

Digital forensics on Tencent QQ-instant messaging service in China

Yunkun Li, Gabriela Mogos

Department of Computing, School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China

Article Info

Article history:

Received Jun 29, 2022

Revised Sep 10, 2022

Accepted Sep 28, 2022

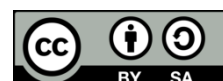
Keywords:

Digital forensics
Instance messaging
Kali Linux
Packet analysis
Wireshark

ABSTRACT

Data packet analysis targeting instant messaging (IM) applications has become one of the most mentioned case examples in the digital forensic industry, considering that the forensic engineers can extract valuable information by analysing the data packets used by the IM software. The crucial part of this process is to accomplish a series of research and investigation, in addition to correctly implement the related forensics tools. This paper is intended to use QQ, a popular IM software in China, as an experiment example, in cooperation with various tools from Kali Linux, a digital forensics-oriented Linux distribution, to present the complete process of the data packet analysis operation. The result concludes from the experiment may be able to provide constructive suggestions to other related digital forensics cases.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Gabriela Mogos

Department of Computing, School of Advanced Technology, Xi'an Jiaotong-Liverpool University

111 Ren'ai Road, SIP, Suzhou, China

Email: Gabriela.Mogos@xjtlu.edu.cn

1. INTRODUCTION

Data packet analysis is one of the most mentioned techniques in the network forensic branch of digital forensics [1]. In practice, there are lots of data packet analysis cases that target the instant messaging (IM) application software. For instance, a network forensic case that focused on discovering the IP address of the target by analyse the data packets of WhatsApp, which is a popular IM platform [2]. Throughout these cases, lots of researchers and forensic engineers has gradually built up a clear understanding of the importance of the forensics targeting IM software, especially when mentioning that, in modern society, people who own smart devices may prefer using one or more than one IM software to communicate with each other, and this may also apply to criminals or commercial espionage who choose to use IM software to transfer critical information.

Tsai *et al.* [3] suggests that the messaging content and the meta data contained by the data packets generated by the IM software have become one of the most critical types of evidence for law enforcement to obtain critical evidence in investigations of certain variations of cybercrime. However, we gradually found out that, in this industry, there are rarely similar forensic cases that target Chinese IM software or social media network platforms. Especially for one platform that is even frequently used: QQ.

Huang *et al.* [4] that focuses on the social impact of QQ in 2013, QQ has become one of the most popular social networks and instant messaging platforms in China, with millions of active users in 2013. In addition to this, quote from the official web page of QQ international [5]: (QQ has become) the most popular personal communications app in history: over 1,000,000,000 registered users across 80+ countries. Considering that there are existing study cases that target the other IM software, some key points of the methodology of research and experiment may be able to be referenced from these cases. Based on some

related works [2], [3] that have used WhatsApp as the main target of the digital forensics, in addition to a few other research and online articles [6], [7], a basic research framework that may fit this project has been proposed.

For the majority of the IM applications, including WhatsApp and QQ, when users are using their preferred smart devices to send messages to other users or other accounts, the message will be encoded as a data packet by the software. These data packets then will be sent by the devices by going through a series of network stacks and transporting to the main server of the IM platform, where these data packets will then be transported to the target user. In this whole information communication process, the mechanism that the software is used to process these data packets is the key differences between these IM applications. This determines how the information will be stored and distributed, including the activity patterns of the data flows and the structure of some critical information that the majority of IM application is required to establish the connection with other users, for example, the internet protocol (IP) address or the account name of the sender and the receiver. By capturing the data packets generated and transferred by this IM software with proper tools, meanwhile analyse how the information is being handled, the digital forensics engineers can then use this knowledge as a formatter to extract useful information from the captured data packets, especially considering that these captured data packets are just a bunch of bit streams which are not in a human readable form.

2. IMPLEMENTATION TOOLS

Wireshark is a popular open-source tool that is often used by network engineers and digital forensics engineers. It is packed with a set of powerful tools that, with the proper configurations, can capture and analyse the data packets of any device [8]. Considering that it has provided an intuitive graphic user interface (GUI), when trying to analyse the captured packets, the information can be directly displayed in an instructional window, and by using the built-in filter functionality, it will allow the user to filter out useless packets and only display the useful ones, which greatly reduces the complexity of the result analysis phase of this project.

In order to provide a relatively maintainable experiment environment, Kali Linux will be considered as the main experiment platform for this project. Kali Linux is a Debian Linux based operating system packed with a set of professional tools that is orienting in different branches of the digital forensics. According to the online documentation provided by Kali officials [9], Wireshark is pre-installed and well-configured so that the tool will work out-of-the-box. Furthermore, the lightweight design of the operating system itself allows it to install as the virtual machine or to-go system, which is designed to be installed on the flash drive and is more user-friendly to the digital forensics engineers considering that it will not produce extra problems that may affect the result of the forensics operations. The special quality of the Kali Linux operating system is especially helpful to the experiment.

In one of the cases that is targeting WhatsApp [2], the researchers have suggested using a "middle man" on the local area network (LAN) to sniff the data packets that are transferred by the IM software. Inspired by this research, in this project, a classic technique that is often used by computer security experts called Man-in-the-Middle Attack based on address resolution protocol (ARP) poisoning. Singh *et al.* [10] will be used to let the Kali Linux machine capture the data packets that will be sent from and to the targets. After initialising the ARP poisoning to the target machine, it will recognise the attacker machine (in this project, that is the Kali Linux machine) as the gateway of the LAN and send all the network traffic to the attacker machine, and all the data packets that originated from the target machine will go through the attacker machine.

At this point, all the attacker machines will need to do is to just listen and capture all the local data packets. To implement this technique, two special tools which are also provided by Kali Linux can be used: *arpspoof* which only provides a command line interface, and *ettercap* which provides an intuitive GUI interface. After brief research of these two tools [11], [12] and considering that the *arpspoof* can be initiated by a simple terminal command, which can streamline the experiment process, this project will select *arpspoof* as the tool to "redirect" the packets from the target. The basic usage of this tool can be concluded as follow command:

```
sudo arpspoof
    -i [network_interface_used_to_initialise_attack]
    -t [target_ip]
    -r [gateway_ip]
```

In addition to this, the Kali Linux machine will need to enable the IP forwarding capabilities before initialising the ARP poisoning so that the target machine can still connect to the Internet. In Kali Linux, this functionality can be enabled by modifying a configuration file in the system using the following

commands:

```
# in root account
echo 1 > /proc/sys/net/ipv4/ip_forward
```

3. METHOD AND RESULTS

3.1. Experiment environment

To analyse the data packet activities of the QQ software, the basic methodology is to study the common user behaviors of the software, and mimic these user behaviors during the experiment, while using the mentioned tools to record the data packets sent by QQ. After observing the normal QQ user activities, in addition to the experience of the authors, they are the QQ users in real life, the following commonly seen use cases of QQ software can be concluded:

- User log in the user's QQ account.
- User checking the friend's list.
- User sending and receiving text to another friend (other user account).
- User log out the account.

Although the QQ desktop application software itself also provides other functionalities, such as Qzone, which is similar to the user profile and user space functionality of Facebook, but this project will only focus on the instant messaging capability of QQ. Initially the experiment of this project was intended to use QQ for Linux version software as the target of packet analysis, considering that this experiment will mainly be conducted on the Kali Linux operating system. Unfortunately, according to multiple articles, forum discussions [13]-[15] and limited support from the official QQ for Linux web page [16], there are inconsistent user experiences and stability issues reported related to the official QQ for Linux version, which itself is also lacks support from the Tencent official, and according to the user feedback and discussions on various Chinese social platforms and the Linux community [17], [18], the user experience of QQ for Linux are "not very pleasant". There is also a Windows-immigration version of QQ developed and maintained by the deepin Linux community [19], but due to the incompatibility issue related to certain software dependencies of Kali Linux, the installation process of the deepin version of QQ caused system dependencies failure (Figure 1). Furthermore, considering the market share of the desktop operating systems available in the market [20] and the download count data of the QQ Windows version software in China according to [21], majority of users may consider installing the Windows version of QQ on the more widely used Windows operating system.

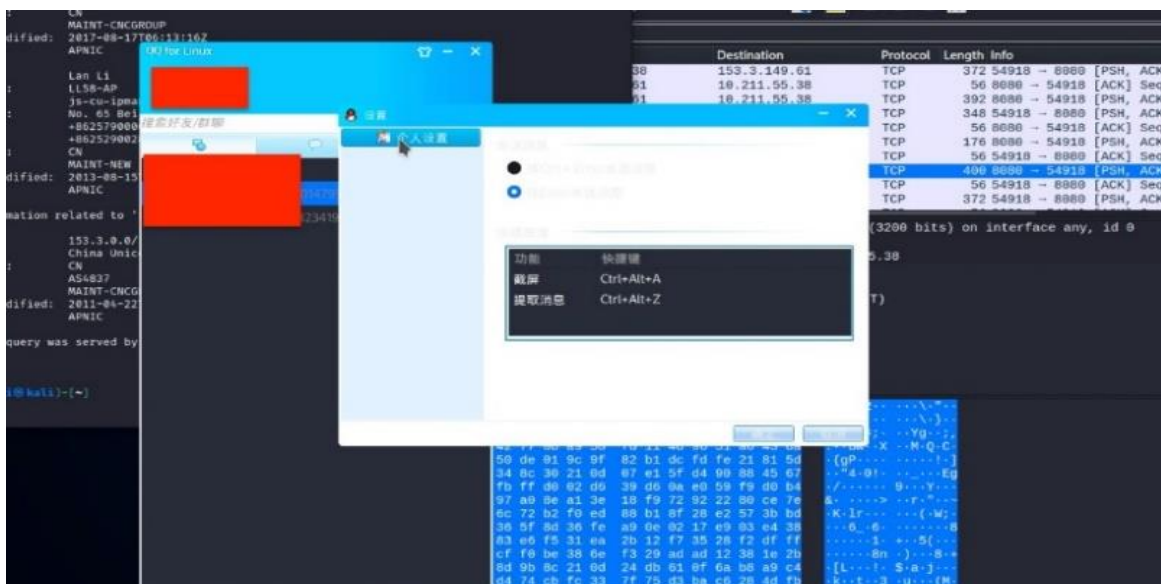


Figure 1. Bad user interface of QQ for Linux

To get more appropriate results that can cover the majority of QQ software use cases, our experiment will use two virtual machines as the experiment environment (Figure 2). Detailed configurations

of the two virtual machines (VM) are shown in the table below (Table 1). One VM will have the Kali Linux operating system with required software installed, the other VM will have the Windows 10 operating system with QQ for Windows software installed.

The host with Kali Linux installed will intercept, capture, and forward all the data packets sent from and to the Windows machine with proper virtual machine network configurations and the ARP spoof technique using the *arpspoof* tool. Wireshark will then be used in the Kali Linux host to analyse the activities and content of these captured data packets. Before the experiment starts, the pre-installed ping tool will be used to check the Internet and LAN connection capabilities of both machines.

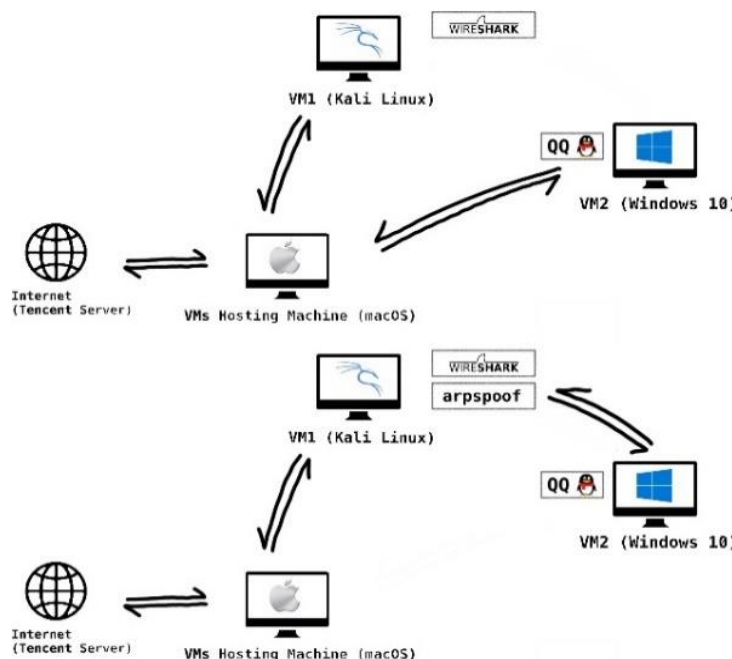


Figure 2. Network topology

Table 1. Machines configuration

Machine name	Host machine	VM1	VM2
Operating system	macOS 10.15.7	Kali Linux 5.16.0	Windows 10 1909
Installed software	Parallel Desktop 15 as the hosting program of these virtual machines	Wireshark version 3.2.5, <i>arpspoof</i> , and other basic dependencies	QQ version 9.5.9 (28650) and other basic software for Windows
Network mode of virtual machine	Connecting to Internet via Wi-Fi	Bridging network interfaces	Bridging network interfaces

3.2. Results

3.2.1. TCP and UDP connection in the link layer

With the proper configuration of virtual machines and the *arpspoof* tool, Kali Linux has successfully retrieved all the data packets sent from the Windows machine. After analysing the content of the data packets by using the build-in filter of Wireshark to get the target data packets, we conclude that QQ uses transmission control protocol (TCP) and user datagram protocol (UDP) connection protocols in the link layer of the open systems interconnection (OSI) network model to establish the connection with the main server and exchange data. One of the main network ports that is being used is the port 8080 (Figure 3). One of the most obvious instances is that QQ software has provided an advanced settings page in the account log in window to allow users to choose whether to use TCP or UDP to connect to the log in server, whereas providing a list of selectable servers to choose from.

In the "User account log in" experiment, after trying to use different connection methods (TCP or UDP) or different log in servers to log in the QQ account, the destination IP address of a few certain data packets which are captured by Wireshark will be the same as the one that is selected in the *log in* window. The value of the "protocol" attribute of these packets are also correspond to the one that is selected (Figure 4).

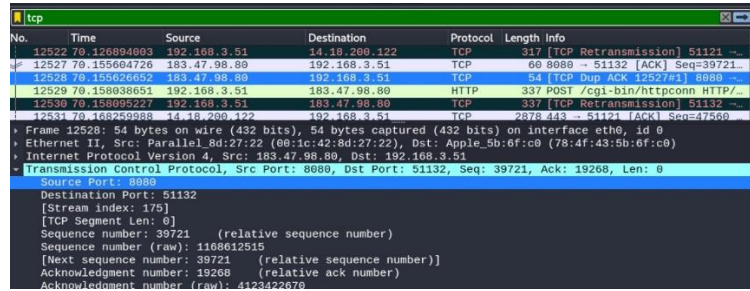


Figure 3. TCP connection captured by Wireshark



Figure 4. Log in page

3.2.2. HTTP and HTTPS connection in the application layer

After logging in, the QQ will start to fetch and load assets related to the account, including account avatar images and the friends list. Data packets that are responsible for carrying the uniform resource identifier (URI) of these assets use hypertext transfer protocol (HTTP) or hypertext transfer protocol secure (HTTPS) protocols. By analysing the HTTP response message of these data packets in the Wireshark window (Figure 5), these assets can be extracted outside the QQ software, but additional techniques related to cookies may require because it seems that, to fetch the assets from these URIs, a specific cookies, which is generated by the QQ software with the current user account log in, is required.

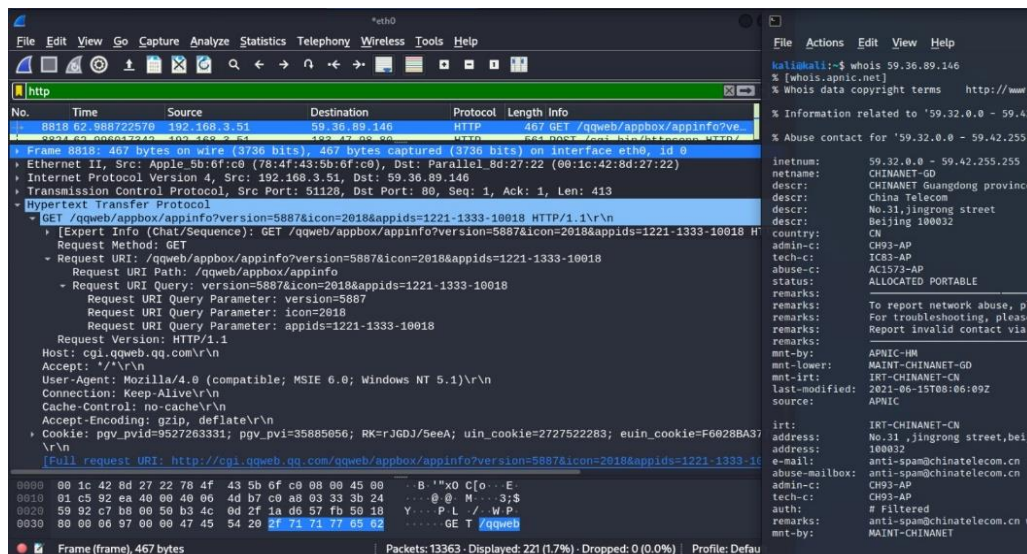


Figure 5. HTTP request captured by Wireshark

3.2.3. OICQ protocol

Another application layer protocol that QQ uses is the OICQ protocol (Perl extension for QQ instant messaging protocol), which was design by Tencent. The OICQ protocol is also readable and being formatted outputted in the Wireshark window. Meanwhile in the official documentation of Wireshark, a list of suggested filters is being provided [22]. As shown in the below screenshot images, by implementing a filter in Wireshark, all the packets that use the OICQ protocol are shown in the inspection window (Figure 6).

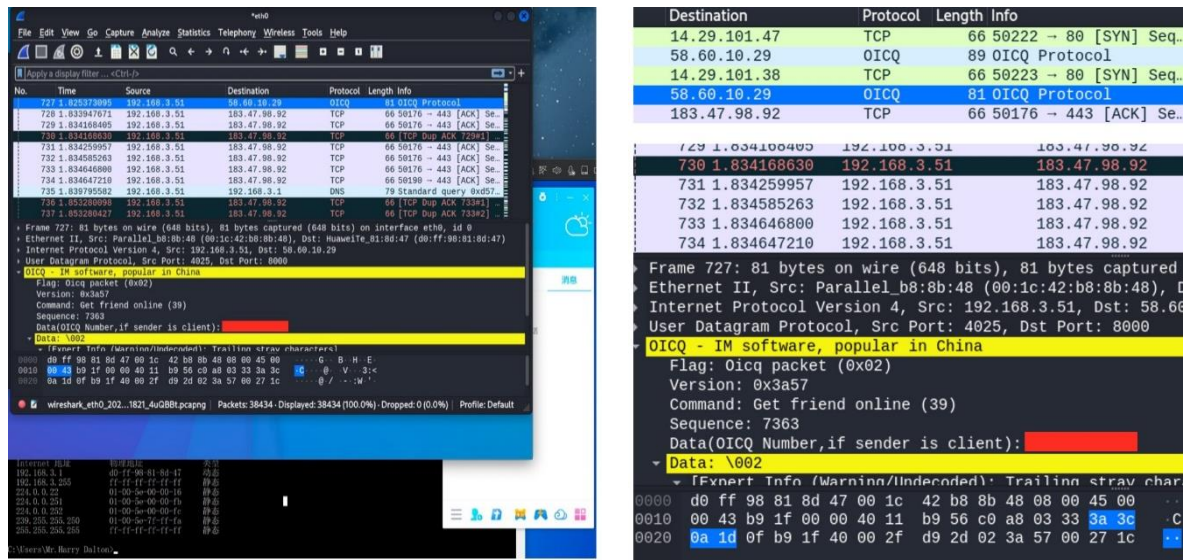


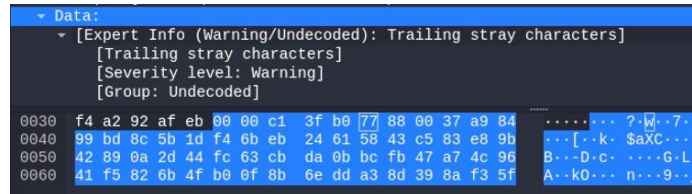
Figure 6. Wireshark inspection window

By reading the given information, the overall structure of the OICQ protocol can be summarised as:

- a. Packet flags indicates that this is an OICQ packet.
- b. Version number (in hexadecimal) of the software that the current user is using.
- c. Command number that indicates what operation this data packet is holding. According to the above screenshot, it shows that this packet is responsible of sending “Get friend online” request to the server. In this experiment, with a series of user behaviors being mimicked, by analysing the content of all the OICQ packets, following OICQ command numbers are being collected:
 - Request KEY (29).
 - Heart Message (2): keep the log in status alive.
 - Get status (13).
 - Get status of friend (129).
 - Get friend's status of group (181).
 - Request extra information (101).
 - Update User information (4).
 - Get level (92): get the account level.
 - Receive message (23).
 - MEMO Operation (62).
 - Signature operation (103).
 - Group name operation (60).
 - Log out (1).
- d. Data sequence number. For some commands sent from QQ, if the received data is too big to handle by one single packet, the data will then be split into a few different parts. For each part of the data, it will be carried by different packets. By using this sequence number, the software can rearrange the data and then display it to the users.
- e. OICQ number, if sender is client (software user). This will be the QQ account number of this user. It is a string of decimal number. Considering that this experiment is using the personal QQ account of the author himself, the account number that is covered by a red block in the above image will not be given.
- f. Other data.

Unlike other assets that use HTTP and HTTPS protocols to carry data, the OICQ protocol is mainly responsible for carrying assets that contain more sensitive information, including but not limited to chat data

(text or images transferred in the chat window) and friend list. Considering the importance of these data, according to [23], QQ seems to be using a multiple-round encryption algorithm called the tiny encryption algorithm (TEA encryption algorithm) to encrypt this part of the data. This can also be proved by the result obtained from the experiment (Figure 7).



```

Data:
- [Expert Info (Warning/Undecoded): Trailing stray characters]
  [Trailing stray characters]
  [Severity level: Warning]
  [Group: Undecoded]
0030 f4 a2 92 af eb 00 00 c1 3f b0 77 88 00 37 a9 84 .....?..7..
0040 99 bd 8c 5b 1d f4 6b eb 24 61 58 43 c5 83 e8 9b ...[.k.$aXc...
0050 42 89 0a 2d 44 fc 63 cb da 0b bc fb 47 a7 4c 96 B..D.c...G.L
0060 41 f5 82 6b 4f b9 0f 8b 6e dd a3 8d 39 8a f3 5f A..k0...n..9..

```

Figure 7. Undecoded information of OICQ packets

When checking the "other data" part in the inspection window of Wireshark, all the data, no matter what "command number" it is being responsible for, is undecodeable by Wireshark. In order to decrypt these data, by referencing a research project [24] that has analysed the encryption mechanism used by few other IM applications, including Facebook IM and Yahoo IM, by using other digital forensics tools to acquire the messenger database files that are stored in the device itself, the messages may be able to be decrypted, but considering that these operations have go beyond the purpose of our research, further attempts to decrypt those QQ messages are not being conducted.

4. FUTURE WORK

Due to the rapid development and evolution of the IM software on the market, not only does more IM software or platforms that have different characteristics keep being introduced, but some existing products have also developed more variations of the software. Including QQ and the developers from Tencent, who have also produced other variations of IM software which are focused on different functionalities. For example, TIM [25] which uses the same account system as QQ, is more focused on team discussions for the company and enterprise users.

In addition to this, QQ also provided software versions for Linux, macOS, Android and iOS platforms. Experiment results from this project may be able to provide constructive suggestions to other projects that are focused on the digital forensics related to these variations of QQ, but different techniques may be required. Use mobile Android version of QQ as an example, an Android virtual device and other related SDK may be required when trying to capture and analyse the data packets via the Android operating system [26]. Meanwhile, there is also a related study by Hao's team [27] that has provided a relatively different research and experiment frameworks about the digital forensics of the IM applications on Android devices. Considering the noticeable difference between the operating system structures of the mobile Android and Windows 10, the encryption method or the structure of the data packets generated by QQ software for these two platforms may be different.

Another application that is also worth mentioning is WeChat [28], which is also developed by Tencent and is more well known to overseas users. Especially since, unlike QQ, WeChat continues to support the progressive web application (PWA) version of the software [29], which allows users who do not have the WeChat software pre-installed on their devices to easily log into the account. Considering the difference between the native software application and the PWA [30], protocols that the PWA version of WeChat may implement different data packet exchange protocols. While Kali Linux has also a set of tools that provide packet analysis testing targeting the PWA [31], these tools may be able to help conduct digital forensics targeting the PWA version of WeChat.

Similarly, in one of the studies that is referenced by this research [2] have analysed the VoIP protocol that is being used by the voice call functionality of WhatsApp. The research framework and experiment design from this study may provide useful suggestions in further studies that are focused on analysing other functionalities of the QQ software.

As previously mentioned, QQ uses HTTP and HTTPS to fetch certain assets related to the user account, and these assets require extra cookies handlers to fetch. Considering that Wireshark can also capture the cookies information used by these data packets, by using this cookies information, in cooperation with the

crawler techniques [32], forensic engineers may be able to use the fetched asset URIs and the corresponding cookies information to fetch more useful information about the target QQ accounts.

A bonus proposal worth mentioning is that there is also a powerful tool called *Burpsuite*, which is focused on web application penetration and forensics, is also provided in Kali Linux. This tool is capable of implementing *cookies* to fetch additional data from HTTP requests [31], [32]. Meanwhile, although the previously mentioned experiment results have suggested that part of the critical information that is related to the QQ user accounts is being encrypted by QQ software using the TEA algorithm, when analysing the structure of the OICQ protocol, a small amount of unencrypted information is still being able to extract. For example, the OICQ (QQ account) number is still decodable by Wireshark. For real-life digital forensics or man-in-the-middle attack cases, the digital forensics engineers or the attackers can still see this part of the data and fetch the QQ account number of the target. This potential problem is crucial, especially when considering that the QQ account number can be treated as one's private personal information. To increase the security level of the QQ software, the OICQ protocol may require more complete encryption for other parts of the data that it contains.

5. CONCLUSIONS

In this project through the analysis of QQ data packets the general characteristics of the data packet structure and activities that are used by the QQ software were also discovered. This includes the basic communication method and OICQ protocol that QQ primarily uses. Although the result that the experiment of this project may yield is still very limited, the authors believe that the discussion of the research model of this project and the preliminary result that the experiment outlines may provide useful suggestions or research directions for other related research, especially for those digital forensics or cybersecurity projects that also target QQ or other variants of the QQ software. This project has only examined the text messaging application of the QQ software. In practice, QQ users may choose to communicate with their friends via voice or video call. Meanwhile, QQ has also provided other functionalities such as file sharing and video conferencing. The protocol that these applications use has not yet been examined.





REFERENCES

- [1] N. Kumari and A. Mohapatra, "An insight into digital forensics branches and tools," in *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, pp. 243–250, 2016. doi: 10.1109/ICCTICT.2016.7514586.
- [2] W. Ahmed, F. Shahzad, A. R. Javed, F. Iqbal, and L. Ali, "WhatsApp network forensics: discovering the IP addresses of suspects," in *2021 11th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2021*, pp. 1–7, 2021. doi: 10.1109/NTMS49979.2021.9432677.
- [3] F.-C. Tsai, E.-C. Chang, and D.-Y. Kao, "WhatsApp network forensics: Discovering the communication payloads behind cybercriminals," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, vol. 2018-Febru, pp. 679–684, 2018. doi: 10.23919/ICACT.2018.8323882.
- [4] R. Huang, H. Kim, and J. Kim, "Social capital in QQ China: Impacts on virtual engagement of information seeking, interaction sharing, knowledge creating, and purchasing intention," *Journal of Marketing Management*, vol. 29, no. 3–4, pp. 292–316, 2013. doi: 10.1080/0267257X.2013.766630.
- [5] C. Xiaoxian, Y. Zhenlong, and X. Yibo, "The mechanism of tencent QQ video communication," in *Proceedings of the The 1st International Workshop on Cloud Computing and Information Security*, vol. 52, 2013. doi: 10.2991/ccis-13.2013.63.
- [6] "Explanation of PC version of QQ protocol, perfectly handle the QQ smart agents (in Chinese)," 2019, Accessed: April 25, 2022, [Online]. Available: <https://www.cnblogs.com/raorao1994/p/10861856.html>.
- [7] "OICQ protocol analysis, analysis the behavior of QQ software using Wireshark (in Chinese)," 2020, Accessed: April 25, 2022, [Online]. Available: https://blog.csdn.net/weixin_39791653/article/details/110897456.
- [8] A. Nath, *Packet Analysis with Wireshark. Livery Place*, Brimingham, UK: Packt Publishing Ltd., 2017.
- [9] Piyush. Verma, *Wireshark Network Security*, vol. 1, no. 1. Packt Publishing Ltd., 2015.
- [10] J. Singh, S. Dhariwal and R. Kumar, "A detailed survey of ARP poisoning detection and mitigation techniques," *International Journal of Control Theory and Applications*, vol. 9, no. 4, pp. 131-137, 2016.
- [11] V. Ramachandran and S. Nandi, "Detecting ARP spoofing: an active technique," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3803 LNCS, pp. 239–250, 2005, doi: 10.1007/11593980_18.
- [12] H. Wolf and B. Weaver, *Kali Linux 2018: Windows Penetration Testing Second Edition*. O'Reilly, 2018.
- [13] "Official website QQ for Windows (in Chinese)," Accessed: April 25, 2022, [Online]. Available: <http://im.qq.com/index.html>.
- [14] "Tencent QQ for Deepin," Community Discussion Deepin Technology, 2012, Accessed: April 25, 2022, [Online]. Available: <https://bbs.deepin.org/post/108049>.
- [15] L. Rui, M. Jia, and H. Bo, "Design and implementation of instant messenger security monitoring system based on protocol analysis," in *2010 Chinese Control and Decision Conference, CCDC 2010*, pp. 4290–4293, 2010. doi: 10.1109/CCDC.2010.5498371.
- [16] Official website of QQ for Linux (in Chinese), Accessed: April 25, 2022, [Online]. Available: <https://im.qq.com/linuxqq/index.html>.
- [17] F. Vedder, "Tencent: a case study on expanding through micro-innovation and strategic partnerships," in *Multinational Management, Cham: Springer International Publishing*, pp. 111–130, 2016.
- [18] D. P. Bovet and M. Cesati, *Understanding the Linux Kernel*, 3rd Edition, O'Reilly, 2005.





- [19] Deepin-wine for Ubuntu and Debian (in Chinese), 2020, Accessed: April 25, 2022, [Online]. Available: <https://github.com/zq1997/deepin-wine>.
- [20] A. Adekotujo, A. Odumabo, A. Adedokun, and O. Aiyeniko, "A comparative study of operating systems: case of windows, UNIX, Linux, Mac, Android and iOS," *International Journal of Computer Applications*, vol. 176, no. 39, pp. 16–23, 2020, doi: 10.5120/ijca2020920494.
- [21] "List of the top software downloads (in Chinese)," mydown.yesky.com, 2022, Accessed: April 25, 2022, [Online]. Available: <https://mydown.yesky.com/rank/>.
- [22] J. Bullock, *Wireshark® for Security Professionals*. Indianapolis, Indiana: John Wiley & Sons, Inc., 2017.
- [23] "In-depth explanation of QQ protocol part one (in Chinese)," Jun. 2020, Accessed: April 25, 2022, [Online]. Available: <https://cloud.tencent.com/developer/article/1644053>.
- [24] N. B. Al Barghuthi and H. Said, "Social networks IM forensics: encryption analysis," *Journal of Communications*, vol. 8, no. 11, pp. 708–715, 2013, doi: 10.12720/jcm.8.11.708-715.
- [25] "TIM, focused on team communication and cooperation (in Chinese)," office.qq.com, Accessed: April 25, 2022, [Online]. Available: <https://office.qq.com>.
- [26] Sirinivas, "Android forensics labs," INFOSEC, Apr. 2016, Accessed: April 25, 2022, [Online]. Available: <https://resources.infosecinstitute.com/topic/android-forensics-labs>.
- [27] H. Zhang, L. Chen, and Q. Liu, "Digital forensic analysis of instant messaging applications on android smartphones," in 2018 *International Conference on Computing, Networking and Communications (ICNC)*, pp. 647–651, 2018. doi: 10.1109/ICCNC.2018.8390330.
- [28] D. W. Anderson, *Connecting a billion people with calls, chats, and more*. Wechat Navigator, Cozhan, 2019.
- [29] W. Chen, H. Lu, M. Li, and Y. Sun, "Network protocol analysis base on WeChat PC version," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11635 LNCS, pp. 288–297, 2019, doi: 10.1007/978-3-030-24268-8_27.
- [30] O. Korzachenko and K. Cherniavskyi, "Applications: revolutionary changes in web development," *Modeling and Information Systems in Economics*, no. 99, pp. 92–101, Nov. 2020, doi: 10.33111/mise.99.8.
- [31] "Kali Linux tutorials - A single stop for Kali Linux tools," Accessed: April 25, 2022, [Online]. Available: <https://kalilinuxtutorials.com/>.
- [32] Official website of Burp Suite, "Manually setting a cookie for Burp's Crawl and Audit," PortSwigger, 2022, Accessed: April 25, 2022, [Online]. Available: <https://portswigger.net/support/manually-setting-a-cookie-for-burp-suites-crawl-and-audit>.

BIOGRAPHIES OF AUTHORS



Yunkun Li     received his bachelor's degrees in Information and Computer Science, from the University of Liverpool (UK) and Xi'an Jiaotong-Liverpool University (China) in July 2021. His research interests include information security, cybersecurity, and some of the interdisciplinary fields joint together with techniques of computer science. He can be contacted at email: Yunkun.Li18@student.xjtlu.edu.cn.



Gabriela Mogos     is Associate Professor in the Department of Computing, School of Advanced Technology, at the Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China. She received her PhD in Computer Science from the Alexandru Ioan Cuza University of Iasi, Romania, and she followed this with a postdoctoral research position at the University of Oradea, Romania. Her interests and research activities include Cybersecurity, Quantum computing with an emphasis on design of new quantum algorithms and quantum cryptographic protocols. She has more than 100 academic publications, books, book chapters, and has acted as principal investigator and co-investigator in international and national research projects. She can be contacted at email: Gabriela.Mogos@xjtlu.edu.cn.