Facebook Forensics in Different Systems

EN 650.757.01 Advanced Computer Forensics

Ruiyang Liu

Abstract

Facebook, was founded in 2004 by Mark Zuckerberg, and over 1.62 billion users visited Facebook daily in 2019. The research focuses on the differences in forensics against Facebook under different systems. The study of Facebook forensics includes user information, chat function, post and comment function. In this paper, we will introduce the forensic tools used to make Facebook forensic snapshot disk file and memory dump process, as well as the forensic process and methods. We will also compare the similarities and differences of disk file and memory dump of three different systems in the above three main information forensics.

1 Introduction

1.1 Research Background

Facebook, was founded in 2004 by Mark Zuckerberg while he was studying at Harvard, is operated and managed by Facebook Inc. now.[1] Over 1.62 billion users visited Facebook daily in 2019.[2] The sheer online ubiquity of Facebook is astounding. As of February 2012, Facebook had over 845 million users (more than the population of Europe) who spent more than 9.7 billion minutes per day on the site. Users share four billion pieces of content per day, including uploads of 250 million photos, and Facebook is now integrated with over seven million websites and applications. Most of the Facebook users join groups and making post. These group posts may contain illegitimate information and being harmful to other users. What's more, among 12- to 17-year old teen, approximately 40% of them are using Facebook. With so many people using Facebook, criminals may easily hide their real identity. The study of Facebook forensics can help reduce information leakage, online crime, etc.

1.2 Research Aims and Methods

The research will analyze and study facebook forensic evidence on three different systems: Linux, Windows 7 and Android mobile. The research will analyze and study facebook forensic evidence on three different systems: Linux OS, Windows 7 OS and Android OS. Experiment environment should be set up properly, which is foundation to start the research. Our

[1] Phillips, Sarah. "A Brief History of Facebook." The Guardian, Guardian News and Media, 25 July 2007, www.theguardian.com/technology/2007/jul/25/media.newmedia

[2] Aboulhosn, Sarah. "18 Facebook Statistics Every Marketer Should Know in 2020." Sprout Social, 21 Jan. 2020, www.sproutsocial.com/insights/facebook-stats-for-marketers/

investigation in this experiment has started the moment we installed the applications or browsed the Facebook web on the systems. We have also created two dummy accounts for the experiments(main account *Ruiyang Liu* and helper account *Rayon Liu*). All detail steps, credentials and communication activities were properly documented. After all the prerequisite are well managed, being able to know what kinds of the Facebook activities and evidences existed are the key to start the forensic progress. After snapshotted the VMs, we can get the virtual disk files whose extensions are .vdi., and the dumps of volatile state information whose extensions are .sav. The analysis were conducted manually by searching several keyword related to experiment in forensically sounds manner. Finally, using image reader to access and analyze the dumps is the research final aim.

This research project identifies the following Facebook activity footprints:

1) Facebook login and account information

2) Facebook friends information

3) Facebook chatting

4) Facebook post and reply to the post

This research includes facebook memory footprints searching and extracting:

1) Web browser cache file

2) Linux OS dump

3) Windows 7 OS dump

4) Android OS dump

1.3 Research Forensics Tools

This research used the following software or forensic tools:

1) Hxd Hex Editor: Hxd Hex Editor is a multi-functional hexadecimal editor viewer. It also supports binary file viewing. Editing and creating ASCII, hexadecimal, decimal, float, double and binary data is also supported. Besides, researchers can make file patches, searching and replacing code through Hxd Hex Editor.

2) Virtual Box Windows 7 Virtual Machine: Windows 7 is one of the most famous operating system developed by Microsoft. Considering that there will be interference from other software when running this research directly on the PC, an absolutely new Windows 7 virtual machine is used for the experiment.

3) ChromeCacheView: ChromeCacheView is a cache data cloning tool that can copy the data you cached on your browser to a specified folder and save it. ChromeCacheView can intercept all cached data on the Chrome browser, including information such as access address, content, and timestamps. In this research, using ChromeCacheView to analyze the Facebook web forensic evidence on Windows 11 computer.

4) FTK imager: FTK® Imager is a data preview and imaging tool that lets you quickly assess electronic evidence to determine if further analysis with a forensic tool such as Forensic Toolkit (FTK®) is warranted. In this research, we use both Hxd Hex Editor and FTK imager to gather information of the dumps. However, I suggest to use HxdHexEditor as the primary tool, since it process lookups quicker.

5) Virtual Box Linux Virtual Machine: Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17,

1991, by Linus Torvalds. Linux is one of the most prominent examples of free and open-source software collaboration.

6) Virtual Box Android 9.0 Virtual Machine: Android is a mobile OS based on a modified version of the Linux kernel and other open source software, designed primarily for mobile devices. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. Analyzing Facebook evidence on Android OS system is critical because most Facebook users chat and post using their mobile phone.

7) Plist Editor: Plist Editor, is a file editing tool that helps users edit Plist files, can run under the Windows7 OS. The Plist Editor can edit Plist files in XML format and binary format, and supports color-coding, replacement, and undo of the code in the file.

2 Facebook Protocol Format in Browser Cache

Before locating any Facebook evidence on VMs, we need to know the format of Facebook protocol that may appear in VM dumps or browser cache. Therefore, we attempted to identify the protocol format of Facebook post, comment, message and chat located in Windows 11 machine browser cache. In the following analysis, two Facebook accounts have been set up for performing Facebook activities. 1094635582@qq.com (User name *Ruiyang Liu*) is the tester account responsible for wall posting, commenting, messaging and chatting on his own, and testfbforensic@yahoo.com(User name *Rayon Liu*) is the helper account responsible for replying to the post and chatting with the tester. Snapshots and dumps of tester's virtual machine status was taken after finishing any Facebook activities. The whole acquisition process was repeated

twice for consistency concern. All files and information in this part is gathered through ChromeCacheView.

Prerequisite:

- Windows PC or VM
- Chrome Web Browse
- ChromeCacheView V2.00

Research Steps: (Recommended to clear browser cache each time you do operation.)

i. Open Chrome Web Browser and surf www.facebook.com

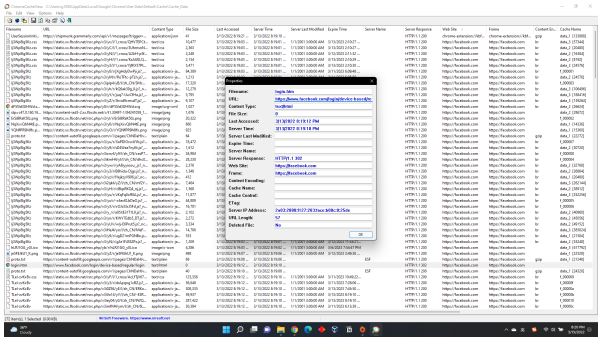
ii. Enter your facebook username and password to login

iii. Send a chatting message to your facebook friend

iv. Post something and reply

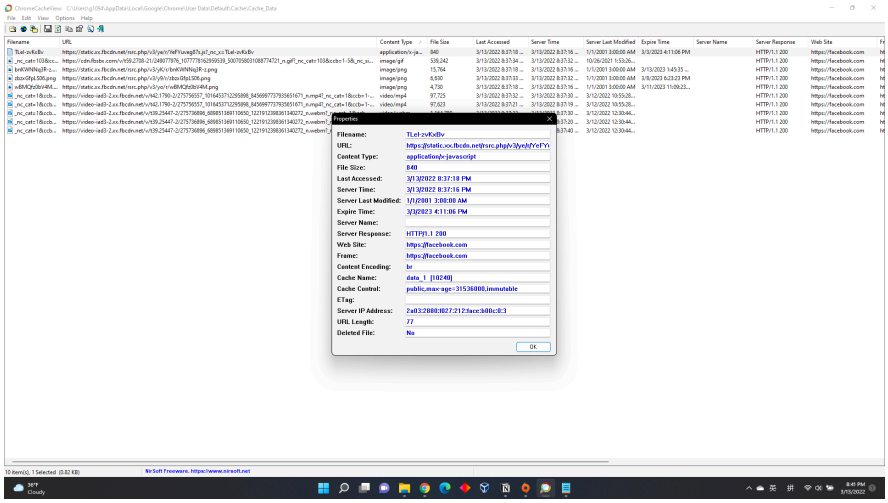
v. Turn on the ChromeCacheView to check the Chrome cache

When we login to Facebook in Chrome, we are able to get login.html. By checking the web browser cache, we can find out all the login or logout activity details, including timestamps and server IP address.

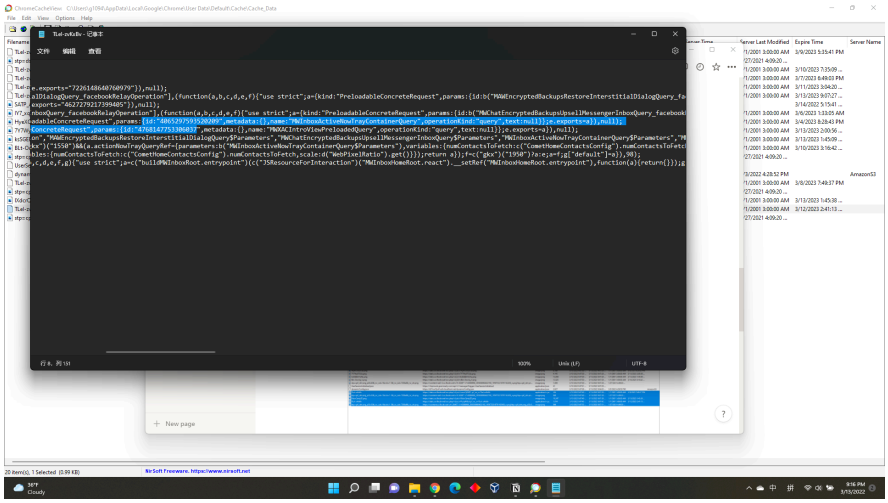


Login info in chrome cache

Next, we send messages to Facebook friends. We are able to get accountID when we look at the cache packets.

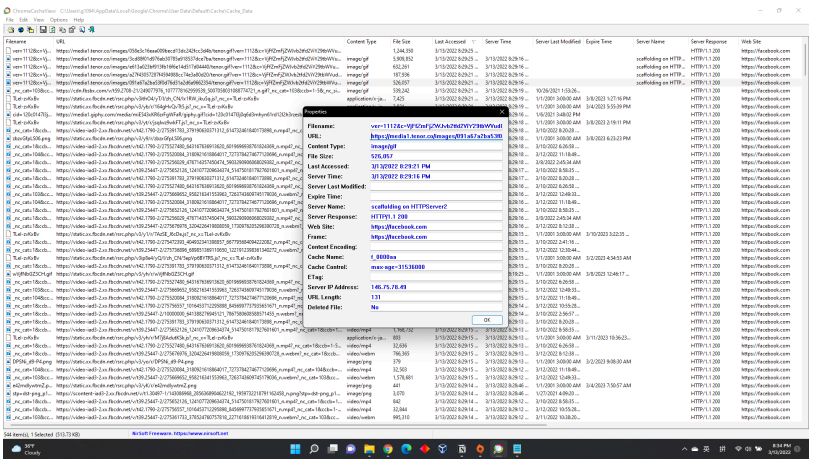


Chat Packet in Chrome Cache



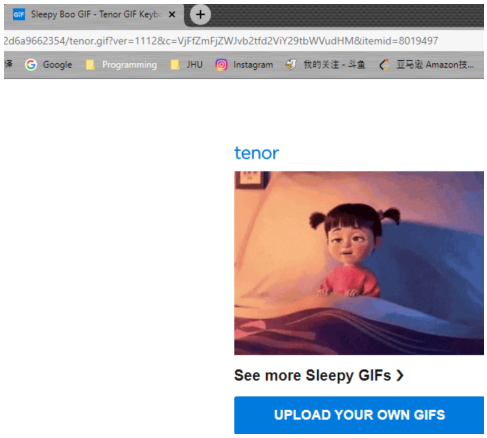
Chat Packet detail

Further, when we send memes to Facebook friends, we can get the meme image url in the cache.



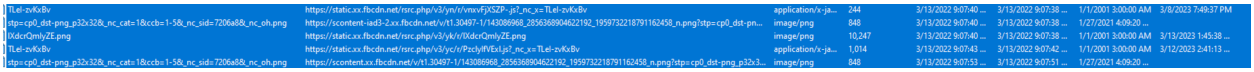
Chat Gif Meme packet

Chat meme in FB

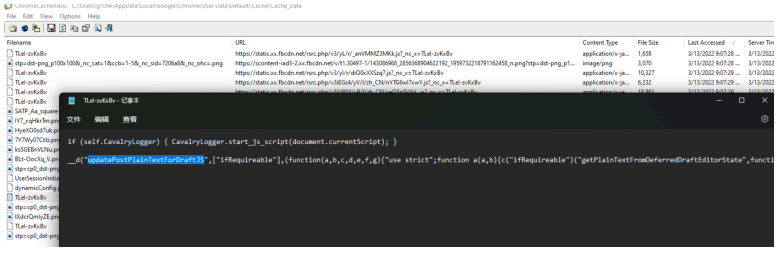


Cache Packet URL corresponds

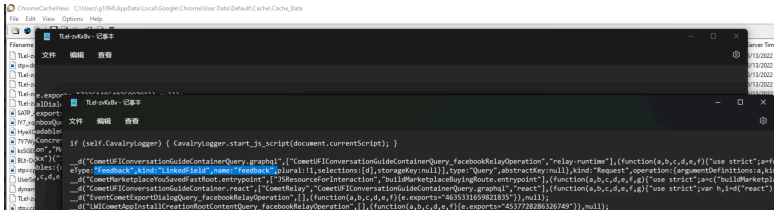
Finally, the test account post something, and the helper account reply to the post. In the cache packets, we are able to see the post and reply activities.



Post and Reply Packets



Post Cache Packet



Reply(Feedback) Cache Packet

3 Facebook Forensics on Linux VM

Prerequisite:

- Linux VM in Virtual Box
- Hdx Hex Editor
- FTK Imager (Optional)
- Chrome Web Browser in the VM

VM Dump Creation Steps:

I. Set up a Linux VM and then install Chrome in the VM

II. Open Chrome Web Browser and surf www.facebook.com

III. Enter your facebook username and password to login

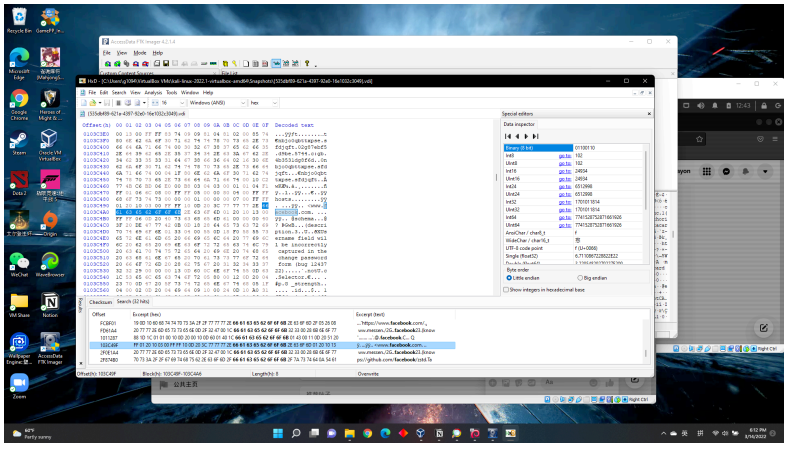
IV. Send a chatting message to one of your facebook friends

V. Post a feed and reply to itself, reply to helper account's post, and helper reply to tester's post.

Facebook Evidence in Linux Virtual Disk Snapshot File (.vdi)

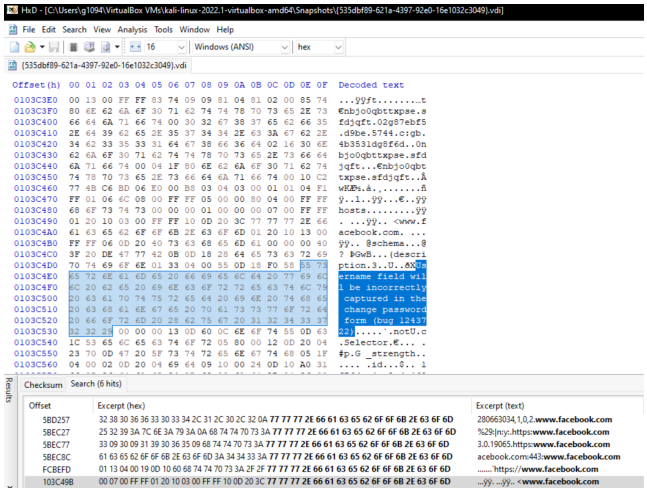
The analysis were conducted manually by searching several keyword.

First, we search for keyword *Facebook* and Facebook URL. We can see multiple results and their excerpt(hex) is the same.



Searching *Facebook* keyword found in Linux vdi file

Secondly, we search for Facebook URL (www.facebook.com). And we can see the information about I changed the password for my test account.



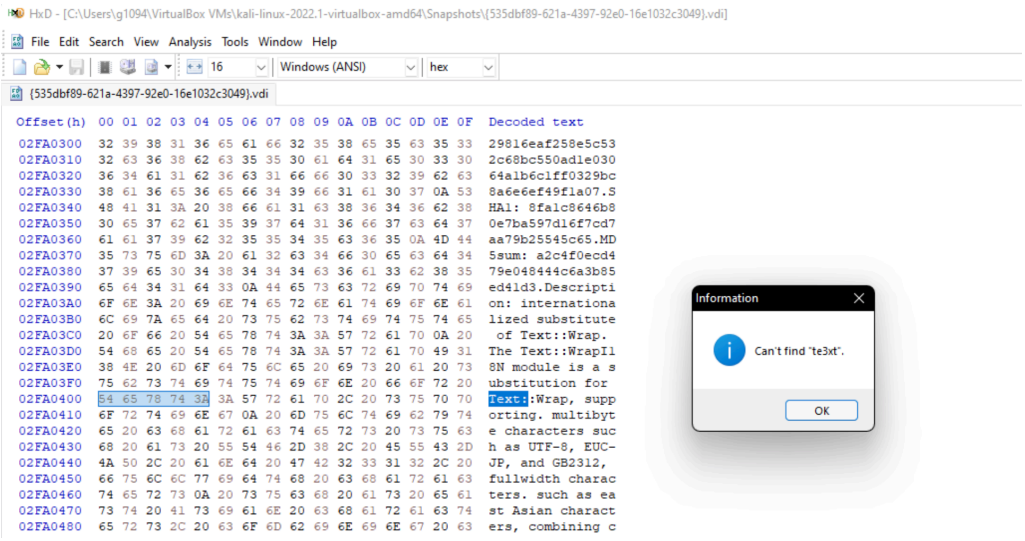
Searching *Facebook URL* keyword found in Linux vdi file

The third search were to find our profile name or user ID once we logged into the social media. Profile name is different with username. Username used to login into the social media while profile name were the name displayed in our social media account. When I want to search *Liu* as my last name of my Facebook username or Ruiyang as my first name of my Facebook username, I get nothing in Virtual disk snapshot file. As a result, you cannot find Facebook username in VM .vdi.

The Fourth search were to find our login email information. The keyword I choose to search my login email is the partial email. To be more specific, my test account is 1094635582@qq.com, I try to search *1094* or *qq.com*. But we also cannot get any login email information in Virtual disk snapshot file.

The fifth search is the friend's account name and our chatting message. I tried to search *text* or specific chat message like *te3xt*. We still cannot find any information.

Facebook Chat message on Linux VM



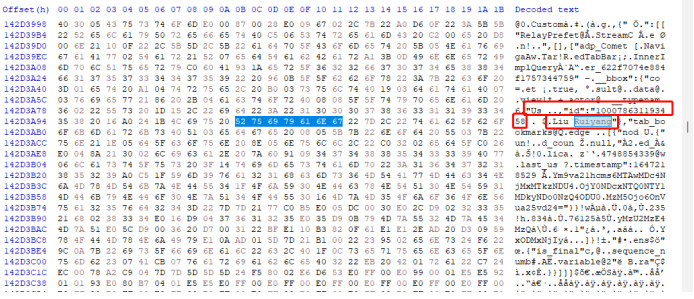
Searching *text_message* keyword not found in Linux vdi file

Facebook Evidence in Linux Memory Dump file (.sav)

User, Login and Friend Evidence in Linux Memory Dump

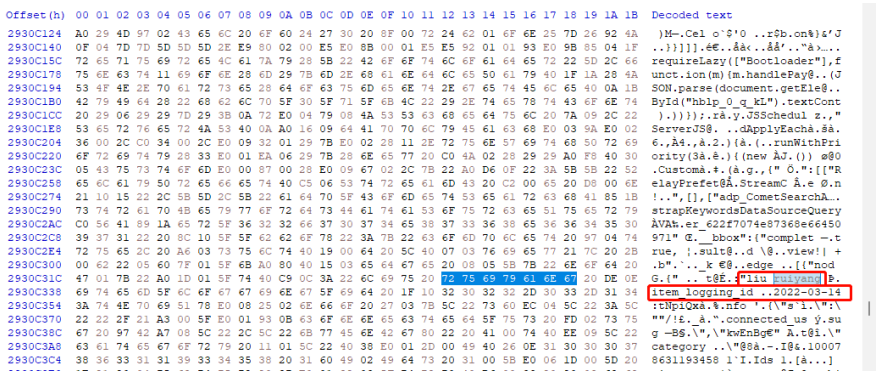
We use the same steps and keywords as previously done in Linux virtual disk snapshot file.

First, we search my account name, and we can get UserID information and my username.



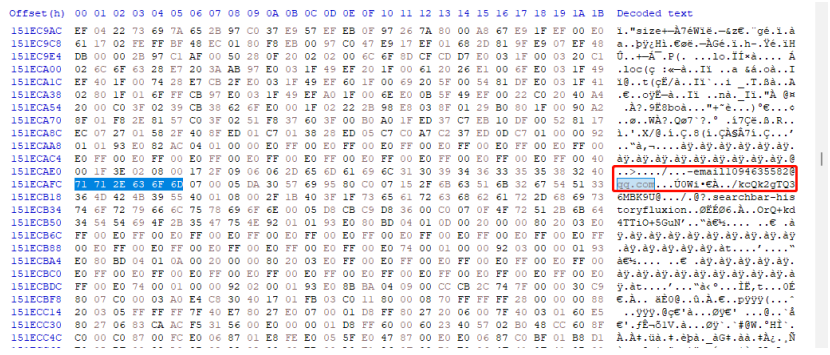
Searching username keyword and find userID in Linux memory dump

Also, as we search the key word of the username, we can find Facebook login information. The login information includes username and the login time.



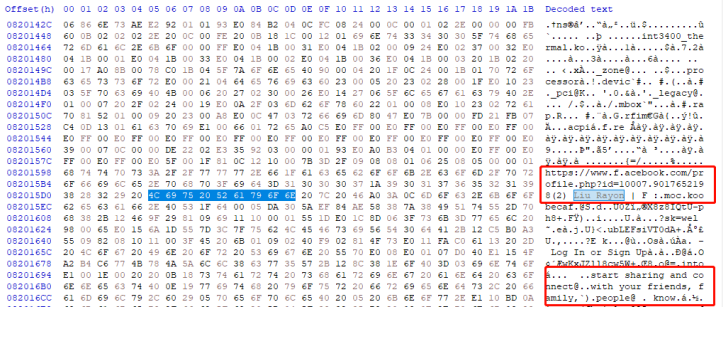
Searching username and get login info in Linux memory dump

Second, we try to find more user information by searching login email. It is certain that this email address is the Facebook information, since this email address is the unique email for Facebook account. Though we are able to find login email in the memory, we cannot find other information through this evidence.

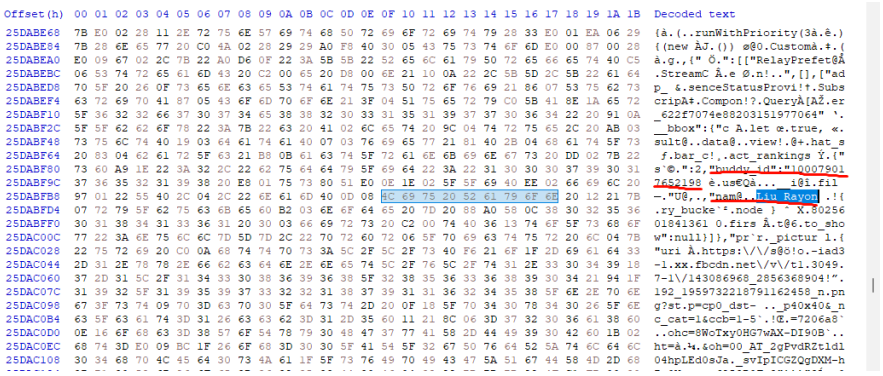


Searching login email in Linux memory dump

Third, we use Facebook friend's username keyword. In Linux, we found facebook friend URL, i.e. php website link. And also, we can find friend userID using friend's username as keyword.



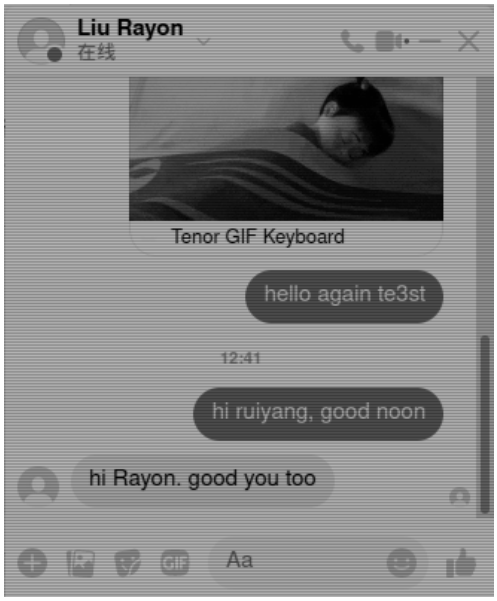
Searching friend username in Linux memory dump



Searching friend username and find his userID in Linux memory dump

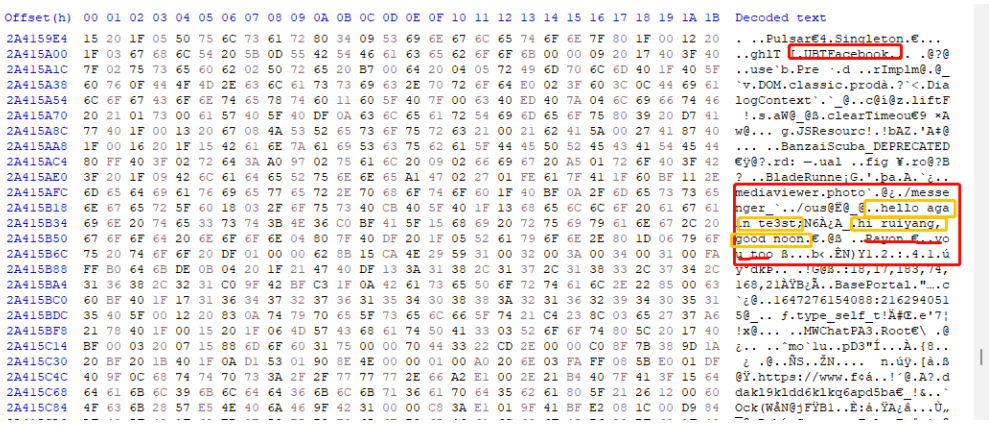
Chat Evidence in Linux Memory Dump

Forth, the main Facebook account *Ruiyang* chat with the helper account Liu Rayon, and following is the screenshot of our chat.

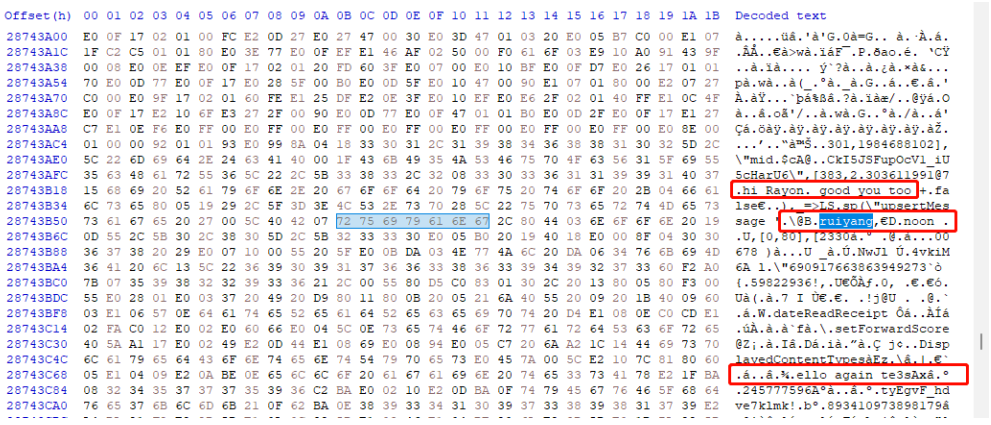


Screenshot of chatting in Linux

In the ram reader, we search chatting keywords, such as *good noon*, *te3st*, etc. In the Linux memory dump, I find multiple evidences. They appear multiple times but I don't know the reason why. This might be my gap research regarding them appearing so much times. Following is the result of the search of chatting evidence.



Evidence1 of chatting in Linux Memory dump



Evidence 2 of chatting in Linux Memory dump

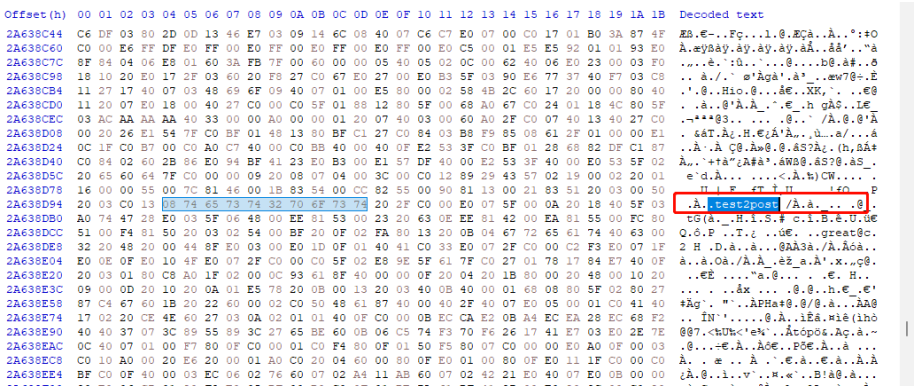
Post and Comment Evidence in Linux Memory Dump

There are some elements to take into account. First, we are trying to find evidence of a post. But if the user did not post on this Linux virtual machine, can we find evidence in the Linux Memory dump. Secondly, whether the results of the forensic evidence will be different if the user comments under his own post and under someone else's post.

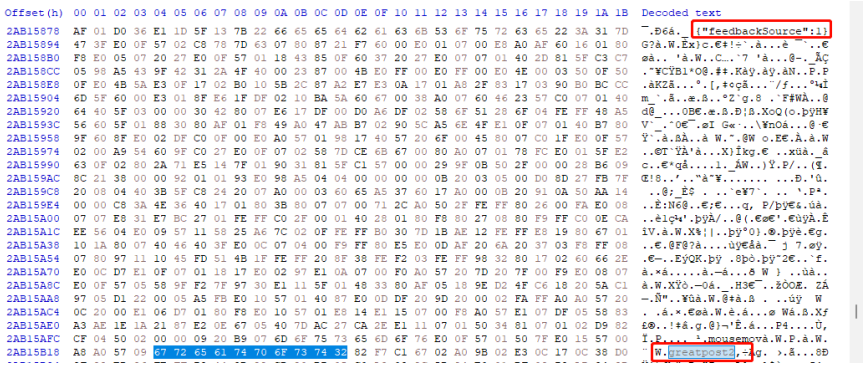
In the first step, I started by posting on a Linux virtual machine and commenting on my post.

Post and comment to myself in Linux VM

By searching the keyword, we are able to find both posting evidence and comment evidence as following.

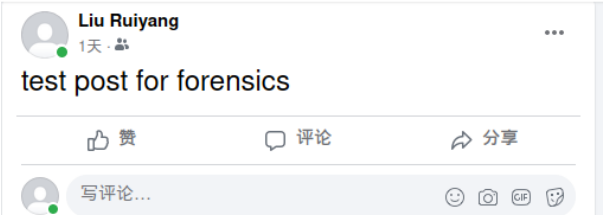


Posting evidence in Linux Memory Dump



Comment(Feedback) evidence in Linux Memory Dump

To check the whether we can get forensic evidence from posting in other devices. We first post something from my Windows local machine, as following.

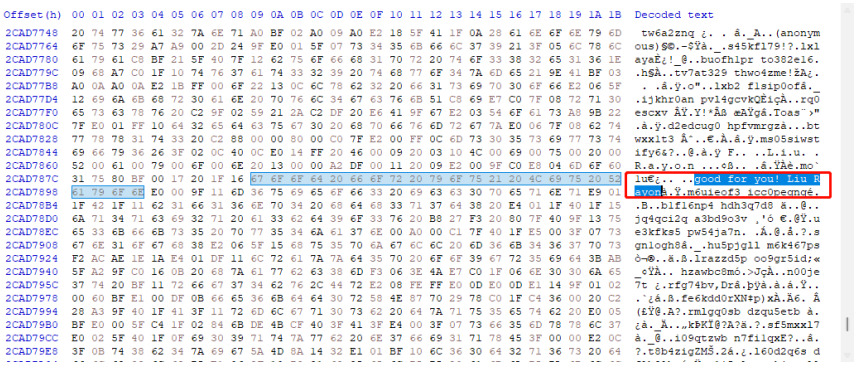


Posting from Windows local machine

I tried multiple keywords to find this post, but there's nothing in the Linux memory dump. As result, when you post from other devices, you cannot find forensic evidence in the RAM.

Second, We use helper account to post and the main account reply the post as following.

Then, we search the comment keyword try to find evidence in the Linux memory. And we can find some evidence.



Comment evidence in Linux Memory Dump

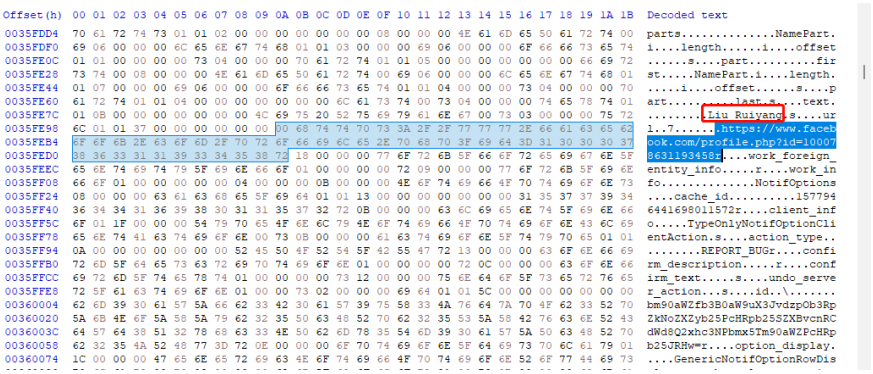
4 Facebook Forensics on Android VM

Overall, the process of finding forensic evidence is similar to previous one on Linux VM. We try to search keywords manually using hex editor.

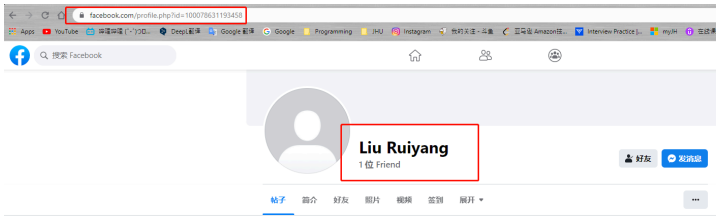
Facebook Evidence in Android Virtual Disk Snapshot File (.vdi)

First thing to notice is that much more readable comparing to Linux VM disk file. There are much less garbled code in hex editor. And this might also be my gap research, what is the reason of the amount of garbled code in the hex editor.

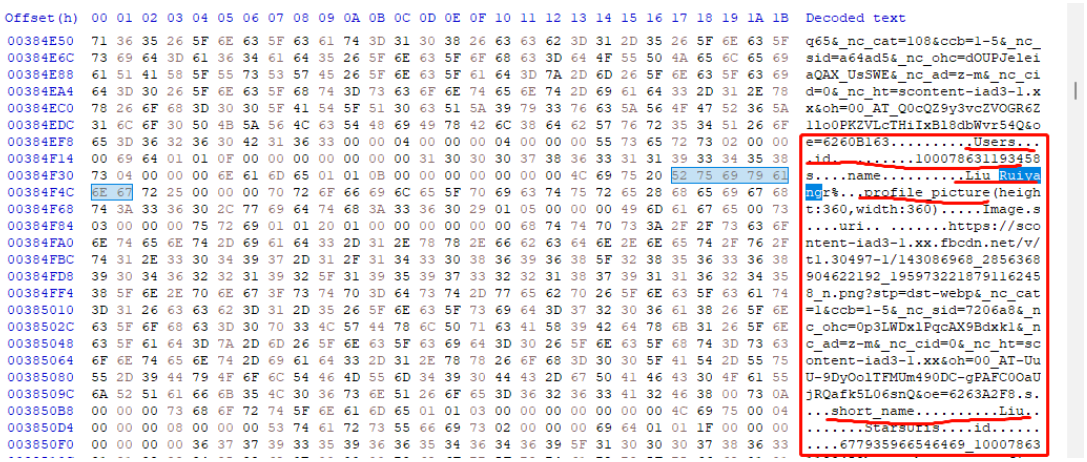
First, we use the keyword my username to try to gather user information in the Android disk file. One evidence I got is the link to my main Facebook page as following.



User page link evidence in Android disk file

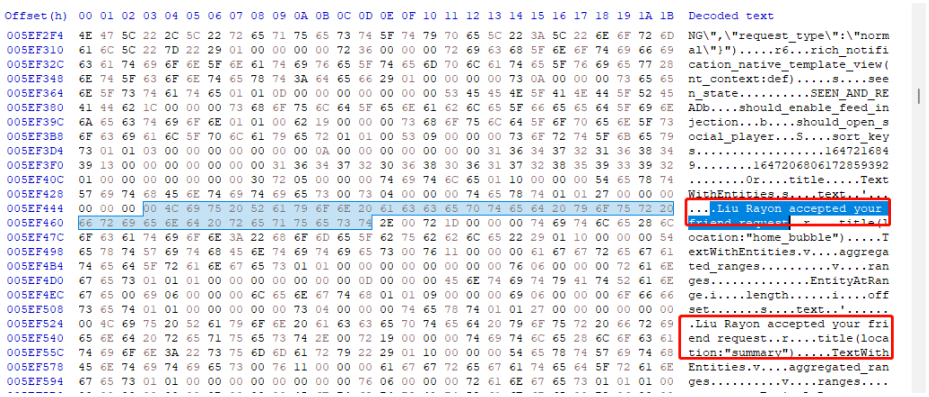


Also, we are able to get userID and other several information like profile picture size, short name, etc. in the Android disk file, as following.

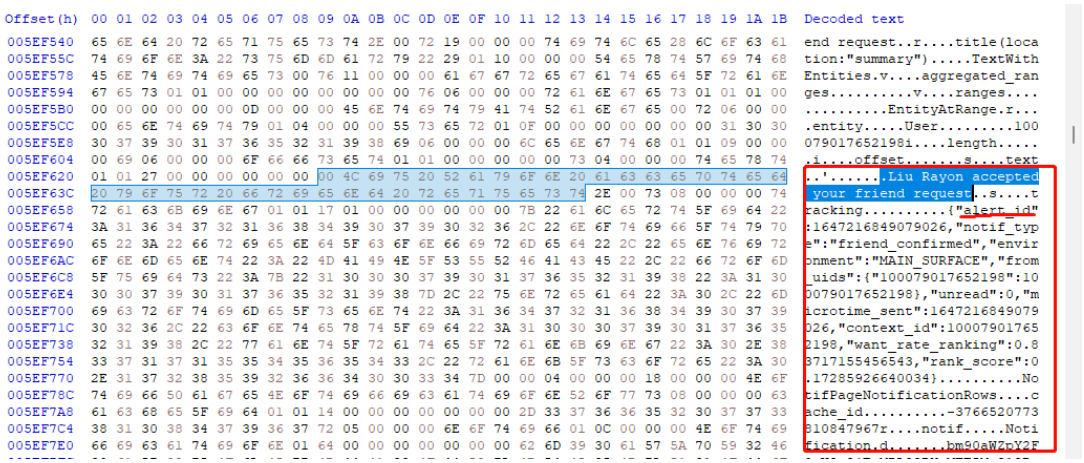


Other user information found in Android disk file

Second, we try to find friends information using friend username as keyword. One thing we can got is the Facebook website alert showing we add helper account *Rayon Liu* as our friend.

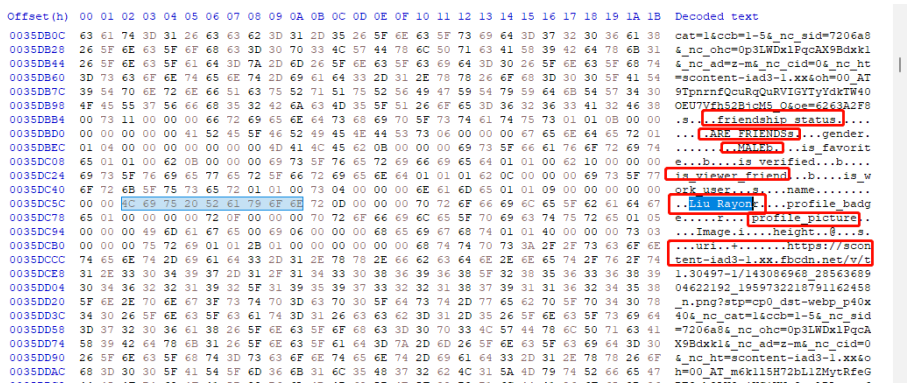


Friend Acceptance Alert found in Android disk file



Second Friend Alert found in Android disk file

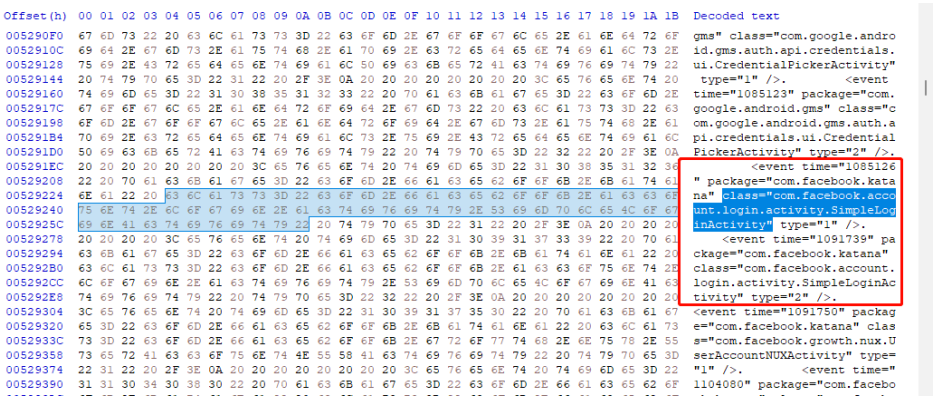
Moreover, we also can get a little friend user information in the disk file. The information is not too much, and the information is provided by a static web log. We can probably guess that the



Little friend information in Android disk file

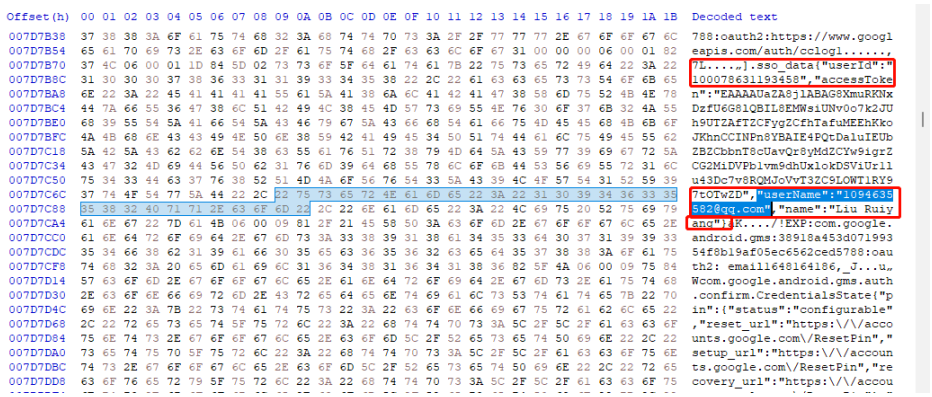
specific user information of our friend here is not saved in a file. Rather, some web logs are saved in the file, and these web caches reveal very little friend information.

Third, we try to search user login email to find some helpful information. One evidence is the login evidence. As shown in the log, we are able to identify login action and its time because of the Android application logs.



Login evidence using email keyword in Android disk file

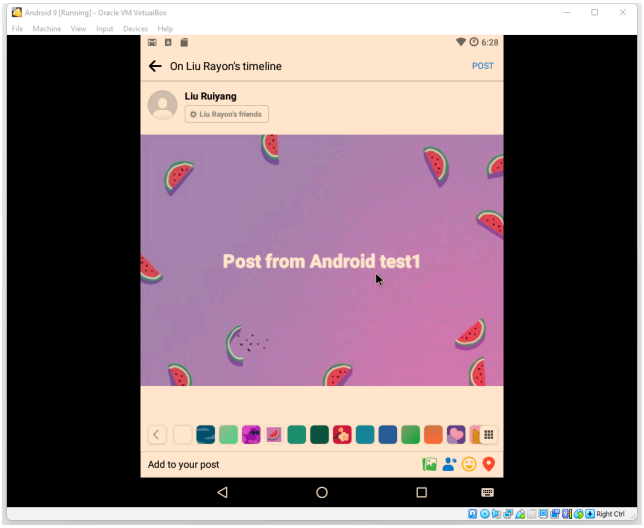
Further, the email keyword help me got the username.



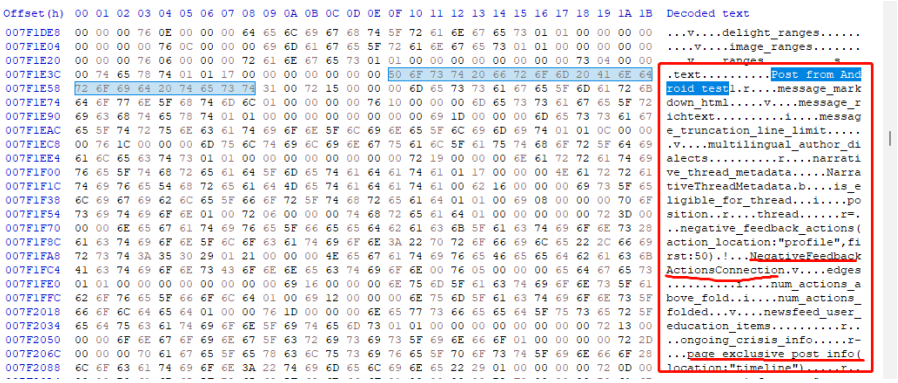
Forth, I continue the conversation with the helper account in the Android VM and search using the chat content keywords. But I am not able to find any information or content related to the chat log. I will elaborate on the specific content of the chat logs in the next analysis of Android memory dump.

Fifth, I tried to find the forensic for posted content and the commenting on the post. The results here are unusual. First the same steps as if I were doing it on a Linux VM, I first published some content with my own account and then commented on myself. At the same time, I also comment on content posted by friends or others. Finally, I try to find evidence by searching for the keywords of the published content and the keywords of the comments.

First, I can find what I have posted, but I find evidence of my comments by keyword or post evidence location.

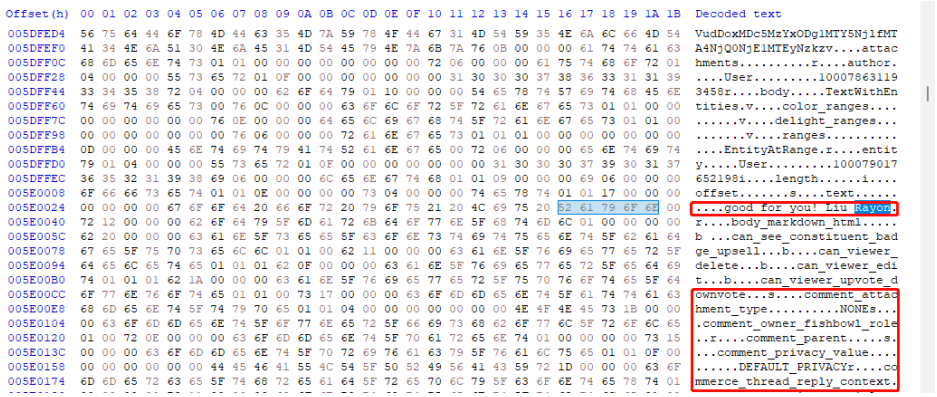


Posting using Android Facebook App in Android VM



Posting evidence found in Android disk file

Second, rather unusually, I found evidence of the comment content in the Linux VM from the previous experiment. But I did not find any comment evidence of operation in Android VM.



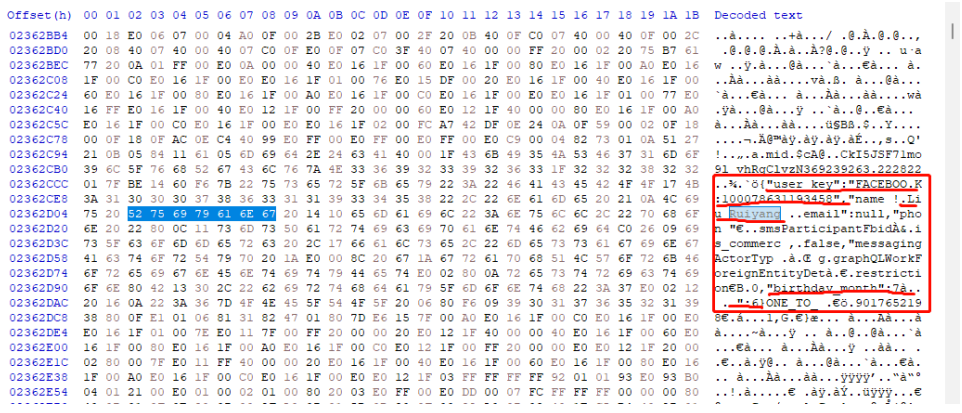
Some comment evidence found in Android disk file

Facebook Evidence in Android Memory Dump file (.sav)

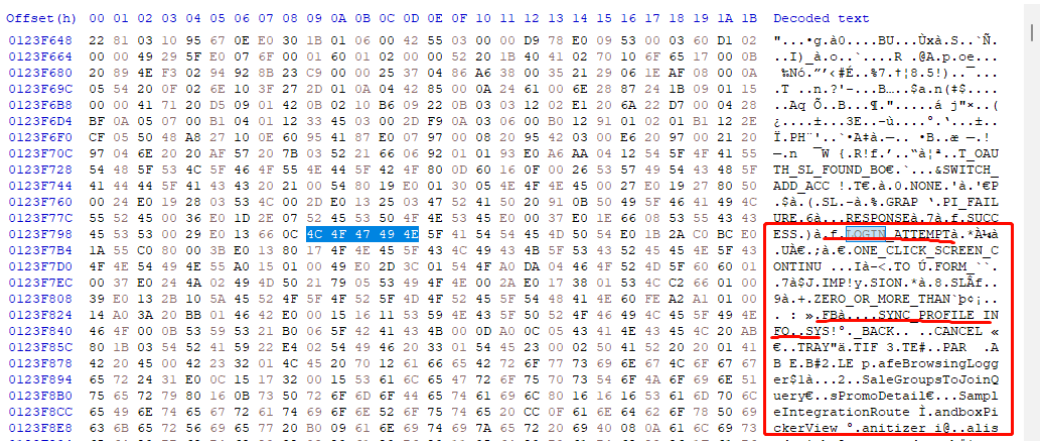
User, Login and Friend Evidence in Android Memory Dump

We use the same steps and keywords as previously done in Android memory dump file.

First, we search my account name, and we can get UserID information, the birthday, and my login attempt log.

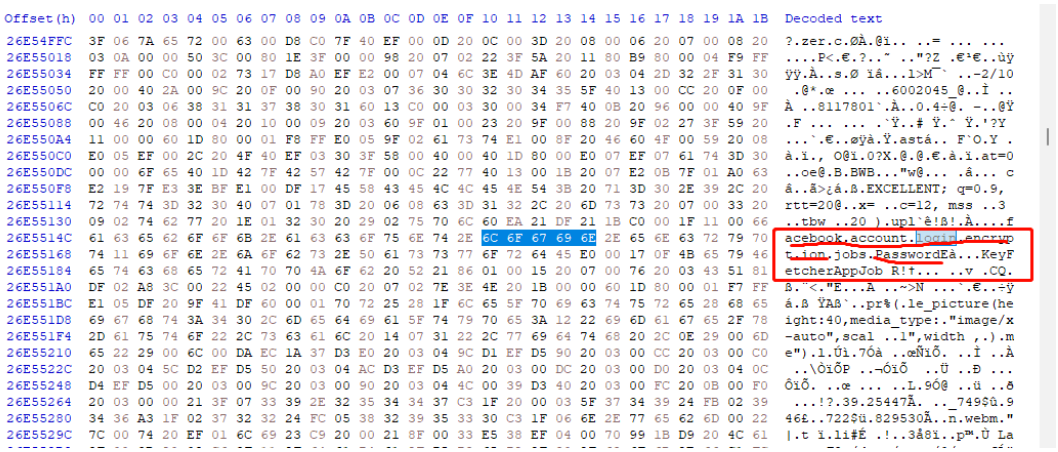


UserID and birthday information found in Android Memory Dump



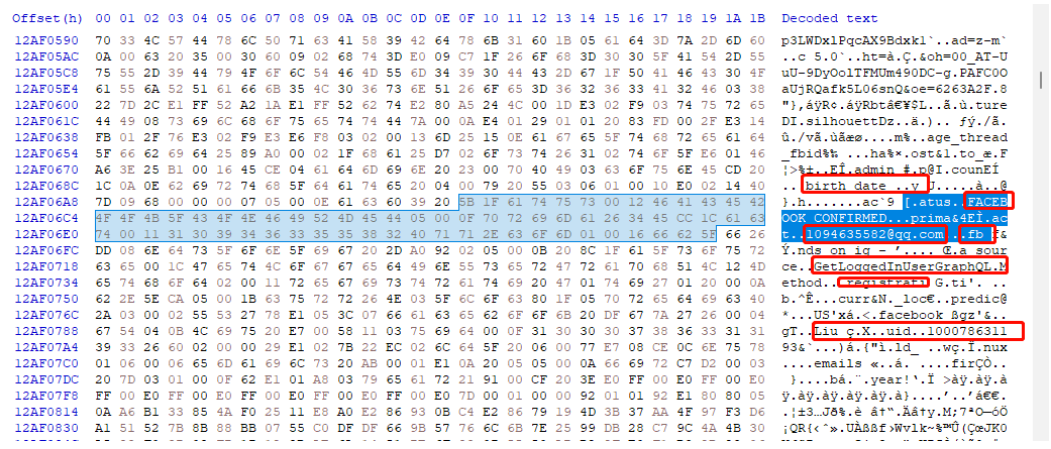
Login Attempt evidence in Android Memory Dump

Moreover, When we login to the account, we can see the login evidence show that the password of the account is being encrypted, i.e., Facebook.account.login.encryption.jobs.Password.



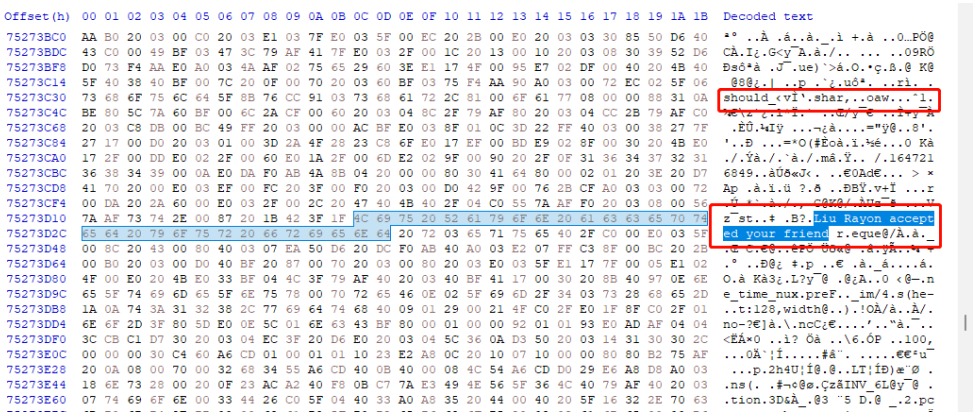
Facebook Login Encrypted Password in Android Memory Dump

Second, We search the email keyword, and get similar user information, such as user birthdate, UserUid, etc.

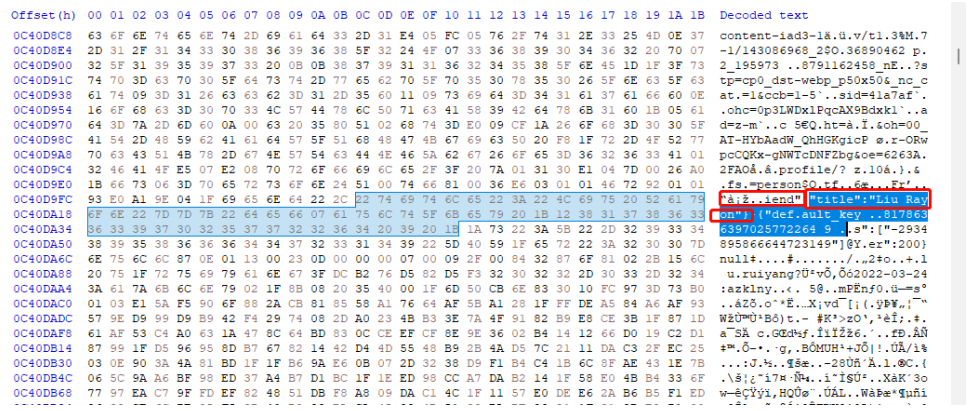


Little user information found using email keyword in Android Memory Dump

Third, we search friends keyword to find friend information. The same as we observed in Android disk file, we can find earlier friend accept alert in Windows local machine. Also, we can get little friend information, such as friend's username, friend keyID.



Friend Acceptance Alert in Android Memory Dump

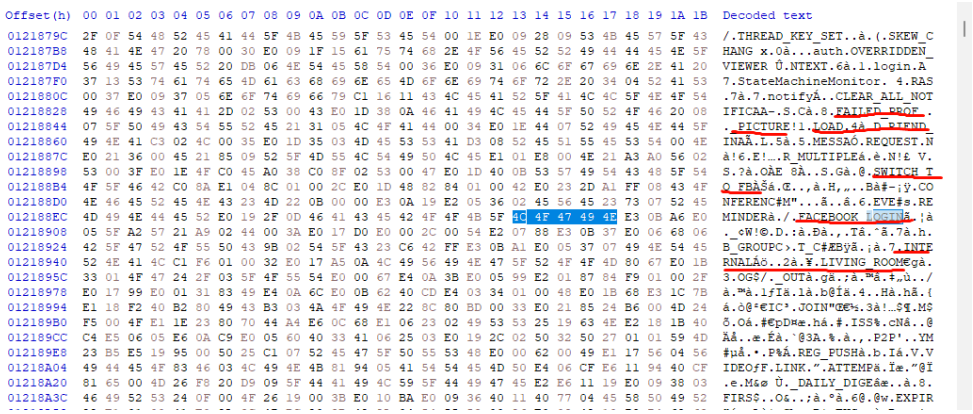


Little Friend information in Android Memory Dump

Chat Evidence in Android Memory Dump

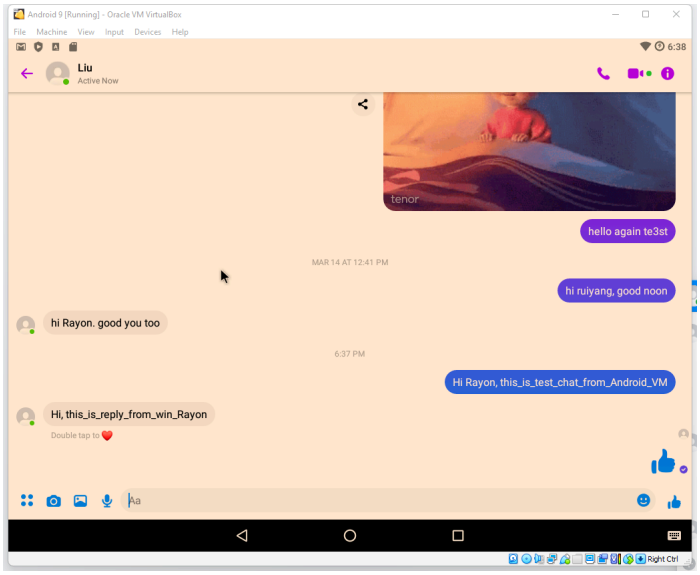
Before you can do Facebook chat in Android Virtual Machine, you need to install Android Facebook chat software and then you can do the function of chatting with your friends. After installing the chat software, we need to authorize the Facebook account login feature. After

authorization, you will be able to chat with your friends. When we search for evidence of login, we also find forensic evidence of authorized login to Facebook chat application.

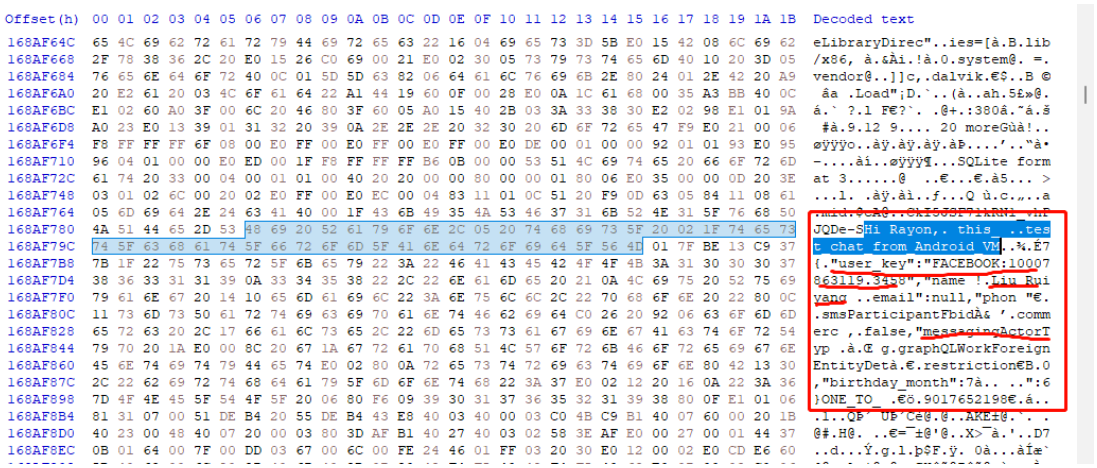


Facebook Chat Application Authorized Login Found in Android Memory Dump

After we installed the chat application, we chat with helper account as the same procedure as we do in Linux VM.



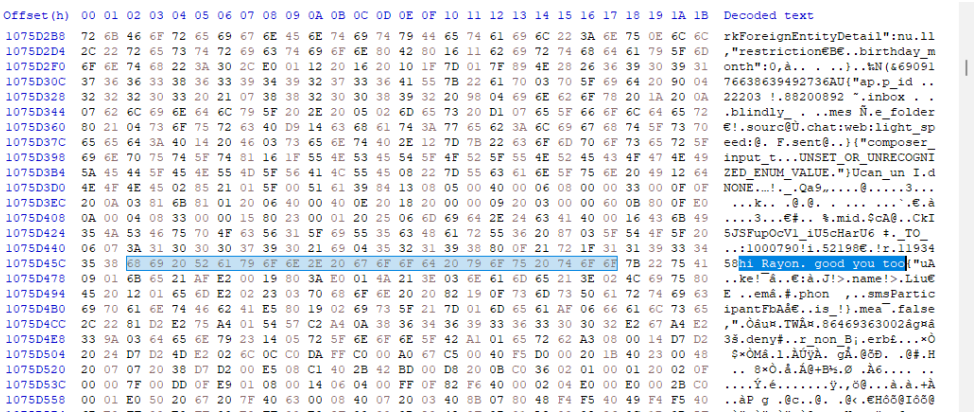
In Android memory dump, we are able to gather both chat message we sent and the message our friend sent to me. Also, we can get additional information, such as userKey, username and the *messageActorType* to inform us that the log is a Facebook message evidence.



Message I sent to my friend evidence in Android Memory Dump

Message friend sent to me evidence in Android Memory Dump

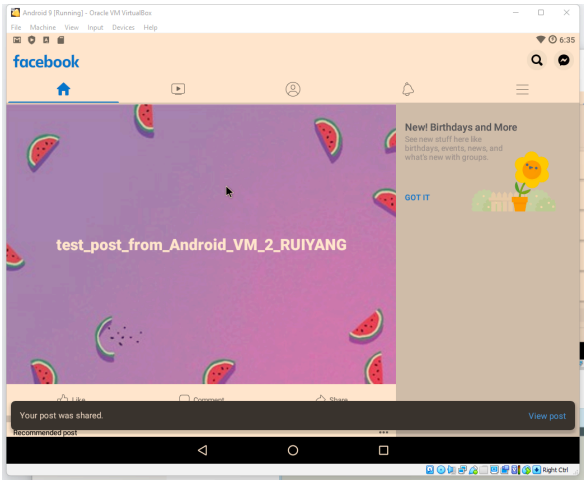
Moreover, we can even get evidence of previous chat contents in earlier experiments.



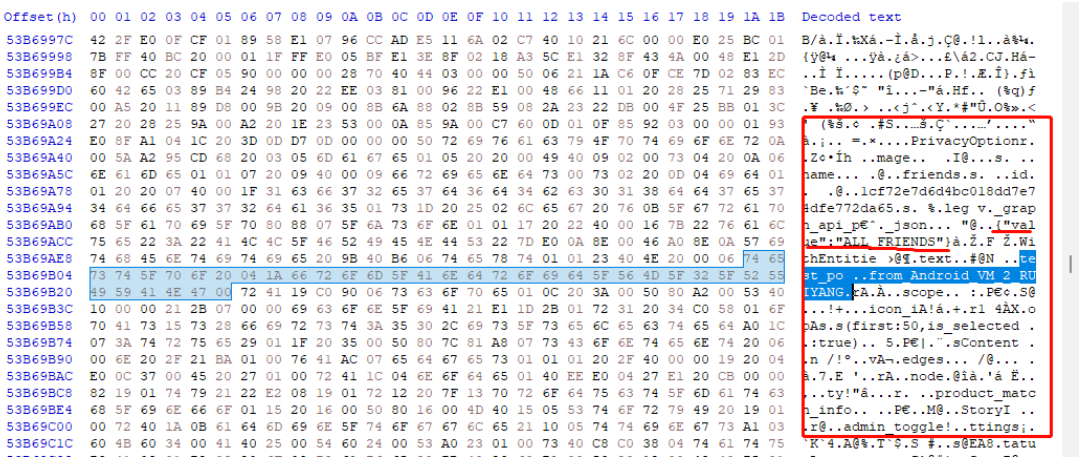
Earlier chat content found in Android Memory Dump

Post and Comment Evidence in Android Memory Dump

First, we post some content in the Android VM Facebook software. Then we search in Hex Editor using the posting content keywords. We can get the forensic evidence of our published content, and the visibility range *ALL_FRIENDS*.

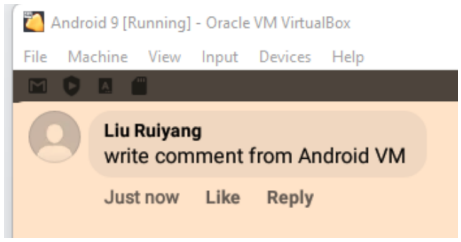


Posting from Android VM Facebook App

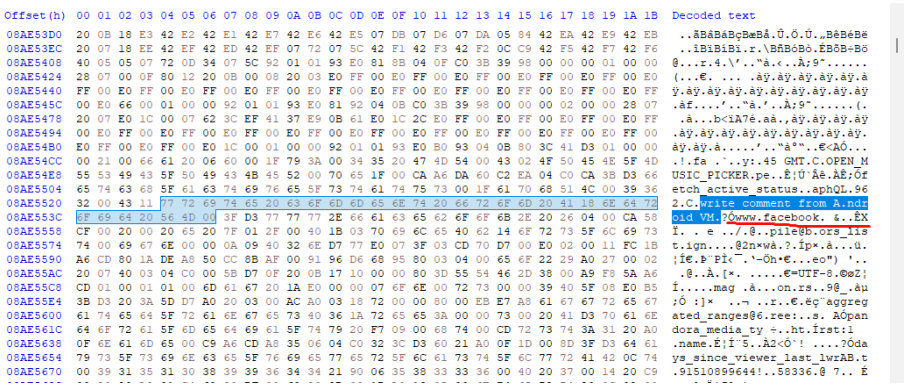


Posting Evidence in Android Memory Dump

Second, we We test whether it is possible to get forensic evidence by commenting under our own published content and commenting under other people's published content. First we tested commenting under a friend's post. We can see the content of our comment in the Android Memory dump.



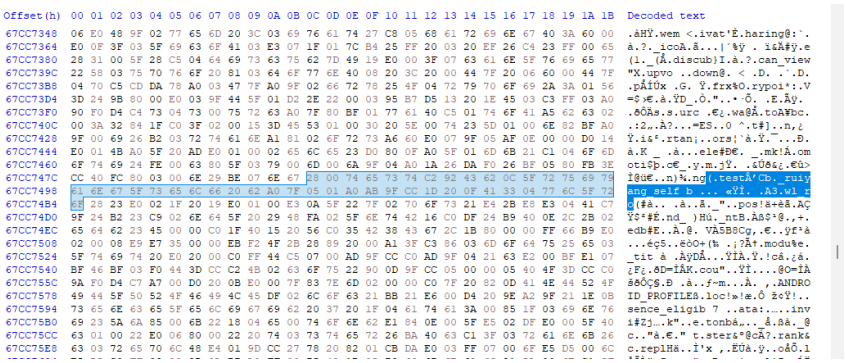
Write comment to other people's post



Comment other people's post evidence found in Android Memory Dump

Then we comment below the content we post. Strangely, we can roughly see that the forensic evidence found in Hex Editor is our comment, but that evidence is not complete (i.e., not a complete sentence, and not exactly identical to the comment). It remains unclear to me whether this is a common pattern or just a special case of my personal experiment.

Write comment to myself's post



Not complete evidence of comment in Android Memory Dump

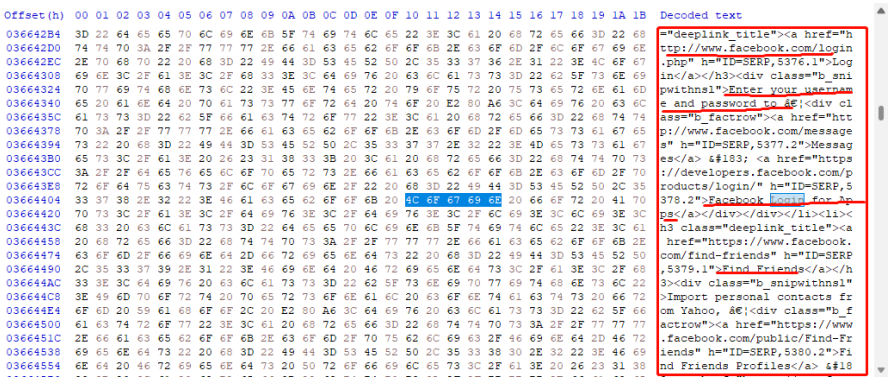
5 Facebook Forensics on Windows 7 VM

Overall, the process of finding forensic evidence is similar to previous two on Linux VM and Android VM. We try to search keywords manually using hex editor.

Facebook Evidence in Windows7 Virtual Disk Snapshot File (.vdi)

Compared to the first two operating systems, you can see a lot of web information and web cache style content in the disk file of windows 7. But compared to the first two operating systems, we get less information in the disk file.

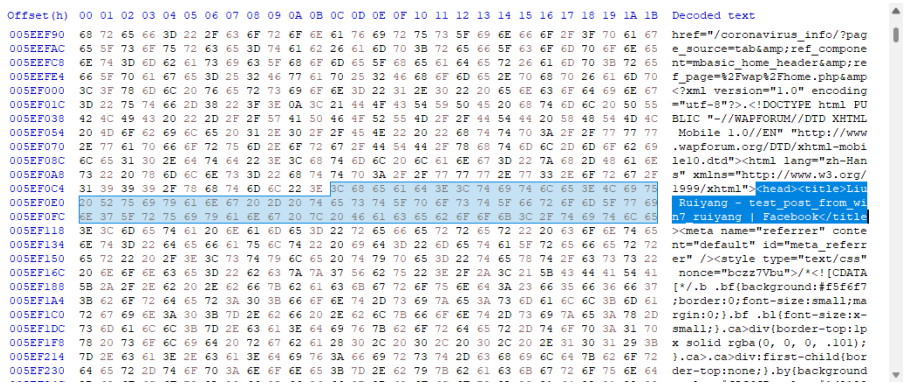
First we still try to retrieve user information using username keywords, email keywords and login keywords first. We can get the information in the form of web HTML displayed directly in Hex Editor, which is something that neither Linux nor Android systems see. We can see the information of our login screen, including the user trying to login and requesting a password. But we do not see further information about the user's personal information. And I didn't find information about login email in the Win7 disk file.



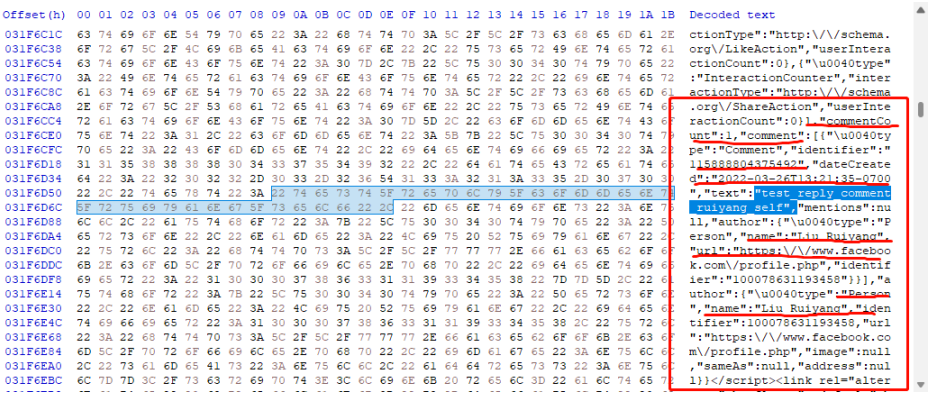
Login page HTML in win7 disk file

Secondly, we chatted with our Facebook friends in Win7 virtual machine, but we did not find any evidence of chat by searching chat content keywords or chat user name keywords.

Third, we post some content in Win7 Virtual Machine and then reply to what we have posted. We can find the posted content and the evidence of our comments on our own posted content in the disk file.



Post evidence in Win7 Disk File



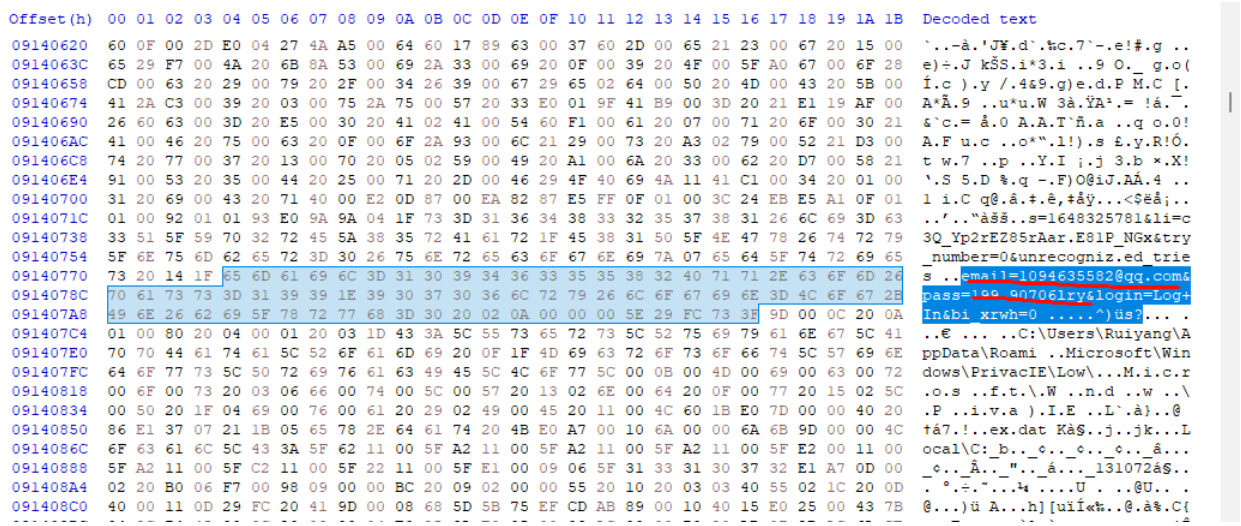
Comment myself evidence in Win7 Disk File

Finally, we tried to comment on content posted by other users as well as our own friends, but I didn't find any forensic evidence.

Facebook Evidence in Win7 Memory Dump file (.sav)

User, Login and Friend Evidence in Win7 Memory Dump

As in the previous two system experiments, I tried to use username keywords, email keywords to search for login and user information. Surprisingly, in the win7 virtual machine, the user login and login password are displayed directly in the win7 memory dump without encryption. In other words, one can get the Facebook login password directly in the win7 memory dump.

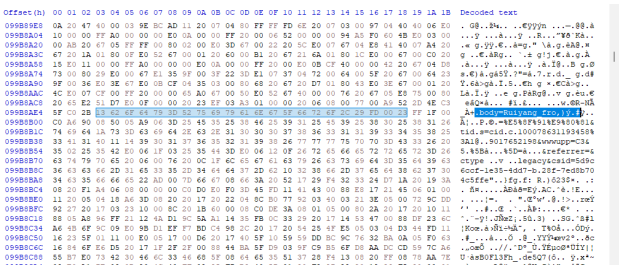


User Login info and Password shows in the Win7 Memory Dump

But in Win7 memory dump, I tried to search for friend username and other keywords and didn't find friend information.

Chat Evidence in Android Memory Dump

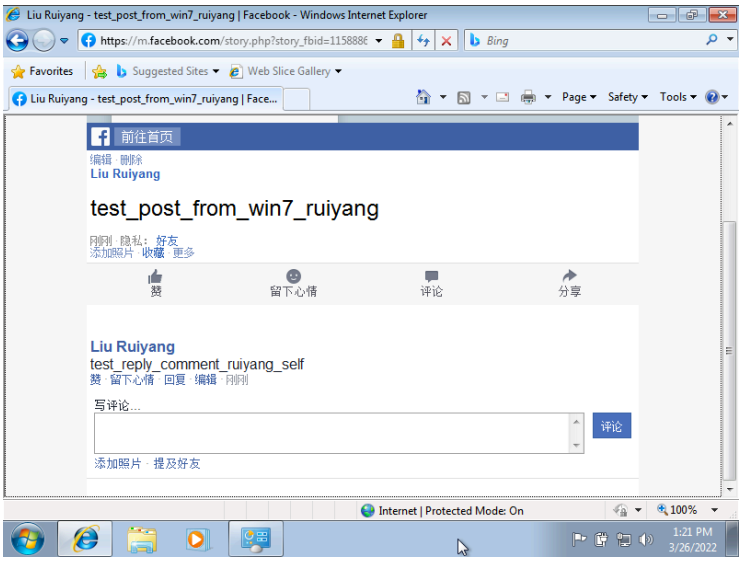
First, you need to talk to your friend in the web page of Win7 Virtual Machine. Then try to search the chat content keywords in Hex editor to try to find the chat forensic evidence. But I can only find a few uncertain words in the Win7 memory dump of the chats I sent, and I can't find any of the chats that the helper account replied to me.



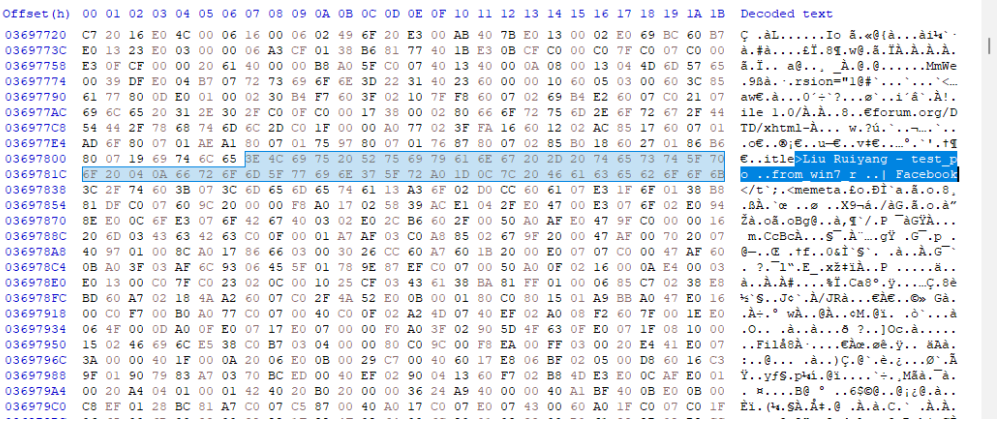
Uncertain chat evidence in Win7 Memory Dump

Post and Comment Evidence in Android Memory Dump

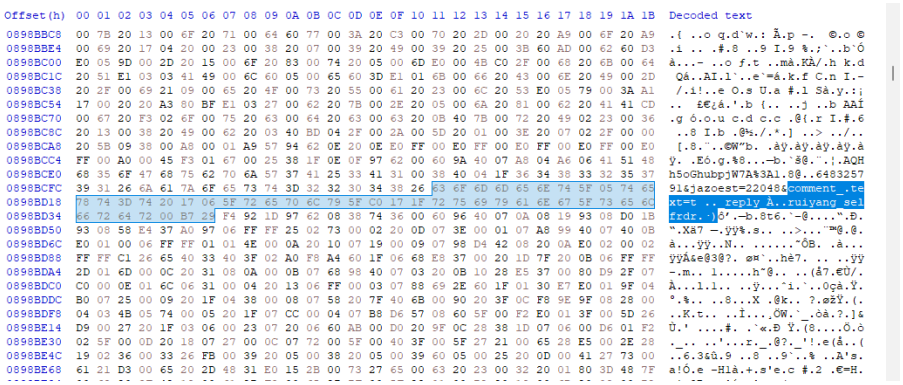
As the same experiment in the first two systems, I first made a post in the Win7 VM and then commented on myself. Then I look for post evidence and comment evidence in Hex editor.



Post and comment to myself in Win7 VM



Post evidence found in Win7 Memory Dump



Comment evidence found in Win7 Memory Dump

6 Conclusion

My research focuses on the differences in forensics against Facebook under different systems. First, I logged in to my Facebook account, chatted with friends, posted content, and commented on other people's posted content under Linux, Android, and Win7 respectively. Then I need to use tools like Hex Editor or FTK to read the disk file and memory dump, and finally use manual search for some relevant keywords to get forensic evidence. Next I will discuss the similarities and differences in the evidence between the three systems.

First, in terms of user information, user login information, and user friend information, the Linux disk file can search for Facebook keywords, but there is no valid user information, user login emails, or friend information revealed. In the Linux memory dump, we can get login information, login time, user ID and friend ID information, but I was not able to get more details about the user. In Android disk file, the readability is higher than Linux, we can get the link of Facebook user's homepage link, user login information, user ID and friend's log of accepting

friend request prompt. In Android memory dump, we can get more information, such as user ID, user birthday, user login request log, user login encrypted password, friend request consent log and so on. Also, we can see the log information that Facebook authorized its chat software to log in. In Win7 disk file, its readability is also better and there is a lot of HTML cached inside the disk file. But we can get very little information about the user, I couldn't find the user ID and the email keyword used to log in. In Win7 memory dump, we can see not only the login request and login email, but even the password used by the user to login in plain text. But other user information and friend information is not found in the Win7 memory dump.

Second, the chat with friends feature is a very important feature of Facebook, so the differences in the three different systems where evidence of information can be obtained are discussed next. In the Linux disk file, no forensic evidence of the chat content was found. In the Linux memory dump, we can get the content of the chats we sent and the content of the chats our friends sent back. In the Android disk file, we can not only see the content of the chat between the two parties, but we can also get the username and user ID of the person we are chatting with, and surprisingly, we can even see evidence of previous chats in the Linux system. It is assumed that the previous chat logs were cached in memory when Android loaded the previous chat logs. In the Win7 disk file, the same as the Linux disk file, no chat-related evidence was found. In the Win7 memory dump, only an incomplete sentence of the sent chat log was obtained, and no evidence of any received chat logs could be seen. So it is not sure how much chat logs can be obtained in Win7 memory dump and its chatting storage logic.

Finally, the post and comment functions are the most basic features of Facebook, and in each system I have posted some content. And in terms of comments, I commented both under what I posted and under what other users posted, as a way to see if they would make a difference in terms of forensics. No evidence of post or comment was found in the Linux disk file. In the Linux memory dump, it is possible to get content posted by yourself in Linux, comments on yourself and comments on others. But there is no evidence of posts in other systems, even though the Linux virtual machine is running. In the Android disk file, the forensic evidence of the post can be obtained, but no evidence related to the comments can be found. In Android memory dump, not only can we get the evidence of the posted content, we can also see the visible range of this post. In the comment forensic evidence, I only found one vague comment evidence (the evidence is not a complete sentence). In the Win7 disk file, we can find evidence of both the post and the comment. In Win7 memory dump, which has almost the same form of evidence as its disk file, we can get evidence of both post and comment.

As you can see, the three systems have different forensic evidence that can be obtained at the three main levels and the readability of the content seen through the hex editor is not the same. Below is my table of whether forensic evidence can be obtained for different files with different features.

	Facebook and URL	Username	Friend account	Email	Login	Post	Comment	Chat
Linux VDI	Yes	No	No	No	No	No	No	No
Android VDI	Yes	Yes	Yes	Yes	Yes	Uncertain	Partial	No
Win7 VDI	Yes	Html cache	No	No	Yes	Yes	Yes	No
Linux Memory Dump	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Android Memory Dump	Yes	Yes (Password encrypted)	Yes	Yes (Password encrypted)	Yes	Yes	Yes	Yes
Win7 Memory Dump	Yes	Yes (Shows passwor d)	Yes (Not so much info)	No	Yes	Yes	Yes	Uncertain(Not complete)

7 Future Work

Due to time constraints, the study has many questions that remain to be addressed.

1. Why the readability of disk file and memory dump of the three systems are very different and what causes it.

2. Why many messages appear many times, for example, a chat content may appear many times in different places in the memory dump, what is the reason.

3. Is there a more comprehensive and efficient way to find evidence of Facebook forensics other than using tools like Hex editor to search for keywords manually.