-I)

192.168.137.125	104.244.42.3	TLSv1.2	1494 Application Data
104.244.42.3	192.168.137.125	TCP	118 https(443) → 55807
104.244.42.3	192.168.137.125	TLSv1.2	2872 Application Data
192.168.137.125	104.244.42.3	TCP	66 55807 → https(443)
192.168.137.125	104.244.42.3	TLSv1.2	1494 Application Data
104.244.42.3	192.168.137.125	TCP	118 https(443) → 55807
104.244.42.3	192.168.137.125	TLSv1.2	2872 Application Data
192.168.137.125	104.244.42.3	TCP	66 55807 → https(443)
192.168.137.125	104.244.42.3	TLSv1.2	1492 Application Data
104.244.42.3	192.168.137.125	TCP	118 https(443) → 55807

Figure 51. Fixed patterns on TCP port 55807 (observation-II)

104.244.42.3	192.168.137.125	TLSv1.2	320 Application Data, Application Data, Application Data
192.168.137.125	104.244.42.3	TCP	66 49868 → https(443) [ACK] Seq=1 Ack=102 Win=866 Len=6
104.244.42.3	192.168.137.125	TLSv1.2	320 Application Data, Application Data, Application Data
192.168.137.125	104.244.42.3	TCP	66 49868 → https(443) [ACK] Seq=1 Ack=203 Win=866 Len=6
104.244.42.3	192.168.137.125	TLSv1.2	320 Application Data, Application Data, Application Data
192.168.137.125	104.244.42.3	TCP	66 49868 → https(443) [ACK] Seq=1 Ack=304 Win=866 Len=6
104.244.42.3	192.168.137.125	TLSv1.2	320 Application Data, Application Data, Application Data
192.168.137.125	104.244.42.3	TCP	66 49868 → https(443) [ACK] Seq=1 Ack=405 Win=866 Len=6

Figure 52. Fixed patterns on TCP port 49868

From the details, it was observed that when a client 192.168.137.125 sent a direct message, it established a connection with 104.244.42.3. Fixed patterns were created on TCP ports 55807 and 49868. On port 55807, client sends TCP segments with 1514 bytes packet length and encrypted Application Data with packet length 198 bytes as shown in Figure 50. In response, the server sends an ACK packet with length 118 bytes and encrypted Application Data having packet length 1696 bytes and this pattern repeated for the rest of the activity.



On the same TCP port, the traffic was observed with different patterns shown in Figure 51. The client sent a 1494 bytes encrypted packet. In response, the server sends an 118 bytes ACK packet. It also sends 2872 bytes encrypted packet and the patterns continued for this activity. Further, it was also observed that these patterns were the same for TCP port 45212.

For TCP port 49868, it was observed that the server 104.244.42.3 sent 320 bytes encrypted Application Data shown in Figure 52. In response, the client sent 66 bytes ACK packet and the patterns were the same for the rest of the activity. The identified patterns for sending a text message are shown in Table 17.

Table 17. Patterns for Sending a Text Message

Client IP	Server IP	Session	Packet Patterns	Packet Length
192.168.137.125	104.244.42.3	On port 55807 (Observation-I)		
		Client → Server	TCP Segments	1514
		Client → Server	Data Packet	198
		Server → Client	ACK	118
		Server → Client	Data Packet	1696
		On port 55807 (Observation-II)		
		Client → Server	Data Packet	1494
		Server → Client	ACK	118
		Server → Client	Data Packet	2872
		Client → Server	ACK	66
		On port 49868		
		Server → Client	Data Packet	320
		Client → Server	ACK	66

### 5.6.5 Sending media as a message

Initially, when the client accesses the application, it establishes a connection with two servers 104.244.42.2 on TCP port 38752 and 104.244.42.3 on TCP port 45212. When a client attaches and sends a media, the traffic is directed to 104.244.42.75 (upload.twitter.com) and 104.244.42.148 (ton.twitter.com) as shown in Figure 53.

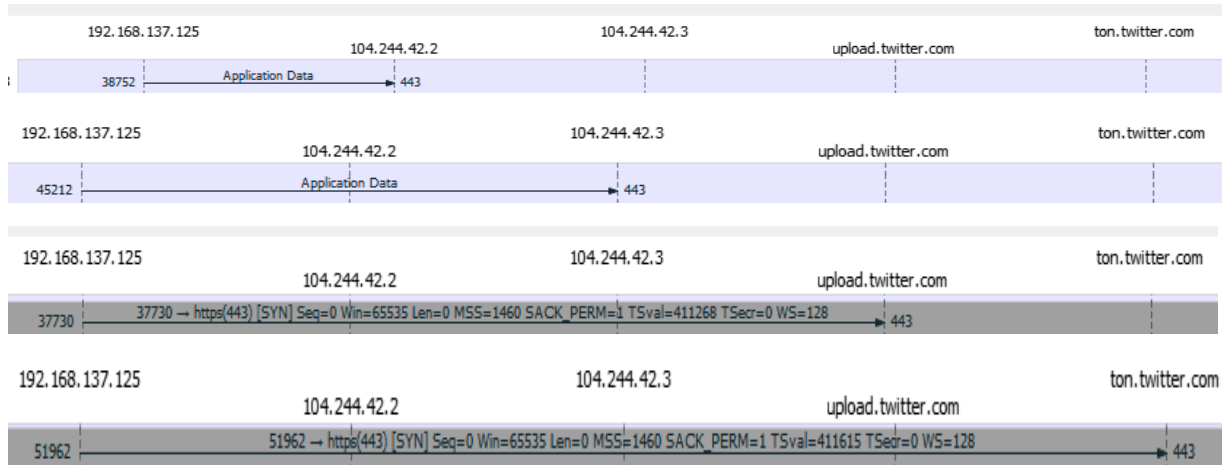


Figure 53. Traffic Flow Graph for Attaching and Sending a Media in Messages

- **upload.twitter.com:** When a Twitter user uploads a media in a tweet or attaches the media in Direct Message, the POST media/Upload and POST direct\_messages/events/new endpoints are used. Media can be any image (.jpeg, .jpg), any GIF or video. Further, uploading a media in tweets and attaching a media in direct messages have the same process. The domain for these endpoints is upload.twitter.com [63] [64] [65] [66].
- **ton.twitter.com:** ton.twitter.com is the media URL when a Twitter client accesses/sends a media in direct messages. The APIA starts with ton and it represents that the image or video is accessed or sent via DMs. It is similar to the URL twimg.twitter.com which is the media URL when accessed in tweets [67].

During analysis, it was observed that when a client uploads an image, a connection is established with upload.twitter.com as shown in figure 54. While uploading the attachment, continuous packets with fixed length were sent from client 192.168.137.125.

192.168.137.125	upload.twitter.com	TCP	74 37730 → https(443) [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
upload.twitter.com	192.168.137.125	TCP	134 https(443) → 37730 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA
192.168.137.125	upload.twitter.com	TCP	66 37730 → https(443) [ACK] Seq=1 Ack=1 Win=87680 Len=0 TSval=411283 TSe
192.168.137.125	upload.twitter.com	TLSv1.2	231 Client Hello
upload.twitter.com	192.168.137.125	TCP	118 https(443) → 37730 [ACK] Seq=1 Ack=166 Win=30208 Len=0 TSval=19203301
upload.twitter.com	192.168.137.125	TLSv1.2	3014 Server Hello
upload.twitter.com	192.168.137.125	TCP	3014 https(443) → 37730 [ACK] Seq=1449 Ack=166 Win=30208 Len=1448 TSval=19
upload.twitter.com	192.168.137.125	TLSv1.2	894 Certificate, Server Key Exchange, Server Hello Done
192.168.137.125	upload.twitter.com	TCP	66 37730 → https(443) [ACK] Seq=166 Ack=1449 Win=90496 Len=0 TSval=41130
192.168.137.125	upload.twitter.com	TCP	66 37730 → https(443) [ACK] Seq=166 Ack=2897 Win=93440 Len=0 TSval=41130
192.168.137.125	upload.twitter.com	TCP	66 37730 → https(443) [ACK] Seq=166 Ack=3285 Win=96384 Len=0 TSval=41130
192.168.137.125	upload.twitter.com	TLSv1.2	192 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
upload.twitter.com	192.168.137.125	TLSv1.2	602 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

Figure 54. Connection Established with upload.twitter.com

Figure 54 shows that the client establishes a connection with upload.twitter.com and performs the TLS Handshake. After the handshake, uploading activity is started where a stream of packets is sent from client

to server until the attachment is completed. When a client uploads an image, a stream of packets with fixed length of 1514 bytes and payload size of 1448 bytes is sent from client to server as shown in Figure 55.

While it was the opposite from ton.twitter.com, in this case, when a Twitter client sends the media in a direct message, URL ton.twitter.com is accessed. A stream of packets with fixed length of 3014 and 1448 bytes payload size is sent from 104.244.42.148 (ton.twitter.com) to the client. This stream of packets was so frequent that the client responded with an acknowledgment packet once the stream was stopped as shown in Figure 56. Table 18 shows the patterns created while sending a media as a message.

### 5.6.6 Playing a video

For the next step, we played several videos and analysed the traffic behaviour. From the analysis, it was observed that when a user played a video, traffic was directed to 68.232.34.217 which belongs to Verizon services. It concludes that this server provides video services to Twitter Inc.

192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=20610	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=22058	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TLSv1.2	1514	Application Data, Application Data								
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=24954	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=26402	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=27850	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=29298	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TLSv1.2	1514	Application Data [TCP segment of a reassembled PDU]								
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=32194	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=33642	Ack=4561	Win=102144	Len=1448	TSval=411400
192.168.137.125	upload.twitter.com	TLSv1.2	1514	Application Data [TCP segment of a reassembled PDU]								
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=36538	Ack=4561	Win=102144	Len=1448	TSval=411401
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=37986	Ack=4561	Win=102144	Len=1448	TSval=411401
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=39434	Ack=4561	Win=102144	Len=1448	TSval=411401
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=40882	Ack=4561	Win=102144	Len=1448	TSval=411401
192.168.137.125	upload.twitter.com	TCP	1514	37730	→	https(443)	[ACK]	Seq=42330	Ack=4561	Win=102144	Len=1448	TSval=411401
upload.twitter.com	192.168.137.125	TCP	118	https(443)	→	37730	[ACK]	Seq=4561	Ack=16266	Win=61952	Len=0	TSval=1920331270

Figure 55. Patterns While Uploading a Photo in Messages

ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=18021	Ack=1649	Win=33024	Len=1448	TSval=729351796
ton.twitter.com	192.168.137.125	TLSv1.2	3014	Application Data, Application Data								
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=20917	Ack=1649	Win=33024	Len=1448	TSval=729351796
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=22365	Ack=1649	Win=33024	Len=1448	TSval=729351796
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=23813	Ack=1649	Win=33024	Len=1448	TSval=729351799
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=25261	Ack=1649	Win=33024	Len=1448	TSval=729351799
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=26709	Ack=1649	Win=33024	Len=1448	TSval=729351799
192.168.137.125	ton.twitter.com	TCP	66	51962	→	https(443)	[ACK]	Seq=1649	Ack=20917	Win=134016	Len=0	TSval=411711
ton.twitter.com	192.168.137.125	TLSv1.2	3014	Application Data [TCP segment of a reassembled PDU]								
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=29605	Ack=1649	Win=33024	Len=1448	TSval=729351807
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=31053	Ack=1649	Win=33024	Len=1448	TSval=729351807
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=32501	Ack=1649	Win=33024	Len=1448	TSval=729351807
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=33949	Ack=1649	Win=33024	Len=1448	TSval=729351807
ton.twitter.com	192.168.137.125	TLSv1.2	3014	Application Data [TCP segment of a reassembled PDU]								
ton.twitter.com	192.168.137.125	TLSv1.2	3014	Application Data [TCP segment of a reassembled PDU]								
ton.twitter.com	192.168.137.125	TCP	3014	https(443)	→	51962	[ACK]	Seq=38293	Ack=1649	Win=33024	Len=1448	TSval=729351807
192.168.137.125	ton.twitter.com	TCP	66	51962	→	https(443)	[ACK]	Seq=1649	Ack=23813	Win=139776	Len=0	TSval=411712

Figure 56. Patterns While Sending a Photo in Messages

Table 18. Patterns for Sending a Media

Client IP	Server IP	Session	Identified Patterns	Packet Length
<b>While Attaching an Image</b>				
192.168.137.125	104.244.42.75 (upload.twitter.com)	Client → Server	SYN	74
		Server → Client	SYN, ACK	134
		Client → Server	TCP Segments with 1448 payload size	1514
<b>While Sending an Image</b>				
192.168.137.125	104.244.42.148 (ton.twitter.com)	Client → Server	SYN	74
		Server → Client	SYN, ACK	134
		Server → Client	TCP Segments with 1448 payload size	3014

Figure 57 shows that when a user starts playing a video, a stream of packets with length 2814 bytes and payload size 1360 bytes is sent from server 68.232.34.217 to client 192.168.137.201. Moreover, when a client pauses or stops a video, the stream of packets is stopped and random behaviour is observed. Table 19 shows the patterns created while playing a video.

Time	Source	Destination	Protocol	Length	Info
2020-03-28 18:35:34.712591	68.232.34.217	192.168.137.201	TCP	2814	[TCP Retransmission] https(443) → 39740 [ACK] Seq=222336 Ack-
2020-03-28 18:35:34.718280	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=225056 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.723702	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=226416 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.729107	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=227776 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.734785	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=229136 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.740486	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=230496 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.745828	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=231856 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.751534	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=233216 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.756947	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=234576 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.762372	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=235936 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.767937	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=237296 Ack=640 Win=296 Len=1360
2020-03-28 18:35:34.773414	68.232.34.217	192.168.137.201	TCP	2814	https(443) → 39740 [ACK] Seq=238656 Ack=640 Win=296 Len=1360

Figure 57. Fixed Patterns for Playing a Video

Table 19. Patterns for Playing a Video

Client IP	Server IP	Session	Identified Patterns	Packet Length
192.168.137.201	68.232.34.217	Client → Server	SYN	74
		Server → Client	SYN, ACK	134
		Server → Client	TCP Segments with 1360 payload size	2814

## 5.7 Results

Python scripting language was used to read the Wireshark traces in order to verify the findings and observed patterns for various activities on Twitter. We created a script for user related activities based on our findings. To verify the findings, we performed four Twitter activities, namely, login, opening an image, playing a video and logout and captured the traffic. The findings were verified by running a script on random packet captures.

From the captured files, we detected patterns of user related activities on Twitter. It was observed that our script correctly identified all four activities. The results in Table 20 show 0 false positives, 0 false negatives, 100% true positives and 100% true negatives.

Table 20. Results

Activity	Detected Patterns	False Positives	True Positives	False Negative	True Negative
Login	Yes	0	100%	0	100%
Opening an Image	Yes	0	100%	0	100%
Playing a video	Yes	0	100%	0	100%
Logout	Yes	0	100%	0	100%

## 5.8 Application of the research

Packet analysis plays an important role in backtracking and putting the real image of a crime. When the traffic is encrypted, there are two ways to get the information. One is to get the decryption key and second is to break the encryption [68]. In case of a secure application server, getting a decryption key is near to impossible for small to medium enterprises. Further, due to strong encryption, it is difficult to break the encryption and extract the information. Therefore, the only solution is to perform a volume based analysis, statistical packet analysis or pattern matching analysis on encrypted traffic and extract the important artefacts to present in the court of law.

If the organization keeps the backup of traffic, it can be fetched in case of any criminal activity. With the help of traffic identification, source and destination IP addresses, source and destination ports, timestamps and volume of the traffic a forensic investigator can create a sequence of evidence. The sequence of evidence will help in summarizing the report of the crime act. In this section, few cases which are mapped to this study are highlighted to show how this research can help in the forensics investigations.

- **Blocking Twitter server in campus premises for the specific time i.e. exams or during the outbreak of false rumours**

When a user accesses twitter.com, the DNS responds with multiple server IP addresses belonging to twitter.com to establish connection. In this case, if the administration takes an action to block the Twitter server, they will need to have knowledge of complete infrastructure of twitter.com. As Twitter has multiple networks to provide services to customers, it is very important to have the knowledge of all the networks. This research provides the complete picture of all the networks used by twitter.com to provide services in this region.

- **Identification of source and confirmation of data leakage on Twitter through behaviour analysis**

Our analysis is based on multiple parameters such as timeframe, destination IP addresses (unicast or broadcast). This research provides the complete flow of the behaviour of a user on Twitter application for different activities. One of the cases of sending images on Twitter is described below. The objective is to identify the source of leakage and confirm the activity.

1. Identification of the source and destination IP addresses: The identification of source IP address can be done through the timeframe. For example, if the sensitive photographs were leaked and received by Twitter users between 15:49 PM to 15:52 PM exact.

A forensic investigator will find the IP addresses communicated on Twitter between this timeframe. Figure 58 shows that client 192.168.0.3 communicated with Twitter server 104.244.42.2 at 15:49.54 and terminated the connection at 15:51:46 on TCP port 51414. It was also observed that during this time, the client again sent SYN messages to the same IP address on TCP port 45438 at 15:50:7.

2. In the next step, the investigator will analyse the behaviour of traffic in that duration and confirm the leakage. Figure 59 shows the stream of packets sent from 192.168.0.3. In our research, we described that these patterns are created when a user uploads the media in direct messages on Twitter. From the figure below, it can be observed that when the photographs were uploaded on

Time	Source	Destination	Protocol	Length	Info
2020-03-11 15:49:54.522297	192.168.0.3	104.244.42.2	TLSv1.2	1339	Application Data
2020-03-11 15:49:54.553296	192.168.0.3	104.244.42.2	TLSv1.2	352	Application Data
2020-03-11 15:50:07.205126	192.168.0.3	104.244.42.2	TCP	74	45438 → https(443) [SYN] Seq=0
2020-03-11 15:50:07.361081	104.244.42.2	192.168.0.3	TCP	74	https(443) → 45438 [SYN, ACK]
2020-03-11 15:51:46.722672	104.244.42.2	192.168.0.3	TCP	66	https(443) → 51414 [FIN, ACK]
Time	Source	Destination	Protocol	Length	Info
2020-03-11 15:50:32.452914	192.168.0.3	d3gzx3srebfdro.clo...	TCP	74	55971 → https(443) [SYN] Seq=0
2020-03-11 15:50:32.494342	d3gzx3srebfdro.clo...	192.168.0.3	TCP	74	https(443) → 55971 [SYN, ACK]
2020-03-11 15:50:33.015139	d3gzx3srebfdro.clo...	192.168.0.3	TLSv1.2	1290	Application Data
2020-03-11 15:50:33.059906	192.168.0.3	d3gzx3srebfdro.clo...	TCP	66	55971 → https(443) [ACK] Seq=1575 Ack=5342 Win=102144 Len=0 TSva
2020-03-11 15:50:33.109281	192.168.0.3	d3gzx3srebfdro.clo...	TLSv1.2	1465	Application Data
2020-03-11 15:50:33.111281	192.168.0.3	d3gzx3srebfdro.clo...	TCP	1514	55971 → https(443) [ACK] Seq=2974 Ack=5342 Win=102144 Len=1448 T
2020-03-11 15:50:33.113156	192.168.0.3	d3gzx3srebfdro.clo...	TCP	1514	55971 → https(443) [ACK] Seq=4422 Ack=5342 Win=102144 Len=1448 T
2020-03-11 15:50:33.115031	192.168.0.3	d3gzx3srebfdro.clo...	TCP	1514	55971 → https(443) [ACK] Seq=5870 Ack=5342 Win=102144 Len=1448 T
2020-03-11 15:50:33.116906	192.168.0.3	d3gzx3srebfdro.clo...	TCP	1514	55971 → https(443) [ACK] Seq=7318 Ack=5342 Win=102144 Len=1448 T
2020-03-11 15:50:33.118780	192.168.0.3	d3gzx3srebfdro.clo...	TCP	1514	55971 → https(443) [ACK] Seq=8766 Ack=5342 Win=102144 Len=1448 T
2020-03-11 15:50:33.120655	192.168.0.3	d3gzx3srebfdro.clo...	TLSv1.2	1514	Application Data [TCP segment of a reassembled PDU]
2020-03-11 15:50:33.122530	192.168.0.3	d3gzx3srebfdro.clo...	TCP	1514	55971 → https(443) [ACK] Seq=11662 Ack=5342 Win=102144 Len=1448
2020-03-11 15:50:33.124405	192.168.0.3	d3gzx3srebfdro.clo...	TCP	1514	55971 → https(443) [ACK] Seq=13110 Ack=5342 Win=102144 Len=1448
2020-03-11 15:50:33.150002	d3gzx3srebfdro.clo...	192.168.0.3	TCP	66	https(443) → 55971 [ACK] Seq=5342 Ack=2974 Win=35840 Len=0 TSval
2020-03-11 15:51:25.278944	d3gzx3srebfdro.clo...	192.168.0.3	TLSv1.2	642	Application Data
2020-03-11 15:51:25.401832	192.168.0.3	d3gzx3srebfdro.clo...	TCP	66	55971 → https(443) [ACK]

Figure 58. Observation of Timeframe for Stream of Packets

Twitter the source IP address communicated with was 143.204.106.11, which provides media services to twitter.com.

From the figures, it is observed that the leakage was done from 15:50:32 when a SYN packet was sent to 132.104.106.11 to establish connection and SYN, ACK packet was received. It was noticed that the stream of packets was sent between 15:50:33 PM to 15:51:25 PM and after 23 seconds, the connection was terminated with Twitter.

Again, in this step, a working Hypothesis will be created based on educated guesses of likelihood of activities and a timeline will be formed. The timeline is strongly supported by evidence, references and timeframe.

**15:49:54 PM, 2020-03-11**--The flow capture begins with Twitter application at port 51414.

**15:50:07 PM, 2020-03-11**--Another connection established with the same IP address on TCP 45438.

**15:50:32 PM, 2020-03-11**--the connection established with 143.204.106.11.

**15:50:33 PM, 2020-03-11**--The stream of packets started from the source address.

**15:51:25 PM, 2020-03-11**--The continuous flow of packets stopped.

**15:51:46 PM, 2020-03-11**--The connection ended with Twitter.

Now, to identify the host of the source IP address, the investigator can check the DHCP lease in the organization's firewall which shows the complete list of IP addresses.

- **Blocking the image or video servers of Twitter to avoid blasphemous and inappropriate content**

In order to block the specific server, we need to have the information about all the media service providers of Twitter. This study gives the complete picture of all the media servers accessed by Twitter.

Because, twitter.com plays an important part in live news updates all over the world, it will not be in favour of being blocked by organizations in a certain region. Therefore, in blasphemy cases, the possible solution is to block the media servers. For example, if a country/organization/educational institute wants to block the image server of a social media application, it is important to have a complete knowledge of image service providers.

- **Individual profiling to eavesdrop on activities of an individual**

- Normal working hours

- Lunch time
- Break time
- Sources of entertainment

## **5.9 Summary**

In this chapter, network forensic analysis of Twitter application was implemented. We followed the framework to implement the behaviour analysis and server identification for multiple activities. In the end, case studies were defined to highlight the significance of this study.



# Chapter 6

## CONCLUSION AND FUTURE SCOPE

### 6.1 Introduction

This chapter concludes the thesis and discusses the future aspect in detail. Future scope defines how this work can be extended using different platforms, tools and technologies. Discussed sections are as follows:

- **Section 6.2** Conclusion
- **Section 6.3** Future Scope

### 6.2 Conclusion

With the advancement in computing power and technology, social media application engagement has increased resulting in a higher crime rate. Therefore, most of the applications have been designed with strong security features. However, during criminal activities, a cybercriminal leaves a significant amount of data in a device or on a network. In the network forensics context, complex security architecture of apps makes it difficult for forensic investigators to find the possible artefacts.

In this thesis, we performed the network forensic analysis of Twitter application on an Android smartphone. Initially, the architecture of the application was unknown and analysis was implemented on encrypted traffic. We explored the application for different features, performed activities and carried forensic analysis to find possible remnants against those activities. For behaviour analysis, traffic classification technique was applied based on different packet level attributes. Moreover, to discover the development of Twitter architecture, a firewall was used. The unique idea of firewall deployment helped in the identification of alternate connectivity options of Twitter application. To conduct the behaviour analysis, we performed different user actions and classified the traffic to find the characteristics of Twitter traffic. The traffic was classified on port based analysis, IP address identification, Packet length, Payload size, Timestamps and fixed pattern analysis.

Following the proposed methodology, we discovered various artefacts against different user activities. Source IP address revealed the identity of Twitter application and helped in the identification of particular server. Fixed packet length and payload size helped in the behaviour analysis of user activities. Further, tracing timestamps showed the duration of event occurrence. During forensic analysis, it was discovered that Twitter application uses different ISPs for its services. We showed that, for image services, Twitter traffic is redirected to MCI communication servers which is owned by Verizon Communications. Similarly, Amazon and Google servers were frequently used for redirecting traffic while performing different

activities. This experiment has discovered the complete architecture of Twitter application which has not been provided in any research work yet.

Since, the analysis of encrypted traffic is a difficult task, we mapped our study to different use cases and showed how it would be helpful in analysing encrypted traffic. At a large scale, this study can be used in serious crime investigations involving this application and in business intelligence solutions. Along with this, our methodology can be adopted to carry out network forensics of any secure application which has a large, distributed and secure architecture.

### **6.3 Future Scope**

The findings of this thesis provide new insights to researchers for conducting the network forensics of secure social media applications on different platforms. In the forensic study of IMO application [30], network traffic analysis of an IMO app was performed on Android and iOS which showed some major differences. Therefore, in future, our research can be extended for traffic analysis on different platforms such as Windows, MAC and iOS. Further, the features on social media applications are updated according to user interests therefore, it is necessary to profile the Twitter application for new features.

For our research, we used Wireshark, however, multiple network monitoring tools such as TCPFlow, Flow-tools, NFDumps, PADS Argus, Nessus Sebek and TCPDump can be used for traffic monitoring and analysis.

## REFERENCES

- [1] Verma, "CYBER FORENSIC SCIENCE," Cyberlekh Pub., Oct. 2018.
- [2] R. Kaur and A. Kaur, "Digital Forensics," *Int. J. of Comp. Apps.*, vol. 50, no. 5, pp. 0975-8887, Jul. 2012.
- [3] B. Carrier, "Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers," *Int. J. of Digit. Evid.*, vol. 1, no. 4, pp. 1-12, Jan. 2003.
- [4] R. Rowlingson, "A Ten Step Process for Forensic Readiness," *Int. J. of Digit. Evid.*, vol. 2, no. 3, pp. 1-28, 2004.
- [5] H. Arshad, A. B. Jantan, and O. I. Abiodun, "Digital Forensics: Review of Issues in Scientific Validation of Digital Evidence," *J. of Info. Process. Syst.*, vol. 14, no. 2, pp. 346-376 Apr. 2018.
- [6] S. Saleem, "Protecting the Integrity of Digital Evidence and Basic Human Rights During the Process of Digital Forensics" PhD diss., Dpt. of Comput. Sci., 2015.
- [7] N. Kumari and A. K. Mohapatra, "An insight into digital forensics branches and tools," in *Proc. Int. Conf. on Comput. Techn. in Info. and Commun. Technol. (ICCTICT)*, New Delhi, 2016, pp. 243-250, doi: 10.1109/ICCTICT.2016.7514586
- [8] R. Ayers, S. Brothers, and W. Jansen. "Guidelines on mobile device forensics." NIST Spec. Pub., vol. 1, no. 1, pp. 85, 2014.
- [9] R. Ahmed, and R. V. Dharaskar. "Mobile forensics: an overview, tools, future trends and challenges from law enforcement perspective," in *Proc. 6th Int. Conf. on E-Govern., ICEG, Emerg. Technol. in E-Gov., M-Gov., India, 2008*, pp. 312-23.
- [10] K. Ruan, J. Carthy, T. Kechadi, and I. Baggili. "Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results." *Digit. Invest.*, vol. 10, no. 1, pp. 34-43, Jun. 2013.
- [11] V. Corey, C. Peterman, S. Shearin, M. S. Greenberg and J. Van Bokkelen, "Network forensics analysis," in *IEEE Internet. Comput.*, vol. 6, no. 6, pp. 60-66, Nov.-Dec. 2002, doi: 10.1109/MIC.2002.1067738.
- [12] E. Casey, "Network traffic as a source of evidence: tool strengths weaknesses, and future needs," *Digit. Invest.*, vol. 1, no. 1, pp. 28-43, Feb. 2004.
- [13] M. Hikmatyar, Y. Prayudi, and I. Riadi. "Network Forensics Framework Development using Interactive Planning Approach," in *Proc. Int. J. of Comput. Appl.*, vol. 161, no. 10, pp. 41-48, Mar. 2017.
- [14] "Introduction to Network Forensics", in *The European Union Agency for Cybersecurity (ENISA)*, version 1.1, 2019.
- [15] E. S. Pilli, R. C. Joshi and R. Niyogi, "A Generic Framework for Network Forensics," *Int. J. Comput. Appl.*, vol. 1, no. 11, pp. 0975-8887, 2010.

- [16] S. Davidoff and J. Ham, "Practical Investigative Strategies," in Network Forensics Tracking Hackers through Cyberspace. Westford, Massachusetts, USA: Prentice hall: Upper Saddle River, vol. 2014, 2012.
- [17] "Twitter," Wikipedia, [Online]. Available: <https://en.wikipedia.org/wiki/twitter>. [Accessed 8 December 2019].
- [18] "What is Twitter and Why should you use it?", UKRI, [Online]. Available: <https://esrc.ukri.org/research/impact-toolkit/social-media/twitter/what-is-twitter/>. [Accessed 8 December 2019].
- [19] S. Aslam, "Twitter by the Numbers: Stats, Demographics & Fun Facts," OMNICORE, Sep. 5, 2019. [Online]. Available: <https://www.omnicoreagency.com/twitter-statistics/>
- [20] "Number of monthly active Twitter users worldwide from 1st quarter 2010 to 1st quarter 2019," Statista, [Online]. Available: <https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>. [Accessed 8 December 2019].
- [21] M. Ahlgren, "50+ TWITTER STATISTICS & FACTS FOR 2020", Website Hosting Ratings, Jul. 5, 2020. [Online]. Available: <https://www.websitehostingrating.com/twitter-statistics/>
- [22] C. Smith, "400 Twitter Statistics and Facts (2020) | By the Numbers," Digital Marketing Ramblings, Nov. 20, 2015. [Online]. Available: <https://expandedramblings.com/index.php/twitter-stats-facts/>.
- [23] G. Udani, "An Exhaustive Study of Twitter Users Across the World," beevolve, Oct. 10, 2012. [Online]. <http://www.beevolve.com/twitter-statistics/>
- [24] A. Trotter-Press Association, "Social media-related crime reports up 780% in four years," The Guardian, Dec. 27, 2012. [Online]. Available: <https://www.theguardian.com/media/2012/dec/27/social-media-crime-facebook-twitter#comments>.
- [25] ElevenPaths, "Investigation report "Twitter Data Leakage"" Eleven Paths, Jun. 29, 2016. [Online]. Available: <https://www.elevenpaths.com/investigation-report-twitter-data-leakage/index.html>.
- [26] R. P. Curiel, S. Cresci, C. I. Muntean and S. R. Bishop "Crime and its fear in social media," Palgrave Comm., vol. 6, Apr. 2020, Art. no. 57.
- [27] M. A. K. Sudozai and S. Saleem, "Profiling of secure chat and calling apps from encrypted traffic," in Proc. 15th IEEE IBCAST, Islamabad, Pakistan, 2018, pp. 502–508.
- [28] V. Ndatinya, Z. Xiao, V. R. Manepalli, K. Meng and Y. Xiao "Network forensics analysis using Wireshark," Int. J. Secur. Netw., vol. 10, no. 2, pp. 91-106 Jul. 2015, DOI 10.1504/IJSN.2015.070421.
- [29] P. Velan, M. Cermak, P. Celeda and M. Drasar, "A survey of methods for encrypted traffic classification and analysis," Int. J. Netw. Mgmt., vol. 25, no. 5, pp. 355-374, Jul. 2015.

- [30] M.A.K Sudozai, S. Saleem, W. J. Buchanan, N. Habib, H. Zia, "Forensics study of IMO call and chat app," *Digit. Invest.*, vol. 25, pp. 5-23, June 2018.
- [31] D. Walnycky, I. Baggili, A. Marrington, J. Moore and F. Breitingner, "Network and device forensic analysis of Android social-messaging applications," *Digit. Invest.*, vol. 14, pp. SS7–S84, Aug. 2015.
- [32] F. Karpisek, I. Baggili and F. Breitingner, "WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages," *Digit. Invest.*, vol. 15, pp. 110–118, Oct. 2015.
- [33] 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 *Proc. 15th ACM CODASPY*, New York, NY, USA, 2015, pp. 297–304.
- [34] M. N. Yusoff, A. Dehghantanha and R. Mahmood, "Network traffic forensics on firefox mobile OS facebook, Twitter and telegram as case studies," *Contemp. Digit. For. Invest. Cld. Mob. App.*, pp. 63–78, Jan. 2017, DOI 10.1016/B978-0-12-805303-4.00005-8.
- [35] J. Muehlstein, Y. Zion, M. Bahumi, I. Kirshenboim, R. Dubin, A. Dvir and O. Pele, "Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application," in *Proc. 14th IEEE CCNC*, 2017, pp. 1–6
- [36] R. Marik, P. Bezpalec, J. Kucerak and L. Kencl, "Revealing viber communication patterns to assess protocol vulnerability," in *Proc. CoCoNet*, Trivandrum, India, 2015, pp. 496–504.
- [37] S. Molnar and M. Perenyi, "On the identification and analysis of skype traffic," *Int. J. Comm. Sys.*, vol. 24, no. 1, pp. 94–117, Jan. 2011
- [38] S. Wu, Y. Zhang, X. Wang, X. Xiong and L. Du, "Forensic analysis of WeChat on Android smartphones," *Digit. Invest.* vol. 21, pp. 3–10, Jan. 2017.
- [39] T. Mehrotra and B. M. Mehtre, "Forensic analysis of Wickr application on android devices," 2013 *IEEE International Conference on Computational Intelligence and Computing Research*, Enathi, 2013, pp. 1-6, doi: 10.1109/ICCIC.2013.6724230.
- [40] C. Anglano, "Forensic analysis of WhatsApp Messenger on Android smartphones," *Digit. Invest.* vol. 11, pp. 201–213, May. 2014.
- [41] C. Anglano, "Forensic analysis of the ChatSecure instant messaging application on android smartphones," *Digit. Invest.* vol. 19, pp. 44–59, Oct. 2016.
- [42] C. Anglano, "Forensic analysis of Telegram Messenger on Android smartphones," *Digit. Invest.* vol. 23, pp. 31–49, Sep. 2017.
- [43] M. I. Husain and R. Sridhar, "iForensics: Forensic Analysis of Instant Messaging on Smart Phones," in *Proc. ICDF2C*, Berlin, Heidelberg, 2009, pp. 9–18.
- [44] N. A. Mutawa, I. Baggili and A. Marrington, "Forensic analysis of social networking applications on mobile devices," *Digit. Invest.* vol. 9, pp. S24–S33, 2012.

- [45] E. S. Pilli, R.C. Joshi and R. Niyogi, "Network forensic frameworks: Survey and research challenges," *Digit. Invest.* vol. 7, pp. 14–27, Feb. 2010.
- [46] J. Bullock, J. T. Parker, "Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework," in *Wiley*, John Wiley & Sons, Inc., Indianapolis, IN, 46256.
- [47] J. W. Creswell, "Identifying a research problem," in *Educational research: Planning, conducting and evaluating quantitative and qualitative research*, 4th ed., vol. 4, Pearson, Ed. Boston, MA, USA, 2012.
- [48] P. Rajasekar, "Research Methodology: An Introduction," 2010. [Online]. Available: <http://www.newagepublishers.com/samplechapter/000896.pdf>. [Accessed 5 December 2019].
- [49] W. Goddard and S. Melville, "Research Methodology: An Introduction," 2004. [Online]. Available: <https://books.google.com.pk/books?hl=en&lr=&id=bJQJpsU2a10C&oi=fnd&pg=PA1&dq=Research+Methodology:+An+Introduction&ots=XtvaNdDO3o&sig=eZnJi5DH40fuShspRoGZCIZ5TFU#v=onepage&q=Research%20Methodology%3A%20An%20Introduction&f=false>. [Accessed 5 December 2019].
- [50] K. Kent, S. Chevalier, T. Grance and H. Dang, "Guide to integrating forensic techniques into incident response," NIST Special Publication, pp. 800–886, Aug. 2006. DOI: <https://doi.org/10.6028/NIST.SP.800.86>.
- [51] E. Peter, R. M. Baiju, N. O. Varghese, R. Sivaraman and D. L. Streiner, "How to develop and validate a questionnaire for orthodontic research," *Eur. J. of Dent.*, Wolters Kluwer, vol. 11, no. 3, Jul- Sep. 2017.
- [52] K. C. Patel and P. Sharma, "A Review paper on pfsense – an Open source firewall introducing with different Capabilities & customization," *Int. J. Adv. Res. Innov. Ideas in Edu.*, vol. 3, no. 2, pp. 2395–4396, 2017.
- [53] "Latest Stable Version," pfsense, [Online]. Available: <https://www.pfsense.org/download/>.
- [54] V. Asghari, Shi. Amiri and Sha. Amiri, "Implementing UTM based on PfSense platform," in *Proc. 2nd ICKBEI*, Tehran, Iran, 2015, pp. 1150–1152.
- [55] "13.35.180.119", ipinfo.io, [Online]. Available: <https://ipinfo.io/13.35.180.119>.
- [56] S. Vandeven, "SSL/TLS: What's under the hood," SANS Institute InfoSec Reading Room, Aug. 2013.
- [57] "Forward Secrecy at Twitter," Engineering, [Online]. Available: [https://blog.twitter.com/engineering/en\\_us/a/2013/forward-secrecy-at-twitter.html](https://blog.twitter.com/engineering/en_us/a/2013/forward-secrecy-at-twitter.html).
- [58] "video.twimg.com," W3 Snoop, [Online]. Available: <http://video.twimg.com.w3snoop.com/>.
- [59] "Twitter Firehose vs. Twitter API: What's the difference and why should you care?" Bright Planet, [Online]. Available: <https://brightplanet.com/2013/06/25/twitter-firehose-vs-twitter-api-whats-the-difference-and-why-should-you-care/>. [Accessed 29 January 2020].
- [60] "Tutorials- Consuming streaming data," Developer, [Online]. Available: <https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data>.

- [61] “pbs.twimg.com Server IP,” Webiplookup, [Online]. Available: <https://webiplookup.com/pbs.twimg.com/>.
- [62] “Information about IP-address - 192.229.220.133,” UANIC domain register, [Online]. Available: <https://whois.uanic.name/eng/ip/192.229.220.133.html>.
- [63] “Upload media- Post media upload Finalize,” Twitter Developer, [Online]. Available: <https://developer.twitter.com/en/docs/media/upload-media/api-reference/post-media-upload-finalize>.
- [64] “Upload media- Media best practices,” Twitter Developer, [Online]. Available: <https://developer.twitter.com/en/docs/media/upload-media/uploading-media/media-best-practices>.
- [65] “Message attachments- Overview,” Twitter Developer, [Online]. Available: <https://developer.twitter.com/en/docs/direct-messages/message-attachments/overview>.
- [66] “Sending and receiving events, Message Create Object,” Twitter Developer, [Online]. Available: <https://developer.twitter.com/en/docs/direct-messages/sending-and-receiving/guides/message-create-object>.
- [67] “Fetching media by URLs with Rest API,” stack overflow, [Online]. Available: <https://stackoverflow.com/questions/33364391/fetching-media-by-urls-with-rest-api>.
- [68] S. Davidoff and J. Ham, “Practical Investigative Strategies,” in Network Forensics Tracking Hackers through Cyberspace. Westford, Massachusetts, USA: Prentice hall: Upper Saddle River, vol. 2014, 2012.
- [69] “Facebook Hack: Social Network Confirms 13.4 Million Follower Twitter Account Compromised,” Forbes.com, [Online]. Available: <https://www.forbes.com/sites/daveywinder/2020/02/08/facebook-hack-social-network-confirms-134-million-follower-twitter-account-compromised/#1f2926513b5e>
- [70] “Unprecedented Twitter Hack Targets Celebrity Accounts,” Statista.com, [Online]. Available: <https://www.statista.com/chart/22296/accounts-compromised-in-twitter-cyber-attack/>
- [71] “An update on our security incident”, blog.twitter.com, [Online]. Available: [https://blog.twitter.com/en\\_us/topics/company/2020/an-update-on-our-security-incident.html](https://blog.twitter.com/en_us/topics/company/2020/an-update-on-our-security-incident.html)