Memory Forensics for QQ from a Live System

Yuhang Gao, Tianjie Cao
School of Computer, China University of Mining and Technology
Sanhuannanlu, Xuzhou, Jiangsu, 221116, China
Email:rainman1919@gmail.com, tjcao@cumt.edu.cn

***Abstract*—Our paper details the techniques to collect sensitive information of the QQ client, which is the most popular instant messaging (IM) in China. We have managed to acquire the contact list, the QQ account, the chats records, the QQ discussion group, the display names and the contents of network notepad. They are of great interest to the examiners. Besides, as the techniques we use to search for process are able to reveal terminated and hidden processes, we are very likely to find sensitive information as long as somebody has logged in the QQ client. What's more, we propose the method of reconstructing the process space by integrating paging file into memory dump file. We have reconstructed the process space of the QQ client in this way and managed to narrow down the scale of sensitive information about QQ.**

***Index Terms*—instant messaging; the QQ client; memory forensics; Microsoft Windows; memory analysis;**

I. INTRODUCTION

There are papers which deal with artefacts left by popular instant messaging clients, such as MSN [1], Yahoo [2], AOL [3], Trillian [4], and Windows Live Messenger [5], Pidgin [6]. These papers pay attention to files on the disk and hope to collect sensitive information via file carving. These papers all list the traces on the hard disk after instant messaging usage, such as contact files, log files and so on. In fact, there's still an instant messaging client that has not been described named Tencent QQ.

The Tencent QQ, which is commonly referred as QQ, is the most popularly used instant messaging in China. It's calculated that there are over three thousand and four hundred million active users nowadays. Besides, the concurrent users of QQ are over thirty million people.

This paper focuses on the Tencent QQ whose version is 8.0 (build 8.0.714.1791). The described results in this paper may differ from new versions of Tencent QQ.

Different from other instant messaging, QQ has its own features. For example, the MSN Messenger sends plaintexts on the network when two users are chatting. However, the QQ prefers to encrypt the message rather than transferring it immediately. Therefore, it may be fairly easy to find sent and received messages of MSN by searching for 'X-MMS-IM-format' on the hared disk, which details the formatting options for the content of the messages. We are not likely to get the contents of chat records from QQ in the same way.

Besides, QQ promises good security on individual privacy by encrypting. All files related to privacy are encrypted and not readable without a successful login. Even if we are able to acquire these files, it takes great efforts to parse them. However, we can collect sensitive information from the physical memory. The QQ client would decrypt all received messages. Therefore, the physical memory would contain the plaintext of these messages.

Thus it can be seen that the forensics for the QQ client is different from other instant messaging. The examiners need new techniques. It's more effective to apply memory forensics in this field. Memory forensics is aimed at analyzing the physical memory from the target computer. It is possible to extract useful message from memory, such as the mapped files [7], the opened network connection [8] and so on.

Our paper is not only a continuation of the series of papers dealing with instant messaging clients, but also an application of memory forensics in this field. In our experiments, we are able to get the contact list, the QQ account, the QQ display names, the chat records and the QQ discussion group.

The reminder of this paper is organized as follows. Section 2 details the sources of sensitive information and our proposed techniques are applicable to all these sources. Section 3 describes the related data structures about reconstructing process space. We would like to reconstruct the process space of QQ to place relevant information together. Section 4 proposes the techniques of reconstructing process space, which help the examiners to collect sensitive information of QQ in a reduced scale. Section 5 gives the results of experiments and shows the sensitive information we find.

II. LIVE MEMORY COLLECTION

A. Sources of Useful Information

For a Windows based system, if we want to get some information about the processes and threads, there are three sources available: the physical memory, the hibernation file and the pagefile.

The physical memory absolutely plays the most important role in volatile forensics, for it contains all meta-information necessary to manage the processes that are currently executed, such as _EPROCESS [9] and _ETHREAD structures. Although Windows based system implements virtual memory management, the kernel is

doi:10.4304/jcp.5.4.541-548

something so important that may never be swapped out to the hard disk. What's more, a lot of important data structures, like _EPROCESS are stored in system space. Therefore, a deep and thorough analysis of physical memory is necessary, and we can collect a lot of valuable information in this way. In fact, recent studies mainly focuse on memory forensics. Our study is aimed at searching for threads and processes from physical memory, as well as gathering and analyzing as much information as possible.

The pagefile also plays an important role in forensics. As windows based systems all implement virtual memory management, Windows memory manager provides a way for a process to allocate and use larger amounts of physical memory than can be mapped into the process virtual address space. In fact, pagefile is an important component of Windows virtual memory management. Pagefile together with physical memory constitutes the main framework of virtual memory management. For example, when free memory falls below a certain threshold or a process's memory page has not been used for a long time, an out-swap operation would take place. The swapped page would be stored in pagefile. So, pagefile is also an important source of forensics.

Hibernation file is a large hidden file located in the root of our boot driver (e.g. c:\hiberfil.sys) used for keeping all PC's memory including CPU's memory and registers upon hibernation.The hibernation file only contains committed memory. Think about this case, if a process is terminated not long ago and the related information is not overwritten, but the memory page of the related information is not commited, we are likely to be unable to extract such information.

We can make a conclusion that physical memory is the main source of volatile forensics. Most of the useful information can be found in physical memory. The methods applied in memory forensics is probably available to pagefile and hiberfile forensics.

B. The Proposed Techniques forMemory Collecting

Since memory forensics is put forward, several useful tools for physical memory dump are developed. These tools may be different in details, but they're theoretically the same. All these tools achieve the goal of directly reading physical memory by loading loadable kernel modules or gaining reading and writing accesses of \Device\PhysicalMemory. These tools maintain footprint inevitably, for they must first be loaded into the physical memory which may overwrite physical memory. Once these tools are executed, the process and thread object will be allocated and initialized from windows kernel pool. This may overwrite the process and thread objects of terminated processes.

Win32dd1.2 [10] is a tool aimed at generating dump file from a live system. It has great compatibility because win32dd is well compatible with Windows based system. The most notable feature is the capacity of generating Microsoft crash dump file without rebooting or extra configures. However, win32dd1.2 takes a relatively long time to dump full physical memory. Taking a long time is one of the main drawbacks of this kind of tool. And it also has the disadvantage we have detailed before.

Windows based systems also develop a mechanism of generating crash dump file [11]. When system generates a crash dump file, it will overwrite pagefile, because creating a new file on the hard disk is determined by the integrity of kernel data structures. Besides, pagefile actually plays an important part in virtual memory management. Examiners can get a lot of useful information by analyzing paging file.

There are other techniques and tools to generate full physical memory dump. IEEE 1394 [12], which is only available when FireWire ports are opened, can access physical memory without having to go through the CPU.

The win32dd1.2 is the best choice among current available techniques generating full memory dump file. It can be used wildly, allowing for its compatibility with windows based system and few restriction. It also leaves as small footprint as possible, without hindering investigators from deeper forensics. Therefore, we choose win32dd1.2 as our generating dump file tools.

III. RELATED DATA STRUCTURES

The examiners are faced with mass data while the physical memory and hard disk are bigger and bigger. It's important to identify useful information from the mass data. We try to place the relevant information about QQ in one file. In this way, we are not to be bothered by the other extraneous information.

Windows based systems all implements the virtual memory management. Each process has an individual process address space of 4GB. This mechanism guarantees each process is not about to be disturbed by the others. In fact, the whole process space contains everything that the examiners are interested in such as the chats records, contact lists and so on. It is of great interest to the examiners if we can reconstruct the whole process space.

This section, we'll focus on the data structures that make up a process. Windows based system implements an object model to provide consistent access to the various internal services. For example, the kernel structure EPROCESS represents the process object. They're of great help to reconstruct the whole process and thread.

A. EPROCESS

An executive process (EPROCESS) represents a Windows process [13]. It not only has many attributes describing the process's current status, but also many pointers to other important internal structures, such as PEB, PCB. The process object plays an important role in Windows scheduling and resource allocation. Process scheduling is necessary because Window based system is a multi-task operating system. Resource allocation is also significant for any process to be run on the system. Such resource could be an access to a section of the computer's memory, data in a device interface buffer, opened files, and so on. The access is actually allocated by system, and

designed as the pointer to allocated source in EPROCESS data structure as shown in Fig.1.

The CreateTime and ExitTime fields give the accurate time that process are created and terminated.The UniqueProcessId field is actually PID, which is the identification of each process. The InheritedFromUniqueProcessId field is this process's parent process ID.The ImageFileName field gives the name of loaded image which is qq.exe for the QQ client.

B. KPROCESS

KPROCESS is the real process control block(PCB) in the Winodws kernel. Because the first member of an EPROCESS is a KPROCESS object, a EPROCESS pointer acturally points to an KPROCESS object as well, and vice versa. This means that EPROCESS and KPROCESS can be typecast interchangeably. The KPROCESS object is shown in Fig. 2. The pcb field in _EPROCESS is claimed to be an object of KPROCESS type.

Pcb also contains the DirectoryTableBase field, which is significant to reconstruct the process's whole address space. It also contains kernel time and user time which we can get acute execution time in kernel mode and user mode.

C. Invalid PTE

A page fault occurs to fault page in the memory when the valid (present) bit in a PTE is zero. However, there are times when the desired page is resident in memory while the valid bit in the faulted PTE is zero. This faulted PTE can be in one of the states described below.

a) Transition PTE: The desired page is already faulted into the memory. It's either in the standby or modified list. The higher twenty bits indicates the page frame number.

b)Demand Zero PTE: It's useless to reconstruct the whole process space.

c)PTE Referring to Page in Paging File: The desired page is swapped out into the paging file. The higher twenty bits indicates the page file offset. The page file number gives the paging file that the required page locates in.

d)PTE Referring to Prototype PTE: The required page is a shared page and the process has not accessed this page before.

D. Prototype PTE

Prototype PTEs are designed to share physical pages between different process spaces. For example, when a process calls the function CreateFileMapping(), a view of a section is mapped to this file. The PTEs in the process space would not refer to the real physical page at first. Instead, they are the pointers to the corresponding prototype PTEs for this section. The PTE would point at the real physical page after the first attempt to access to this virtual area.

```
lkd> dt !_EPROCESS
nt!_EPROCESS
   +0x000 Pcb              : _KPROCESS
   +0x06c ProcessLock      : _EX_PUSH_LOCK
   +0x070 CreateTime       : _LARGE_INTEGER
   +0x078 ExitTime         : _LARGE_INTEGER
   +0x080 RundownProtect   : _EX_RUNDOWN_REF
   +0x084 UniqueProcessId  : Ptr32 Void
.....................................
   +0x14c InheritedFromUniqueProcessId : Ptr32 Void
   +0x150 LdtInformation   : Ptr32 Void
.....................................
   +0x174 ImageFileName    : [16] UChar
.....................................
```

Fig. 1 The _EPROCESS represents process object.

```
kd>!kdex2x86.strct kprocess
nt!_KPROCESS
   +0x000 Header           : _DISPATCHER_HEADER
   +0x010 ProfileListHead  : _LIST_ENTRY
   +0x018 DirectoryTableBase : [2] Uint4B
.....................................
   +0x038 KernelTime       : Uint4B
   +0x03c UserTime         : Uint4B
.....................................
```

Figure 2. The KPROCESS provides page directory.

Prototype PTEs are a memory management structure and never reside in a page table page. They are allocated from paged pool.

The prototype PTEs also have a similar format as normal PTEs[14]. There are three kinds of prototype PTEs which are useful to us. They are valid prototype PTE, transition prototype PTE, as well as prototype PTE referring to page in paging file. We can locate the desired page according to these prototype PTEs.

As prototype PTEs are not resident in a page table, there is a need to acculate the exact address of the prototype PTE according to the faulted PTE. We propose the formula PteToProto() as shown in Fig. 3.

III. RECONSTRUCT THE PROCESS SPACE

The virtual memory mechanism takes full advantage of the physical memory. It also causes fragmentation in the memory. The pages of the same process are located here and there in physical memory. It's also an effective way to reconstruct the process space to correlate information.

This section, we detail the way to reconstruct the whole process space. The reconstructed process space contains everything about itself. Therefore, the examiners can search for sensitive information in a much smaller area.

Andreas Schuster [15] have proposed the method of using search patterns to search for the said object. This method can find not only the hidden processes but also the terminated processes. Therefore, even if the QQ client is closed, it is still possible to reconstruct the whole process space.

```
#define GetPdeAddress(va)   ( (va >> 22) << 2) + \
                      DirectoryTableBase&0xfffff000))
#define GetPteAddressV(va, pde) ((((va>>10)&0x3ff)<<
                      + (pde)&0xfffff000))
#define GetPtePagefileOffset(pde, va) =   \
                    ((pde & 0xfffff000)+ \
                    ((va & 0x3ff000)>> 12))
#define GetPagefileOffset(pte, va) ((pte & 0xfffff0(
                    (va & 0xfff))
#define PteToProto(PTE) ( (((PTE >> 11) << 9) +  \
                  (((PTE) << 24) >> 23) + \
                  MmPagedPoolStart))
```

Fig. 3- Macros for address translation

We find the EPROCESS object of QQ if the ImageFileName field is qq.exe.

B. Obtain the Page Directory Table

Since the first member of an EPROCESS is a KPROCESS, the DirectoryTableBase can be obtained. The DirectoryTableBase[0] in pcb actually incicates the page frame of the page directory table. The DirectoryTableBase locats at the offset of 0x18 of the KPROCESS.

C. Get the page directory entry(PDE)

The first ten bits of virtual address is the index in page directory table, which describes the location of the page table needed to map the virtual address.

D. Parse the Prototype PDE

We need to parse the prototype PDE.

1)Valid PDE: If the valid bit is set one, then it is a valid PDE. Examiners can get the PTE via the macro GetPteAddress as shown in Fig. 3.

2)Transition PDE: If the valid, prototype bits are set zero and transition bit is set one, this means it is a Transition PDE. The higher twenty bits indicates the page frame number that the page table locates. Examiners can also get the PTE through macro GetPteAddress.

3) PDE Referring to Pagefile: If the valid, prototype and transition bits are all set zero, the entry points to a frame in one of the paging files. It means the page table page is swapped out into the paging file in the hard disk. The offset of PTE in paging file can be calculated by the macro GetPtePagefileOffset as shown in Fig.3.

E. Parse the Prototype PTE

The page fault handler locates the PTE which describes the page. Once the PTE is found, it can be in one of states. However, what we are concerned about are only four states as follows.

1) Valid PTE: If the valid bit is set one, it is a valid PTE. The higher twenty bits is the page frame number the PTE points at.

2) Transition PTE: If the valid, prototype bits are set zero and transition bit is set one, it is a Transition PTE. The higher twenty bits indicates the page frame number that the PTE points to. The desired page is in memory on either the standby or modified list.

3) PTE referring to Pagefile: If the valid, prototype and transition bits are all set zero, the entry points to a frame in one of the paging files. The frame is shown in the macro GetPagefileOffset in Fig. 3.

4) Prototype PTE: If the valid bit is zero and prototype bit is set one, this means it's a PTE referring to prototype PTE. We develop the formula of converting PTE to Prototype PTE via reverse engineering. This formula is different from we have saw in other papers. The macro PteToProto is shown in Fig. 3. MmPagedPoolStart is the start address of the paged pool. Every valid prototype PTE locates in the paged pool. The value of this kernel global variable is 0xE1000000.

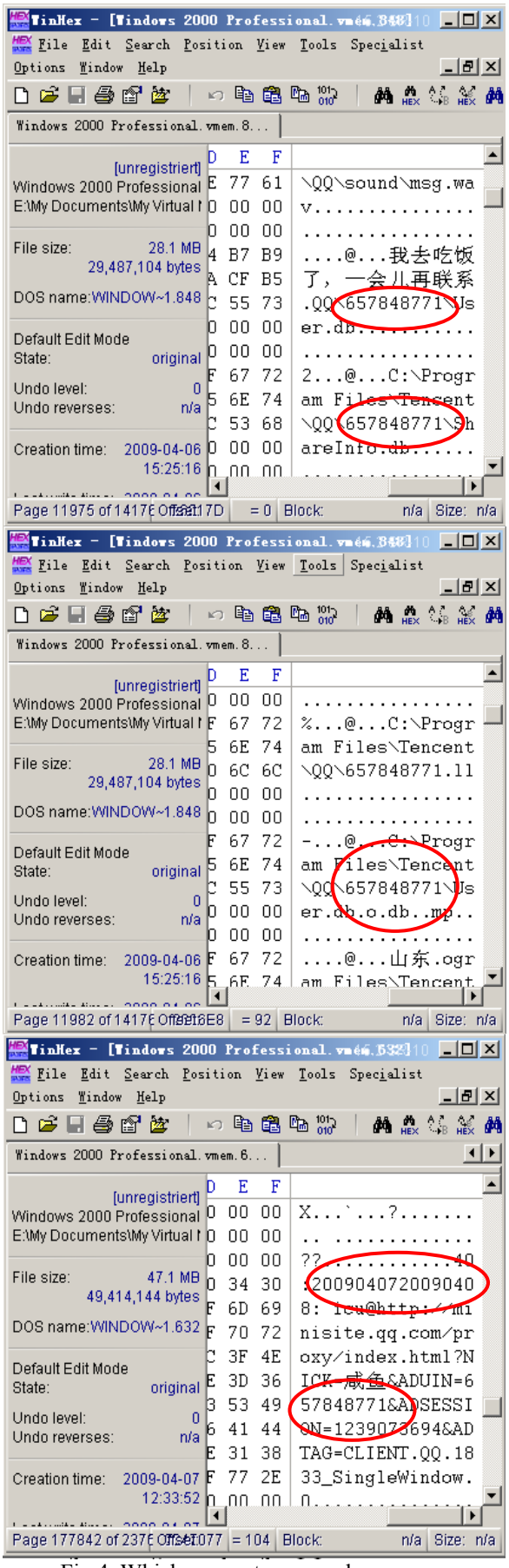

Fig.4. Which accounts are used.

The valid and transition PTEs as well as PTE referring to paging file directly indicate the location of required page while prototype PTE needs further processing.

5)Analyze the Prototype PTE: Prototype PTEs are created for the sake of sharing pages between multiple address spaces. It has a similar format as normal PTE.

The valid, transition and prototype PTE referring to paging file is processed the same as what we do with corresponding ordinary PTE. In these cases, we can get corresponding pages from memory dump file and paging file.

IV. Results Of Experiments

A series of experiments have been made to collect sensitive information from the dump file of QQ client. We have managed to get the contact list, the chats records, the QQ group and something else. They are of great interest to the examiners.

A. Which accounts are used

It's important to know which account the respondent has logged in. We can acquire more information based on this. For example, the simplest but most effective way is to demand for chat records and contact lists from the cooperation of the account.

There are two methods of confirming which accounts are used.

The first way is that the QQ client usually creates the corresponding file folder for each account after a successful login. For example, the respondent has logged in with QQ number: 123456. Then the QQ client is about to create the file folder whose path is "D:\Program Files\Tencent\QQ\123456". This folder contains everything about this account. Most of the files in this folder would not be deleted after a logout.

What's more, it's impossible to log in a single QQ client with different account at the same time. Therefore, only one account can be found in each independent process space. We can search for "\Tencent\QQ\" in the dump file of QQ process. In this experiment, we logged in with QQ number 657848771. The logged-in account can be retrieved as shown in Fig .4.

The second method is to search for 'http://minisite.qq.com/proxy/index.html' string in the memory dump file. After a successful login, the QQ client would open the page whose URL is http://minisite.qq.com. Besides, more parameters such as the time of login, the computer name, the nick name and the QQ number would be transferred as shown in Fig.4. These are the basic actions after a successful login. Therefore, it is very likely to get the account number.

B. Contact List

The contact list contains the members that respondent frequently talks with. The examiners expect to know who they are and what they have talked about.

The contact list is stored in the "user.db" file which is placed in "D:\Program Files\Tencent\QQ\657848771\". Because the users are likely to log in with the same account in another computer, the QQ client would

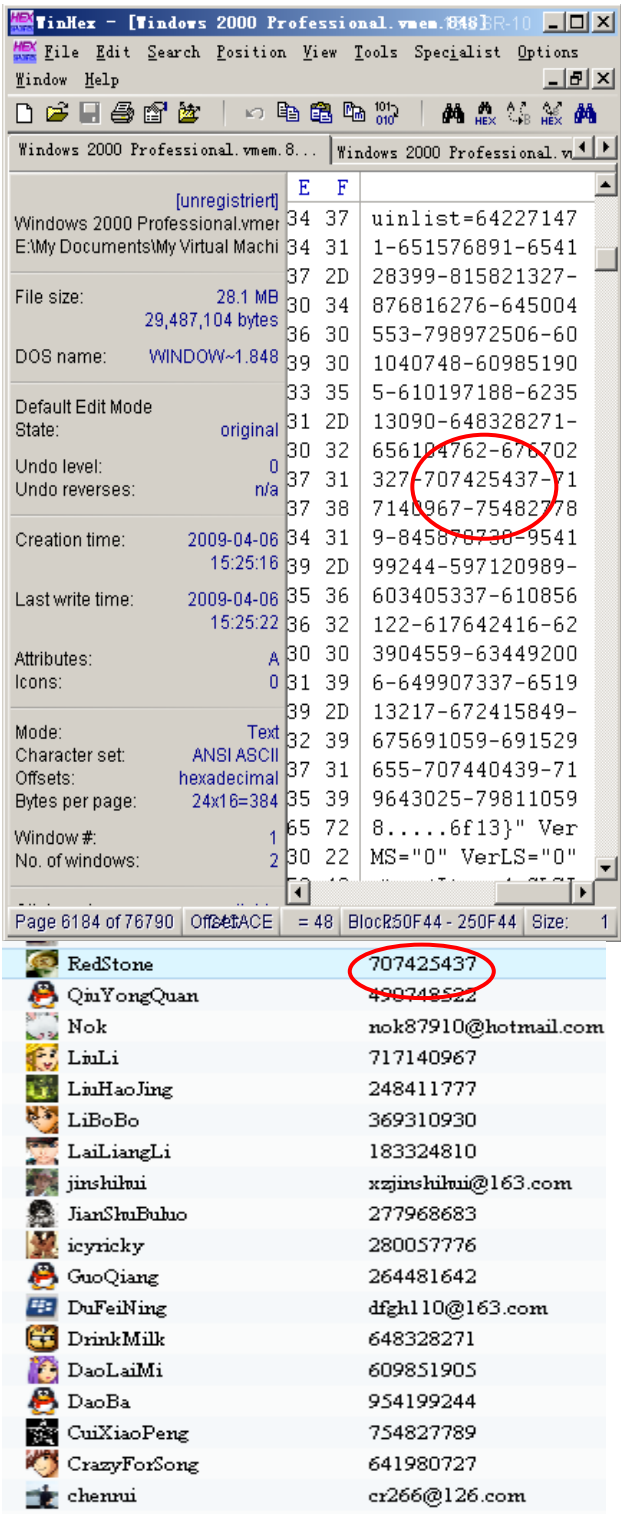

Fig. 5 Acquire the contact list via "list="

upgrade the contact list each time when the user logs in. The modifications will be written back into the "user.db" file. This file encrypted and the decryption is only known to the Tencent.

In the Windows Live Messenger, every member in the contact list is identified by the email address. However, each user of QQ has a nickname or username. There are times when one's name is a duplicate of the other's.

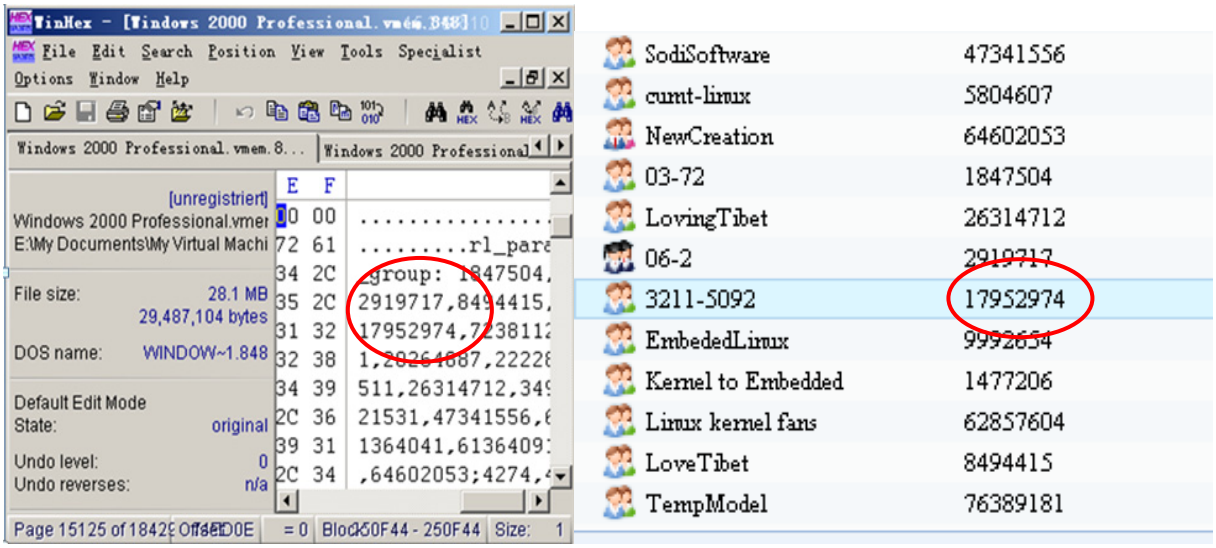

Fig. 6 The discussion group this account joins.

Therefore, the nicknames are useless to the examiners. We need to get the QQ number of contact person.

As we explore into the process space of QQ, we found that the QQ client uses the http request to acquire and update the contact list. The contact list is encapsulated inside the http request. In our reconstructed process dump file, we have managed to find the http request. Besides, we have found that the parameter "inlist" stands for the contact person who is online and the parameter "uinlist" represents the contact person who is offline. The results are shown in Fig. 5.

C.The QQ Disscussion Group

The QQ discussion group is a place to make friends and solve problems together. The examiners are interested in which group the respondent joined in. Besides, the members of QQ discussion group are listed on the group's homepage, it's useful to acquire the QQ discussion group numbers.

We are able to find the QQ discussion group numbers by performing a string searching on the process dump file with "rl_para_group:". This keyword gives all groups that the respondent joined in as shown in Fig. 6.

D.The Notepad function.

The QQ client provides network drivers to store files which are uploaded by the users. It also offer a notepad where the file is placed as "*.rtf". The QQ users are used to keeping things such as password or unfinished work in these files.

It's possible to demand for the uploaded files of the QQ user from the Tencent Cooperation. However, there are times when the QQ user would not like to save the file after editing it. There's no doubt that the examiners would not acquire it from the server. On the other hand, since the contents in the notepad need to be uploaded into the network driver, it's should be organized to be easily transported and parsed by the client. That means, it must appear in physical memory.

In our experiment, we started the network notepad provided by QQ client and enterd a series of numbers "18952230050". This phone number is used as keyword and we used WinHex to search on it as shown in Fig. 7.

In Fig. 7, we can see that the notepad is organized as the Rich Text Format (RTF). The RTF is a method of encoding formatted text and graphics for easy transfer between applications [16]. This file format is text formatting and we can conclude from Fig.7 that the character set is ANSI and the color is default.

Once we find the corresponding area that the notepad locates in, it's easy to recover the file. We use the keyboard shortcuts "Ctrl + shift + N" provided by WinHex and save it as "20090406-12-25-10test.rtf". The file can be opened and parsed well by Microsoft Word.

We can conclude that the keyword in RTF can be used to search for notepad in the reconstructed process space in QQ, such as the string "{\rtf1\ansi".

E. The display names.

The QQ users often use display names for the acquaintances in the contact list. It helps to correlate the person on the network with the one in the real world. Such as, the QQ user has a friend named James in the real world and named fisherman on the network. It's well worth using James as the friend's display name on the QQ client.

The users can change the display name whenever they log in. Therefore, after a successful login, the QQ client would request for the display names for every member in the contact list. These messages are transferred by the HTTP and not encrypted. We have carefully inspected the behaviors of QQ client when logged in. We found that the QQ client is likely to use "les\Tencent\QQ\QQ.exe P?" as parameter to request for the display names of this account.

We logged in with the QQ number 657848771 and search within the reconstructed process space of the QQ client. We managed to find the display names being transferred. The results are shown in Fig. 8.

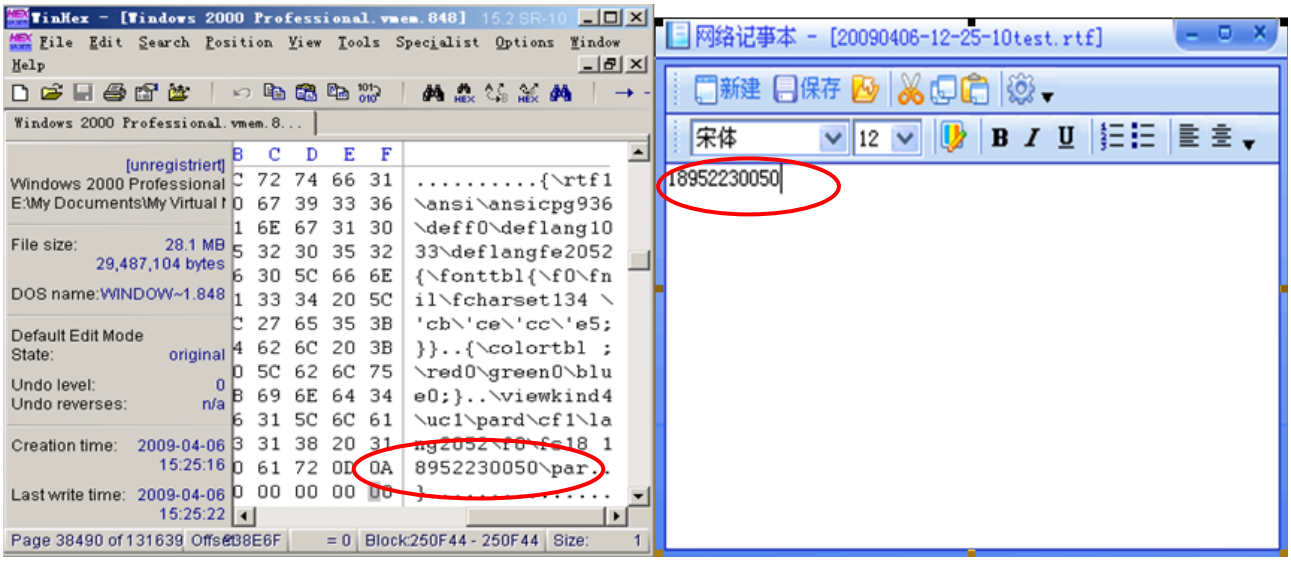

Fig. 7 The contents of the network notepad.

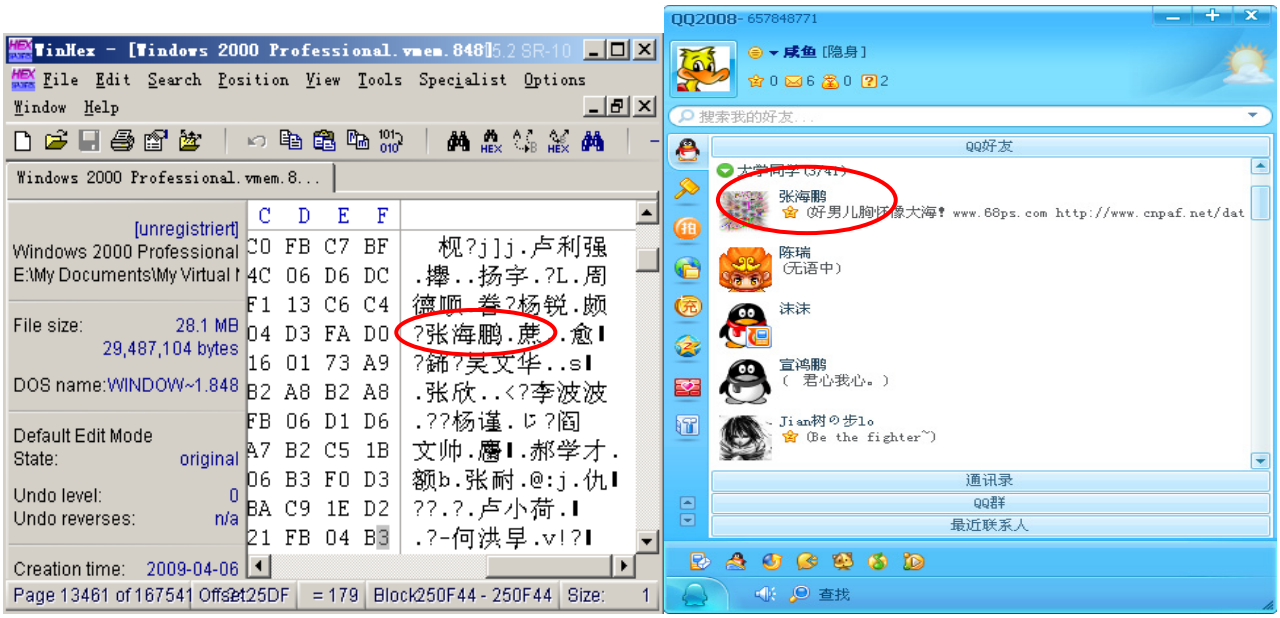

Fig. 8 The display names of this account.

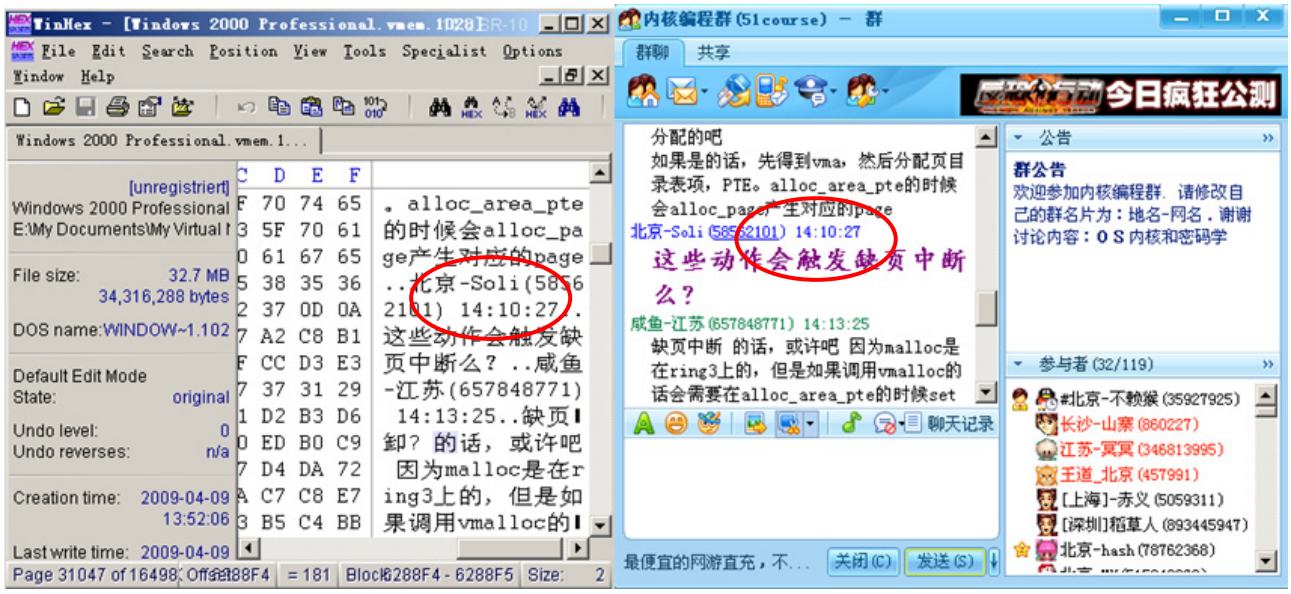

Fig. 9 Use the regular expression to find the chats records.

There are three places where examiners can find chat records. The first place is on the hard disk. The QQ creates a folder named after the QQ number of each user. The chat records are located in the file whose path is "D:\Program Files\Tencent\QQ\657848771\MsgEx.db". The MsgEx.db under this folder contains all chats records and other interesting information. Besides, this file is a little database implemented by the Tencent QQ itself and not appropriately optimized. Even if the user deletes all the chat records in the control panel of QQ, this file would not shrink. That means, the original chat records are still in this file. However, the information is encrypted and the decryption is unknown.

The second way is to analyze the OICQ protocol. The QQ use this protocol to transport messages as most instant messaging do. However, the chat records are also encrypted using Tiny Encryption Algorithm (TEA).The last but the most effective way is to retrieve the chat records from the physical memory. All sended and received messages must be plaintext to be readable for QQ users. If we reconstruct the process space, it's possible to retrieve the chats records. We have made a series of experiments to obtain the chats records. The chats records are well organized as follows: the account name, the QQ number, the message time, the message content. We use a regular expression to find these chats records. The results are shown in Fig. 9.

V. Conclusions And Future Work

In this paper, we have managed to reconstruct the process space of the QQ client. The invalid PTEs are taken into consideration and the paging file is made fully use of. Therefore, the reconstructed process space contains most sensitive information we need. Because we put all relevant information together, the examiners would not be bothered by collecting sensitive information from a big scale as Qian Zhao, Tianjie Cao [17] do.

We also have collected sensitive information of the QQ client such as the contact list, chats records, network notepad, display names and so on. The collected information is of great interest to the examiners.

We also gave an overview of some known live memory collection techniques on a Windows based system. A comparison between different techniques of memory collection is made, and we drew the conclusion that win32dd is a good choice taking its usability and compatibility into account.

In fact, process is the elementary unit of resource allocation in system. TCP ports, files, and device are all resources. EPROCESS has an object table, from which we can get what resources the process maintains. We would like to enumerate all resources of the QQ client and gather more sensitive information from this.

References

[1] Dickson M. An examination into MSN Messenger 7.5 contact identification. Digital Investigation Volume 3, Issue 2, June 2006, Pages 79-83.

[2] Dickson M. An examination into Yahoo Messenger 7.0 contact identification. Digital Investigation, Volume 3, Issue 3, September 2006, Pages 159-165.

[3] Dickson M. An examination into AOL Instant Messenger 5.5 contact identification. Digital Investigation, Volume 3, Issue 4, December 2006, Pages 227-237.

[4] Dickson M. An examination into Trillian basic 3.x contact identification. Digital Investigation, Volume 4, Issue 1, March 2007, Pages 36-45.

[5] van Dongen Wouter S. Forensic artefacts left by Windows Live Messenger 8.0. Digital Investigation, Volume 4, Issue 2, June 2007, Pages 73-87.

[6] Wouter S. van Dongen. Forensic artefacts left by Pidgin Messenger 2.0. Digital Investigation, Volume 4, Issues 3-4, September-December 2007, Pages 138-145.

[7] R.B. van Baar*, W. Alink, A.R. van Ballegooij. Forensic memory analysis: Files mapped in memory. Digital Investigation, Volume 5, Issues 1-2, September 2008, Pages 34-48.

[8] Andreas Schuster . The impact of Microsoft Windows pool allocation strategies on memory forensics. Digital Investigation Volume 5, Supplement 1, September 2008, Pages S58-S64.

[9] Sven B. Schreiber, Undocumented Windows 2000 secrects, a programmer's cookbook. Pearson Education Corporate;2001.

[10] Win32dd – a free kernel land and 100% open –source tool to acquire physical memory. http://win32dd.msuiche.net/.

[11] Microsoft.. Windows feature lets you generate a memory dump file by using the keyboard.

[12] David R. Piegdon and Lexi Pimenidis. Targeting Physical Address Memory. Detection of Intrusions and Malware, and Vulnerability Assessment, Volume 4579. 2007. Pages 193-212.

[13] Russinovich Mark, Solomon David. Microsoft Windows internals.4th ed. Redmond, Washington: Microsoft Press; 2005.

[14] Kornblum Jesse D. Using every part of the buffalo in windows memory analysis. Digital Investigation 2007;4:24–9.

[15] Schuster Andreas. Searching for processes and threads in Microsoft windows memory dumps. Digital Investigation 2006;3; 10–6

[16] Rich Text Format (RTF) Specification, version 1.6 http://msdn.microsoft.com/en-us/library/aa140277.aspx.

[17] Qian Zhao, Tianjie Cao. Collecting Sensitive Information from Windows Physical Memory，Journal of Computer, pp. 3-10, Volume 4, Number 1, January 2009.