Assembly Reverse Analysis on Malicious Code of Web Rootkit Trojan

Yong Wang[1,2] Dawu Gu[1]
1. Dept. of computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, China
e-mail: wy616@126.com

Janping Xu[2] Fenyu Zen[3]
2. Dept. of computer Science and Technology
3. School of Economics and Management
Shanghai University of Electric Power
Shanghai, China

***Abstract*—Web rootkits Trojan, which can download virus from remote control server and hide in BIOS, is very harmful to web security. Reverse assembly analysis on web rootkit Trojan can help virus analyzer to trace malicious code and find some immunization methods. The paper presents deeply reverse analysis methods of web rootkit Trojan according to malicious assembly codes. The MASM assembly instructions in malicious code are compared with turbo ASM to find the difference. Some famous Trojan, such as web downloader machine dog Trojan and BIOS Trojan, are assembly reverse analyzed. Finally, the paper proposed some detection and immunization methods of web rootkit Trojan using assembly language.**

Keywords-trojan; malicious code; reverse analysis; assembly language

I. INTRODUCTION

Rootkit is the system kernel technology, which is always used by the hackers to enter other computers and get the root privilege. Rootkit technology include kernel hooks, which consist of three main ways: import address hooking, inline function hooking and injecting DLL in userland processes [1]. Interrupt descript table (IDT) hooking and SSDT hooking belong to kernel mode hooks [4].

Rootkit Trojan uses the technology to control victim computers with the permanent or consistent, undetectable presence [1]. Web Rootkit Trojan can download virus from remote control server, and hide the Trojan server in web pages.

Rootkit Trojan can hide the server in BIOS, which sounds impossible and terrible. John Heasman proposed implementing and detecting PCI rootkit. He presented a method to persisting a rootkit in the system BIOS via the Advanced Configuration and Power Interface (ACPI). It was demonstrated that the ACPI tables within the BIOS could be modified [5]. John G. Levine presented a methodology for detecting and classifying rootkit exploits in his thesis [6]. He also made research on rootkit exploited in system call table [7] and characterized rootkit retrieved from honeynets in papers [8].

To explore rootkit Trojan, Francis M. David made research on hardware supported rootkit concealment [9], Christopher Kruegel presented several ways to detecting kernel-level rootkit through binary analysis [10]. John G. Levine proposed that rootkit categorization approach, which could help system administrators to identify the extent of specific infections, aiding in optimal recovery and faster reactions to future attacks [11]. Some rootkit test Trojan is designed in kernel mode, which can pass the virus scan and firewall, rootkit detection of some famous virus software. Gmail servers begin to use HTTPS to log in to web site with username and password.

In order to defense the Trojan's attack, analyzer needs reverse assembly analysis on web rootkit Trojan, which can help virus analyzer to trace malicious code and find some immunization methods.

We compare turbo ASM and MASM of malicious code to find how Trojan or virus use Rootkit assembly instruction. We choose some famous Trojan, web downloader machine dog Trojan and BIOS Trojan, to reverse assembly analysis. Finally, try to find detection and immunization methods of web rootkit Trojan using assembly language.

II. ASSEMBLY ANALYSIS OF MALICIOUS CODE

A. Turbo ASM Analysis for CIH Virus

Turbo Assembler (TASM) is Borland Turbo assembler. Turbo Assembler 5.0 is a full featured stand-alone assembler which includes many tools needed to create and debug assembly programs for 16 and 32 bit DOS and Windows platforms, including Windows 3.X, Win95, Win98, and NT. Some of the tools included are assemblers, linkers, console style debuggers, and resource compilers. Each of these tools comes in a 16 bit and a 32 bit version.

CIH virus is compiled by TASM 4.0, can kill all hard disks even BIOS by modify IDT to get ring0 privilege. Virus code doesn't reload into system and can call hook file system by file system API Hook. CIH can modify entry point of system API Hook. When system opens existing PE file, the file will be infected even the file is read only, and the file doesn't be infected again. When the file is infected, the modification date and time of the file also don't be changed.

The most important procedure is getting ring0 privilege by modifying IDT. The procedure is as bellow.

- Get IDT base address by instruction SIDT.
- Calculate base address of interrupt.
- Close interrupt request by instruction CLI.
- Modify interrupt to virus procedure.
- Open interrupt request by instruction STI.

978-0-7695-3817-4/09 $26.00 © 2009 IEEE
DOI 10.1109/WISM.2009.107

B. *MASM analysis for rootkit Trojan*

Rootkit Trojan can be compiled by Microsoft assembler besides Turbo Assembler. The Microsoft Macro Assembler (MASM) is an x86 high-level assembler for DOS and Microsoft Windows. Earlier versions were MS-DOS applications, and MASM 5.0 has lots of instruction set. MASM611 and MASM615 added the capability of producing programs for windows console applications and windows applications. MASM32 is an integrated system of include files, import libraries and macros, which is for windows API programming and 32 bit assembler for the Windows platform.

The most used instruction of Rootkit Trojan are kernel instruction such as interrupt handler. An interrupt handler may be executed in one of two ways: an application program containing an INT instruction automatically calls the handler, execute an interrupt service routine. The CLI (clear interrupt flag) instruction disables interrupts and the STI (set interrupt flag) instruction enables interrupts [2, 3].

INT 21h Functions 25h and 35h make it possible to install interrupt handlers. Function 35h can get interrupt vector and return the segment offset address of an interrupt vector. Call the function with desired interrupt number in al. The 32-bit vector is retuned by MS-DOS in ES: BX. The following statements would retrieve the INT 9 vector. Interrupt handle procedure in real address mode is as bellow.

- Save old interrupt vector. Move vector number 1ch to al register and move 35h to ah register, use interrupt 21h to get interrupt vector and save registers for restore.
- Set new interrupt vector. Move virus procedure offset to dx register and move virus procedure segment to ax register. Using DOS function 25h in ah register to set new interrupt vector.
- Restore old interrupt vector. Move 25h to ax register and use int 21h to restore old interrupt vector.

The virus handle procedure result is shown as Figure 1.

Figure 1. Results of virus interrupt handle in real address mode.

III. Reverse Analysis on Web Rootkit Trojan Using Assembly Codes

A. *Analysis on Machine Dog Trojan*

Machine dog is one of Trojan which can infect explore.exe and userinit.exe regedit.exe and so on. The Trojan can realize the self start by the system infection. The Trojan can also return to SSDT to original state, which cause certain security software's defense and IDS error.

Once userinit.exe failed to init, the administrator user can't login in the system even they input username and password correctly. The system just shows the login windows again. Many users had to format computers to install operating system again.

You can put a windows setup with same version in driver, choose system repair menu to expand original userinit.exe and explorer.exe. The command instructions are as follows:

attrib -a -b -h –s c:\windows\system32\userinit.exe
attrib -a -b -h –s c:\windows\explorer.exe
expand g:\i386\explorer.ex_ c:\winnt\
expand g:\i386\userinit.ex_ c:\winnt\system32\

After that, the windows can login in correctly. You need to find the virus using anti-virus software. Many affection files are founded and recovery or deleted.

Virus analyzers usually choose IDA tools to analysis the visual structure of virus. The machine dog visual structure using IDA version 5.2 is as Figure 2:

Figure 2. Visual structure of machine dog using IDA.

OllyDbg version 1.1 is a 32-bit assembler level analysing debugger for Windows, which can traces registers, recognizes procedures, API calls, switches, tables, constants and strings. Analysis the virus string, the analyzer can find the virus affection directory or Trojan IP and other important information. The machine dog reference string is as Figure 3:

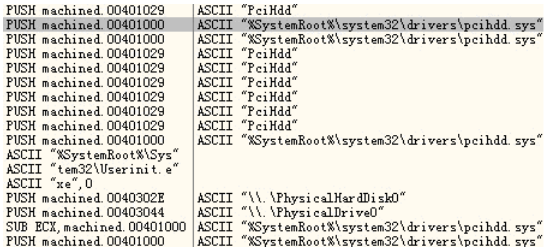

Figure 3. Reference string of machine dog using OllyDbg.

From the Figure 3, we can find import information. Reference string of machine dog is as follows:

%SystemRoot%\system32\drivers\pcihdd.sys

PE Explorer version 1.98 r2 is the most feature-packed program for inspecting the inner workings of virus, which can look inside these PE binary files, perform static analysis, reveals a lot of information about the function of the executable. After analysis the PE structure, the virus import and export details are clear. The machine dog Trojan import table is Figure 3:

TABLE I. IMPORT TABLE USING PE EXPLORER

kernel32.dll		advapi32.dll
WriteFile	ExitProcess	ControlService
SizeofResource	SetFilePointer	CreateServiceA
CloseHandle	FindResourceA	DeleteService
CreateFileA	FlushFileBuffers	OpenSCManagerA
DeleteFileA	GetModuleHandleA	OpenServiceA
DeviceIoControl	GlobalFree	StartServiceA
ExitProcess	LoadResource	CloseServiceHandle
GlobalAlloc	OutputDebugStringA	
LockResource	ReadFile	
RtlZeroMemory	SetEndOfFile	
ExpandEnvironmentStringsA		

B. Analysis on BIOS Trojan

BIOS is the abbreviation of Basic Input and Output System, which contained on EEPROM cards. BIOS has also supported power management routines and adheres to Advanced Configuration and Power Interface (ACPI) standards. Because it is possible to write to BIOS flash memory based on motherboard settings, such as Intel. Usually the motherland default settings don't be allowed writing BIOS, yet the settings can be changed by hardware setup program.

BIOS Trojan is the virus that can hide in BIOS and connect to remote computer. Once the Trojan infects the BIOS, the anti-virus software can't delete the virus and even after you install windows or linux again. It sounds impossible and terrible, but it is truth.

In February 2006, John Heasman presented a means of persisting a rootkit in the system BIOS via the Advanced Configuration and Power Interface (ACPI). It was demonstrated that the ACPI tables within the BIOS could be modified to contain malicious ACPI Machine Language (AML) instructions that interacted with system memory and the I/O space, allowing the rootkit bootstrap code to overwrite kernel code and data structures as a means of deployment [5]. PCI Rootkit can reside on Sound Cards, Modems, Network Cards, Capture Cards or any other PCI device that has an Expansion ROM and no Trusted Platform Module or ROM write protection. Most current PCI devices are possible to this form of Rootkit infection although newer models have some form of ROM protection.

An attacker can place rootkit Trojan code in an Expansion ROM of many PCI devices that have no ROM protection. When the PC boots up, the Trojan code in the ROM is activated. Award Bios Editor version 1.0 or another tool trick that enables you to extract and replace original settings from the motherboard bios with the modified version. The award bios editor is as Figure 4:

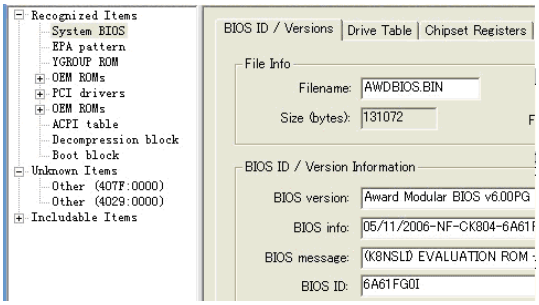

Figure 4. Award BIOS editor.

The tools can disassemble original settings using ndisasmw.exe. ASM instruction can disassemble as follows:

Address	Binary Code	Mnemonic
000000E4	06	push es
000000E5	0FA8	push gs
000000E7	66B8FFFFFFFF	mov eax,0xffffffff
000000ED	668986E901	mov [bp+0x1e9],eax

IV. WEB ROOTKIT TROJAN DETECTION AND IMMUNIZATION

A. Definition of Web Trojan Scan

Web Trojan scan program can be compiled using MASM32. The definition composed of the include files and include library files. The scan program needs also define some Trojan keywords in virus library. The details definition as table 2:

TABLE II. DEFINITION OF WEB TROJAN SCAN

include	includelib	Virus Key words
kernel32.inc	kernel32.lib	WScript.Shell
user32.inc	user32.lib	shell_exec
wsock32.inc	wsock32.lib	insert into
shell32.inc	shell32.lib	cmd
comctl32.inc	comctl32.lib	Shell.Application
advapi32.inc	advapi32.lib	RUNAT=SERVER
windows.inc	…	…

B. ASM Key Instruction of Web Trojan Scan

When scan procedure begin to detect potential risk, the scan program needs compare the web instruction with the virus keywords library. The rich and accurate keywords library is very important. ASM key instructions of web scan procedure are as follows:

```
mov edi,0
mov esi,0
mov  ecx, fileSize
cld
cmpNextString:
  …
mov  webString[edi],stringKey[esi]
repe cmpsb
  …
loop cmpNextString
 cmp ecx,0
 jnz  webTrojanFound
 .exit
webTrjanFound
```

C. Web Trojan Immunization Methods

Trojan immunization includes kill virus process, delete possible auto files, build auto immunization files or folder, append files attributes and authority, edit registry for web Trojan immunization. The immunization details methods are as follows:

1) Kill virus process:

```
tskill svhost
tskill swchost
tskill IGM
tskill IGW
```

2) Delete authority, attribute and files:

```
cacls c:\auto.exe /e /p everyone:f
attrib c:\auto.exe -r -h –s
del c:\auto.exe /f /q
```

3) Append authority, attribute and files:

```
echo>>c:\auto.exe
attrib c:\auto.exe +r  +h +s
cacls c:\auto.exe /e /p everyone:n
```

4) Build immunization files and folder:

```
cacls c:\windows\ptsshell.exe /e /p everyone:n
md gaga..\
md haha..\
```

After the immunization run, web virus can't delete folder gaga and haha, unless you delete in command line as follows:

```
rd gaga..\
rd haha..\
```

5) Edit registry for web Trojan immunization:

```
reg delete "HKEY_LOCAL_MACHINE\ SOFTWARE\Classes\CLSID\{00000566-0000-0010- 8000-00AA006D2EA4}" /f
```

There are many other commands for web Trojan immunization. The virus analyzer can expand the command library.

V. DISCUSSION

Reverse assembly analysis on web rootkit Trojan can help virus analyzer to trace malicious codes and find some immunization methods. The assembly reverse analysis methods is very important ways used in our actual works. The actual test results show that we still needs other tools to analysis malicious code.

Many web Trojans begin to encrypt the malicious code and data packets against debug software and IP tracking. Even analyzer capture the IP packet, they also need to decrypt the cipher text. Some mail servers begin to use HTTPS to log in to web site with username and password. 126 mail server uses SSL security feature and Gmail using HTTPS. In the lab test, users can create web Trojan only by click button. Building a web Trojan which can't be found by famous anti-virus software seems easily.

There are many other ways besides assembly code reverse analysis. In order to find the Trojan, you need to empty cookies, scan system files, monitor register table, IDT, GDT, LDT or SSDT changes.

ACKNOWLEDGMENT

The paper is supported by National hi-tech research and development project No.2006AA01Z405. Supported by Shanghai Postdoctoral Scientific Program (No.08R214131). Supported Innovation Program of Shanghai Municipal Education Commission (No. 09YZ346).

REFERENCES

[1] Greg Hoglund, and James Butler, Rootkits: Subverting the Windows Kernel, Addison Wesley Professional,2005.

[2] Intel Corporation,IA-32 Intel Architecture Software Developer's Manual Volume 3:System Programming Guide,2004.

[3] Kip R.Irvine, Assembly Language for INTEL-Based Computer,Fourth Edition,Pearson Eduaction Asia Limited and Tsinghua University Press, 2005.

[4] Chris Ries, "Inside windows rootkits", VigilantMinds Inc.2006

[5] John Heasman, "Implementing and detecting a PCI rootkit", An NGSSoftware Insight Security Research (NISR) Publication,2006. pp:1-15.

[6] John G. Levine, "A methodology for detecting and classifying rootkit exploits", Georgia Institute of Technology, 2004.

[7] John G. Levine, Julian B. Grizzard, Phillip W. Hutto , and Henry L. Owen, "A methodology to characterize kernel level rootkit exploits that overwrite the system call table", SoutheastCon, 2004.Proceedings. IEEE 26-29,Mar 2004 pp:25-31.

[8] John Levine,Julian Grizzard, and Henry Owen, "Application of a methodology to characterize rootkits retrieved from honeynets", Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC 10-11 June 2004 ,pp:15-21.

[9] David, F.M. Chan, E.M. Carlyle, J.C. Campbell, and R.H, "Cloaker: hardware supported rootkit concealment",Security and Privacy, 2008. SP 2008. IEEE Symposium on18-22 May 2008,pp:296-310, DOI 10.1109/SP.2008.8

[10] Kruegel, C. Robertson, W. and Vigna, G "Detecting kernel-level rootkits through binary analysis",Computer Security Applications Conference, 2004. 20th Annual 6-10 Dec. 2004,pp:91-100.

[11] Levine, J.F. Grizzard, J.B. and Owen, H.L. "Detecting and categorizing kernel-level rootkits to aid future detection",Security & Privacy, IEEE Volume 4, Issue 1, Jan.-Feb. 2006,pp:24-32.