Network Forensic Analysis of Twitter Application on Android OS

Alia Umrani
School of Computer Science and Information Technology
University College Cork (UCC)
a.umrani@cs.ucc.ie

Yousra Javed
School of Information Technology
Illinois State University
yjaved@ilstu.edu

Muhammad Iftikhar
Computer Science Department
Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology
m.iftikhar@lrk.szabist.edu.pk

***Abstract*— The increased use of secure social media apps on smartphones has attracted the interest of criminals looking to engage in illegal activities. The additional user confidentiality requirements, as well as the security features of these applications, complicate forensic investigations. The extensive analysis of encrypted traffic sessions, on the other hand, has the potential of identifying involved parties and their activities. In this paper, we conduct a network traffic analysis of Twitter, a popular social media application that uses encryption to protect information transmitted over the network. We concentrate on the Android platform to generate Twitter traffic, analyze fixed patterns, and extract artifacts based on various user actions. A firewall is also used to investigate Twitter's hidden design flexibilities and alternate connectivity options. Through our analysis, we were able to correctly identify the flow of Twitter traffic, user-related information, and fixed patterns to classify different user activities on Twitter.**

Keywords—Android, Encryption, Network Forensics, Traffic analysis, Twitter.

I. INTRODUCTION

Online communication through social networking has given cybercriminals an open platform to commit criminal activities [1]. The increased use of social media applications on smartphones has also increased the rate of cyber incidents [2]. However, forensic investigation of such applications has become difficult due to the intense security features and diverse architecture of such applications.

Millions of individuals utilize the safe social media platform known as Twitter worldwide. On average, 145 million people use Twitter every day. About 120 million people use desktop and mobile applications to access Twitter, out of which 31.1% use its Android application [4], according to Twitter statistics [3]. This has resulted in more crimes being committed using the Twitter application. For instance, 650 people were charged with crimes related to Twitter in 2008 [5]. Similarly, in 2016, a user leaked the sensitive data of 41 million Twitter accounts including email addresses and credentials [6]. In 2020, a study connected the frequency of tweets with real murder cases, and the results were highly accurate in identifying the regions with the highest crime rates. [7]. Other studies suggest that cybercriminals were also involved in the hacking of celebrity Twitter accounts [8] and the theft of Twitter employee credentials used to access internal resources [9]. With the rise in Twitter-based crimes, it has become pertinent to carry out network traffic analysis which can help investigators in analyzing the communication flow. However, the distributed architecture and encryption of information over the network makes this task challenging. Conversely, the extensive analysis of encrypted content can reveal various artefacts of user behavior [10] on secure applications.

Several studies have been proposed on traffic analysis of Twitter to identify user activities, however, many techniques failed to classify different user actions from the encrypted traffic. In 2016, a study proposed the analysis of encrypted network traffic of social networking applications [11] where the application's traffic is identified via IP address and timestamps. However, analysis based on these features can only identify a few user activities.

We introduce a detailed network profiling of Twitter and carry out the technique of fixed pattern analysis of individual packets to identify and classify the user activities. The identification and classification are based on extensive packet-level traffic analysis on Android OS. In network forensics, packet analysis is a basic technique used to trace back a network activity to a particular time [12]. Our analysis does not require prior knowledge of the protocol structure and security architecture of Twitter. Instead, the classification is performed through the analysis of a stream of fixed patterns created against each user action. Fixed pattern analysis is based on the observation of IP address and port, the number of bytes exchanged for each packet, the flow of TLS handshake, timestamps, and flow graph analysis. Therefore, an extensive behavior analysis technique can be used to retrieve important artefacts that can help a forensic investigator. To the best of our knowledge, there is no existing work on network forensic analysis of Twitter based on the fixed pattern analysis technique.

In this paper, the network traffic analysis is performed against different chat and media sharing activities, by first understanding the flow of traffic [13]. This resulted in the identification of certain fixed patterns that helped in the understanding the traffic behavior and classification of user activities. A firewall is also included to learn the distributed Twitter architecture. The installation of a firewall helped in the blocking of traffic, forcing the client to connect to the server via alternate connectivity paths, revealing other connectivity options and hidden design flexibilities in Twitter's architecture. The main contributions of this paper are as follows*:*

1) We present the hidden design flexibilities in the distributed architecture of the Twitter application that can help identify Twitter traffic over the network.

2) We propose a framework to perform an extensive Twitter traffic analysis for classifying the encrypted traffic into 6 different user activities on Twitter and identifying patterns that can assist investigators.

The rest of the paper is organized as follows. Section II presents the related work. Section III discusses the methodology of traffic analysis. Analysis and results are elaborated in Section IV. Section V discusses the practical applications and limitations, and section VI concludes the paper.

II. RELATED WORK

Several studies on Twitter network forensics have been conducted to analyze artefacts, vulnerabilities, fake accounts, and spammers on Twitter. We provide an overview of related work on network forensic analysis of social media applications in this section.

Afzal et al. in [21] presented an encrypted network traffic analysis of signal messaging application that applies encryption to secure communication over the network. Using payload analysis, the study detects encrypted traffic activities such as chatting, media messages, audio, and video calls. Farhood et al. [11] proposed a study for Android OS device and network forensic investigation of Twitter, Facebook, LinkedIn, and Google. Client and server IP addresses were retrieved from network forensics analysis, and timestamps were analyzed to determine the duration of activities. The retrieved artifacts, however, did not identify the user activities. Daniel et al. [14] examined the network and device forensics of 20 social media applications. The study focused on extracting artifacts from unencrypted traffic such as text message exchanges, videos, pictures, passwords, screenshots, and voice notes. However, network forensic analysis was only performed on the 20 applications that had unencrypted traffic over the network. Mohd et al. [15] examined Facebook, Twitter, and Telegram traffic on Firefox Mobile OS. To capture traffic and analyze user actions, a virtual cloud environment was used. The user credentials for the Telegram application were obtained from unencrypted traffic. The originating IP address, originating countries, and certificate providers were identified for Facebook and Twitter. Their work, however, is not synchronized between performed user activities and traffic behavior analysis against those activities. As a result, traffic classification becomes more difficult. Furthermore, the presented approach requires a large amount of system memory. Molnar et al. [16] attempted to identify and analyse Skype traffic using various algorithms. These algorithms are based on the dynamic flow and packet characteristics. By detecting the port, IP addresses, and timestamps, the Skype hosts and connection initiation was identified. The discovered parameters were then used to analyse the flow of calls on the Skype network. Their work is solely focused on identifying Skype traffic.

Furthermore, several machine learning and data mining techniques for encrypted traffic classification have been proposed. Using supervised machine learning, the encrypted traffic for Gmail, Facebook, and Twitter on Android smartphones has been analyzed [10]. A labeled dataset was created and trained to recognize user actions. IP addresses, flow direction, port, and packet size are all defined parameters. However, the attributes for analyzing the communication flow of encrypted Twitter traffic and classification of user actions, are limited and can only identify known features (although the attributes show high accuracy, but since it is a supervised machine learning, it can create a high false-positive rate if new features are involved in the packet header) . Wang et al. [17], presented a machine learning-based technique for classifying iOS app usage from encrypted 802.11 frames. However, the frames are only collected for 5 minutes by running apps from various categories, which can lead to inaccuracy if the training dataset is missing any frames. Similarly, Marik et al. [18] presented a data mining approach for identifying and evaluating vulnerabilities in Viber communication patterns. According to their findings, the Viber application has a dynamic and distributed server architecture, as well as an unknown protocol structure. However, the server is identified probabilistically, which can result in inaccurate results. Furthermore, no proper statistical evaluation of user activities exists. Sudozai et al. [2] presented a device and network forensic analysis of the IMO application. Their method correctly identifies and categorizes user actions on the IMO app. Twitter has a more distributed infrastructure that requires a detailed analysis of fixed patterns containing various features.

This paper adds to the existing literature by conducting a detailed encrypted network traffic analysis of Twitter on Android OS. A fixed pattern analysis technique is used along with packet-level analysis to help in traffic classification and user behavior identification. The proposed methodology is generic and can be applied to any secure application with a client-server architecture based on HTTPS.

III. METHODOLOGY

This section discusses the experimental setup and framework for network forensic analysis of Twitter.

A. Experimental Setup

We set up a network infrastructure to intercept traffic (see Figure 1). For internet access, a target mobile device was connected to a Wireless Access Point (WAP). The WAP traffic was routed through the LAN interface of a FreeBSD distribution pfSense firewall [19]. A PC was also connected to the WAP to capture the traffic with Wireshark. The firewall's network interface was linked to a layer-3 switch.

Wireshark was used to capture the traffic for each user activity on Twitter. We used various Wireshark filters to further narrow down the captured traffic [20]. The firewall was set up to filter mobile device traffic so that we could only focus on Twitter traffic. We used a firewall to discover the Twitter application's hidden design flexibility. Application developers create applications with flexibility in design to ensure service availability. There are multiple servers distributed across the network to provide backup connections in case servers becomes unavailable. As a result, the goal of using a firewall is to identify potential alternate Twitter connections by imposing some restrictions on known servers. This assisted us in determining the alternate possible server IP address ranges used by Twitter for each activity.

Following the configuration of the environment, the Twitter app version-8.22.0-release.00 was installed on an Android smartphone. The application was then used to carry out six different user actions. A single user on an Android

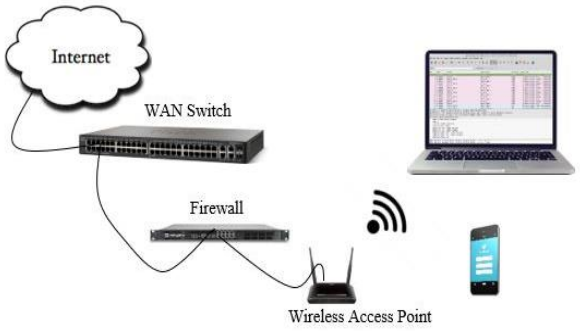

Figure 1: Experimental Setup

mobile performed each user activity 10 times with a 15-minute interval. The traffic was captured against each activity

for behavior analysis and to identify the IP address ranges of servers responsible for various services. The firewall rules were updated after each analysis, and the process was repeated.

B. Analysis Framework

We downloaded and installed Twitter on Android OS for network forensic analysis. After installation, we investigated some of the application's features. Login, logout, media sharing, and chat activities are the most common features. Figure 2 depicts our framework for analysis. The process begins with the identification of the IP address range and port range shown on the left. Following the range observation, the process for behavior analysis against various user activities is demonstrated. Following the environmental setup, the application is accessed, and activity is carried out. We enabled the traffic capturing tool before beginning an activity. In the first step, traffic was collected in response to various user actions. We deduced the server IP addresses and ports from the captured traffic. Each newly observed IP address was noted and blocked using firewall filters. The IP address range observation and blocking procedure was repeated until the application server stopped responding. This investigation resulted in the complete set of Twitter servers.

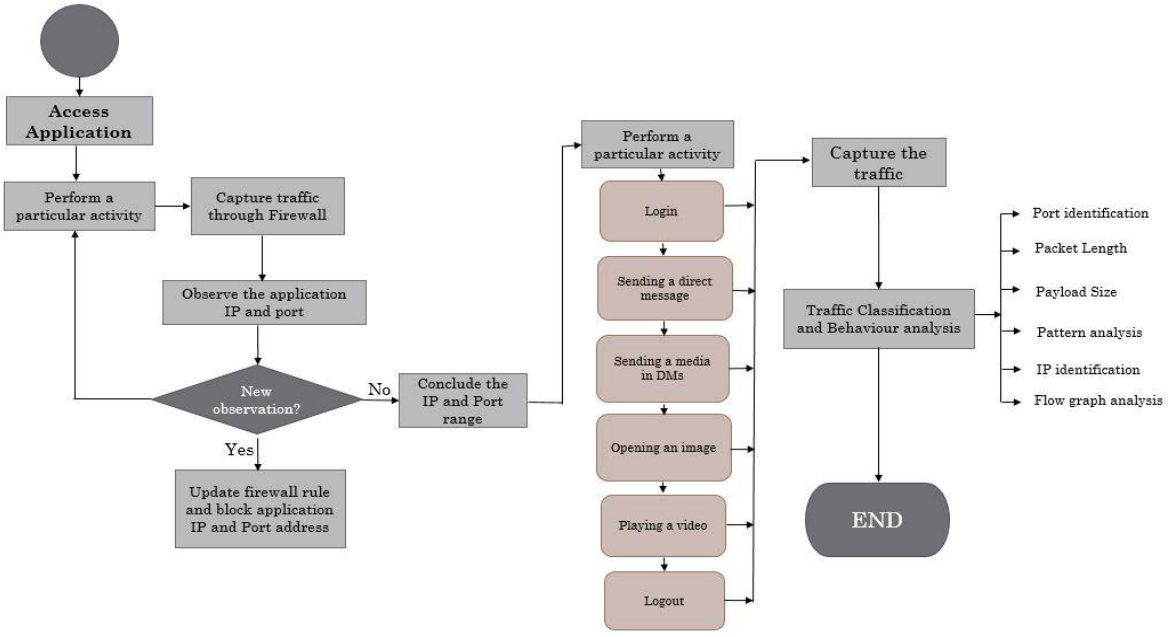

Figure 2. Analysis Framework

For behavior analysis, we carried out several activities and recorded traffic for each one. The behavior analysis process was not iterative, and the performance of one activity was not dependent on the performance of another. The observed behavior includes Fixed pattern analysis, IP address and port identification, packet length, payload size, and flow graph analysis. The experiments for behavior analysis and IP address range analysis were performed within 4 months.

IV. ANALYSIS AND RESULTS

This section discusses the IP address range analysis and behavior analysis in relation to the six user actions listed in Table I. The activities were chosen based on their frequency of use and their involvement in crimes such as illegal data sharing via messaging from private accounts. Furthermore, the network traffic generated by these activities is random, requiring detailed analysis to uncover fixed patterns.

A. IP address range identification

The process of identifying IP address ranges for login servers, chat servers, and media servers, as well as the results obtained, are described below.

1) Login server: We logged into Twitter in a controlled environment to observe the IP address range of the login server. The application was accessed without any IP address restrictions on the firewall in the first step. Each time the client accessed and logged in, it established a connection with the Twitter server using the initial IP addresses listed in Range I (see Table II). Following that, we blocked IP addresses found in Range I, accessed Twitter, and analyzed the traffic. According to the results, the client connected with an IP address from Range II. Similarly, the process was repeated, and IP addresses in Ranges III and IV were discovered. After blocking the entire observed range, the client was unable to log into the application, implying that Twitter uses the observed server IP addresses to authenticate and provide access to the client. Table II contains the entire IP address range.

2) Media server: To identify the IP address range of the media server, we opened and viewed an image in the media tab and captured the traffic. During this activity, the client was always communicating with 93.184.220.70. When the captured traffic from multiple files was analyzed, it was discovered that the client was constantly connecting to another IP address 192.229.220.133 (belonging to Verizon communication services). We then blocked these IP addresses and identified the IP address range for accessing an image in the following steps. Following the analysis, the new IP address was discovered to be within the same range as the previous IP address 192.229.233.50. After restricting all of the observed ranges for the chat server, it was discovered that the client was unable to open an image. This meant that the

media server's observed IP addresses were 93.184.220.70, 192.229.220.133, and 192.229.233.50.

TABLE I. USER ACTIVITIES

Activity	Description
Login	Using credentials, a user logs into the application
Viewing images	The user opens an image in a media tab
Playing videos	The user plays any video in a media tab
Sending text messages	User types and sends a text message to another user
Sending media messages	The user attaches a media and sends it to another user
Logout	User logs out of the application

3) Chat server: In this activity, we sent a text message to another Twitter user in order to determine the IP address range of the chat server. In the first step, we sent a text message and recorded the traffic. It was discovered that the client established a connection with the destination IP addresses in Range V (see Table II). Similarly, when we repeated the process, we discovered IP addresses in Range VI and VII. When the entire range was blocked, the client was unable to send any text messages via the Twitter app. The above range identification was created specifically for sending a text message. We also sent media (images, videos) in messages and observed the IP address range. When the client sent an image on the first try, the connection was established with IP addresses in Range VIII (see Table II). During this activity, a new destination IP address 13.35.180.119 belonging to Amazon.com Inc. was discovered.

The new observation with Range IX IP addresses resulted from further blocking of the obtained range. The client was still able to send media in direct messages, and Range X IP addresses were examined. Finally, the client was unable to send any media messages after blocking the entire range. Table II displays the entire IP range for the chat server.

B. Behavior analysis for Twitter Activities

The unique traffic patterns can be used to recognize and categorize user activities. We carried out six different Twitter activities and analyzed the traffic behavior in relation to these activities. This section presents the accuracy of the traffic behavior analysis results.

1) Login: We logged into Twitter using an Android Galaxy J7 phone and recorded the traffic for analysis. The first step was to identify Twitter traffic on the network, which can be accomplished using a number of techniques. One method is to repeat the same activity multiple times and capture the synchronized traffic to determine the IP address. We repeated similar steps in the beginning to determine the IP address. We discovered several packets in the captured files after logging in, including notifications on a mobile device and background application updates, etc. Wireshark filters were used to filter Twitter traffic, and behavior was analyzed. Step-by-step behavior analysis is defined in the subsections below.

When the client first logged into the application, a SYN packet with the IP address 192.168.137.230 was sent requesting a connection. When the server with IP 104.244.42.66 received the connection request, it responded with SYN, ACK packets, and the connection was established. The TLS Handshake was performed after the connection was established. The user was logged into the application via HTTPS-based authentication after the TLS Handshake.

TABLE II. IP ADDRESS RANGE

Activity	Observed Range
Login server	
Login before IP restrictions	Range I: 104.244.42.66, 104.244.42.65, 104.244.42.131
Blocking Range I	Range II: 104.244.42.130, 104.244.42.195, 69.195.186.128, 69.195.161.128
Blocking Range II	Range III: 104.244.42.194
Blocking Range III	Range IV: 104.244.42.129, 104.244.42.67, 104.244.42.2
Blocking Range IV	The client was unable to login into the application
Chat server (Text message)	
Sending a text message before IP restrictions	Range V: 104.244.42.67, 69.195.160.128
Blocking Range V	Range VI: 104.244.42.66, 104.244.42.194, 104.244.42.3
Blocking Range VI	Range VII: 104.244.42.2, 104.244.42.129, 104.244.42.193
Blocking Range VII	The client was unable to send a text message
Chat server (Media message)	
Sending a media message before IP restrictions	Range VIII: 104.244.42.148, 13.35.180.119, 104.244.42.194, 104.244.42.195, 69.195.168.128, 69.195.167.128
Blocking Range VIII	Range IX: 104.244.42.2, 104.244.42.131, 143.204.106.90
Blocking Range IX	Range X: 104.244.42.84, 104.244.42.66, 143.204.106.11, 143.204.106.28, 104.244.42.203
Blocking Range X	The client was unable to send a media

The user enters the credentials (username and password) via the login form. For Twitter, the information over the network is encrypted, therefore the user credentials cannot be retrieved. Further, if the user enters invalid credentials, the server sends a retransmission packet and redirects the client to the login page.

2) Logout: We analyzed traffic packets in the logout activity. When a user logs out, the HTTPS session with the server is terminated. While logging out, we noticed that the client connected to one of three server IP addresses: 104.244.42.2, 104.244.42.193, and 104.244.42.65. The client with the IP address 192.168.137.230 sent a [FIN, ACK] message to the server with the IP address 104.244.42.65 on two client TCP ports 60072 and 60071 with a packet length of 66. The server, on the other hand, sent an 'Encrypted Alert' message to the client on port TCP 60071 with a payload size of 31 bytes and a packet length of 180. The server also sent the [FIN, ACK] packet on port 60071 which shows that the connection is terminated. Table III shows the traffic patterns for logout activity.

3) Opening an image: When a client opens the media tab, it sends multiple DNS requests to pbs.twimg.com, Twitter's image hosting domain. Another request is made to video.twimg.com, Twitter's video hosting server. When a Twitter client opens an image, a connection is made with 93.184.220.70, which is the IP address of the pbs.twimg.com

server. After the connection is established, the client with IP 192.168.137.61 performs a TLS Handshake with pbs.twimg.com source TCP port 43245 and destination HTTPS port 443.

Table III. TRAFFIC PATTERN FOR LOGOUT

Session	Packet Patterns	Packet Length
Client → Sever	[FIN,ACK]	66
Server → Client	Encrypted Alert	180
Server → Client	[FIN, ACK]	118

When the handshake is finished, the client and server exchange encrypted data. While the image is open, the server sends TCP segments with a fixed packet length of 3014 to the client. The payload size of these packets was also found to be fixed at 1448 bytes. In response to the server's packets, the client sends an acknowledgment for each packet. Table IV displays the discovered pattern for opening an image.

TABLE IV. TRAFFIC PATTERN FOR OPENING AN IMAGE

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

4) Direct message: When a Twitter client sends a direct message, a predefined set of packets is sent to a specific IP address and port. This can be used to identify Twitter chat activity. We discovered that the client with IP 192.168.137.125 connected to Twitter with IP 104.244.42.3 on two different client TCP ports, 55807 and 49868. Further investigation revealed that the client sends TCP segments with a fixed packet length of 1514 bytes and encrypted application data with a packet length of 198 bytes on port 55807. The server acknowledged by sending an ACK with a fixed length of 118 bytes and encrypted application data with a fixed length of 1696 bytes.

5) A similar pattern was observed in another observation for the same TCP port. The packet length, however, differs from the previous observation. The Twitter client sent an encrypted packet with a length of 1494 bytes, to which the server responded with an ACK packet with a length of 118 bytes. It also sent encrypted data with a fixed length of 2872 bytes, and this pattern was repeated for this IP and TCP port. For TCP port 49868, the Twitter server with IP 104.244.42.3 sent an encrypted application data with a length of 320 bytes to the Twitter client with IP 192.168.137.125. The client sent an ACK with a fixed length of 66 bytes, and the patterns remained consistent throughout the activity. Table V depicts the identified pattern for sending a text message.

Sending media message: When a Twitter client sends a direct message with an image attached, the traffic is routed to upload.twitter.com. While the image is being attached, a connection is established with upload.twitter.com, which has the IP address 104.244.42.75, and a TLS Handshake is performed before the server sends the encrypted data. When the data is received from the server, the uploading activity begins, in which a stream of packets with a fixed packet length of 1514 and a payload size of 1448 bytes is sent from client to server until the attachment is completed. When a user sends the attached media, a connection is established with ton.twitter.com, which has an IP address of 104.244.42.148, and a stream of packets with a fixed packet length of 3014 bytes and a payload size of 1448 bytes is sent to the client from ton.twitter.com. The stream of packets is so frequent that when it stops, the client responds with an acknowledgment packet.

TABLE V. TRAFFIC PATTERN FOR SENDING A TEXT MESSAGE

Session	Identified Packet Patterns	Packet Length
On port 55807 (Observation 1)		
Client → Server	TCP Segments	1514
Client → Server	Data Packet	198
Server→ client	ACK	118
Server→ client	Data Packet	1696
On port 55807 (Observation 2)		
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

6) Playing a video: Several videos were played, and traffic behavior was analyzed. Based on the analysis, whenever a user watches a video, the traffic is routed to IP 68.232.34.217, which belongs to Verizon services. As a result, this server provides video services to Twitter Inc. It was discovered that some continuous patterns with a length of 2814 and a payload size of 1360 bytes are sent from the application server with IP 68.232.34.217 and port 443 to the client IP 192.168.137.201 at port 39740. Furthermore, random packets are generated when a client pauses or stops a video.

7) Protocol verification: We verified the identified patterns by running scripts on random packet captures. We searched the captured files for patterns related to Twitter user activities. Following verification, it was discovered that our script correctly identified all user activities with 0% false positives and false negatives and 100% true positives and true negatives. Login, open an image, play a video, and logout are among the identified activities.

V. DISCUSSION

This section discusses the practical implications and limitations of this work.

A. Practical implications

Since Twitter is a widely used application, it has become a major platform for criminals to engage in illegal activities such as leaking sensitive data, spreading hate speech, and spreading rumors against the government or a specific organization. In the event of data leakage, it must be determined whether the source is internal or external. The first step in investigating this is to gain access to the suspected network and collect traffic dumps for the duration of interest. Organizational backups can be used to obtain traffic dumps. The next step is to identify and filter a specific application. Our research reveals artifacts from encrypted traffic that can be used to create a chain of evidence. For example, timestamps can be used to reveal the duration of an event's occurrence. IP addresses can reveal the identity of a server that has been accessed, such as login, media, or chat server. Following the identification of a server, pattern analysis and

flow direction analysis assist in determining which activity was performed, i.e., application login, chatting, media accessed or exchanged. Furthermore, IP packet headers reveal the identity (via source IP), which can be revealed by involving relevant ISPs. As a result, identifying IP ranges of different servers and performing behavior analysis can help a forensic investigator solve a case.

B. Limitations

Our analysis shows that the header of IP packets contains the source and destination IP addresses, which helps in the identification of the parties involved. However, if an organization has defined Network Address Translation (NAT), forensic investigators will have a difficult time identifying the parties involved. Furthermore, the identification of user activities is based on fixed pattern analysis of encrypted packets; however, if Twitter employs additional security measures such as padding, the fixed patterns will be difficult to analyze.

VI. CONCLUSION

In this paper we conducted a network forensic analysis of Twitter on Android OS. Before our analysis, the application's IP range for different servers (e.g., chat server, media server) was unknown, and the network traffic analysis was performed on encrypted traffic. We used this technique to identify the IP addresses responsible for certain features of the Twitter App, and the results aided in activity pattern analysis. We investigated the application for various features, carried out activities, and conducted forensic analysis based on the traffic classification technique to find possible remnants of those activities, which can be of great assistance to forensic investigators. A firewall was also used to discover Twitter's hidden design flexibilities. The firewall deployment helped in the identification of Twitter's alternate connectivity patterns. To conduct the behavior analysis, we performed various user actions and classified the encrypted traffic in order to discover various characteristics. Port-based analysis, IP address identification, packet length, payload size, timestamps, and fixed pattern analysis were used to classify the traffic. We discovered various artefacts in response to various user events. The destination IP revealed the Twitter server's identity. Fixed packet length and payload size helped in user behavior analysis, and timestamps displayed the duration of event occurrence.

REFERENCES

[1] N. Al-Mutawa, I. Baggili, and A. Marrington, “Forensic analysis of social networking applications on mobile devices,” Digit. Invest., vol. 9, pp. S24-S33, Aug. 2012.

[2] 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.

[3] 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/.

[4] G. Udani, “An Exhaustive Study of Twitter Users Across the World,” beevolve, Oct. 10, 2012. [Online]. http://www.beevolve.com/twitter-statistics/

[5] 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

[6] ElevenPaths, “Investigation report “Twitter Data Leakage” Eleven Paths, Jun. 29, 2016. [Online]. Available: https://www.elevenpaths.com/investigation-report-twitter-data-leakage/index.html.

[7] R. P. Curiel, S. Cresci, C. l. Muntean and S. R.Bishop “Crime and its fear in social media,” Palgrave Comm., vol. 6, Apr. 2020, Art. no. 57.

[8] F. Richter, “Unprecedented Twitter Hack Targets Celebrity Accounts,” Statista, Jul. 16, 2020. [Online]. https://www.statista.com/chart/22296/accounts-compromised-in-twitter-cyber-attack/.

[9] Twitter Inc., “An update on our security incident,” Twitter Inc., Jul. 30, 2020. [Online]. https://blog. twitter.com/en _us/topics/ company/2020/an-update-on-our-security-incident.html.

[10] 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

[11] F. N. Dezfouli, A. Dehghantanha, B. Eterovic-Soric, and K. K. R. Choo, “Investigating Social Networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms,” Australian J. of forensic sciences, vol. 48, no. 4, pp. 469-488, 2016.1080/00450618.2015.1066854.

[12] L. F. Sikos, "Packet analysis for network forensics: A comprehensive survey," Digit. Invest., vol. 32, pp. 200892, Feb. 2020.

[13] 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.

[14] D. Walnycky, I. Baggili, A. Marrington, J. Moore, and F. Breitinger, “Network and device forensic analysis of Android social-messaging applications,” Digit. Invest., vol. 14, pp. SS7–S84, Aug. 2015.

[15] M. N. Yusoff, A. Dehghantanha, and R. Mahmod, “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.

[16] 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.

[17] Q. Wang, A. Yahyavi, B. Kemme and W. He, "I know what you did on your smartphone: Inferring app usage over encrypted data traffic," 2015 IEEE Conference on Communications and Network Security (CNS), 2015, pp. 433-441, doi: 10.1109/CNS.2015.7346855.

[18] 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.

[19] 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.

[20] J. Bullock, J. T. Parker, “Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework,” in Wiley, John Wiley & Sons, Inc., Indianapolis, IN, 46256.

[21] Afzal, A.; Hussain, M.; Saleem, S.; Shahzad, M.K.; Ho, A.T.S.;Jung, K.-H. "Encrypted Network Traffic Analysis of Secure Instant Messaging Application: A Case Study of Signal Messenger App" Appl. Sci. 2021, 11, 7789.