An Introduction to Undetectable Keyloggers with Experimental Testing

Mehdi Dadkhah*[1], Mohammad Davarpanah Jazi[2], Ciobotaru Ana-Maria[3], and Elaheh Barati[4]

Abstract—Keyloggers are used as a tools by attacker to steel user's usernames and passwords in e-commerce, Social network, Mail service and etc. There are many security software for detecting keyloggers and some technique have introduce for dealing with them. In this paper we will show that keyloggers can be undetectable from security software. We will make a keylogger and then will change the structure of keylogger then test it against popular security software in the world. At final we will show that many security software cannot detect keyloggers. Our goal is to introduce this new challenge.

Index Terms— Keyloggers, Security software, Anti viruses, Encoding.

1 Introduction

With increasing growth of communication networks, social interactions and financial transactions have been migrate to virtual environments. Internet is one of the most substantial platform for most people's social interactions and transactions. However, the notable challenge in online transactions is security in cyber environments and to understand the hazards accompanied with this communication platform. Because of the increased use of Internet and virtual environments in daily affairs such as financial transactions, this platform has become the focus of attackers and swindlers, for stealing user's information by keyloggers that they are one kind of malwares.

Malware is a program which intends to do unwanted or disrupting tasks in operating system without user's permission [1]. The first malware was a virus which was written in early 1980s with the purpose of disrupting stored information in computer systems. Then first network worms were born [2] in 1988 for contaminating SunOS and VAX BSD systems. It attacked these systems through network vulnerability and after inserting, ran a disruptive program on the system. From 2000, new techniques were invented for jobbery by malware and the main goal of malware was misusing computers as zombies [3]. In that time backdoor software was massively used. Since 2003 spying becoming popular and other goals like stealing passwords were performed by attackers. One of this malware is keyloggers. Keyloggers are the software that capture anything that type with user's keyboard. They also can take image from user's PC desktop then send this captured information to predefined email or FTP address. In many case, keylogger can spread via Network or Flash memory. Hacker use wide range of programing language to make keylogger such as C#, VB, Java. For detecting keyloggers many technique have made. In [4] have introduced a technique for dealing with keyloggers that use a malicious profile for detecting behavioral that similar to keyloggers behavioral. This approach can fail because in many case, keyloggers use a Gmail Service as predefined Address and also use "port 587" for sending information via email. This behavioral is very similar to Email services software such as Outlook or Firefox Sun Bird. Also virtual keyboard are design to deal with keylogger [5], when user use virtual keyboard, user's information will not enter via users keyboard so keyloggers cannot captured anything but advanced keylogger can take image from PC desktop then attacker can guess the password. Another way for dealing with keylogger use sequence of random characters, in this technique the information that captured by keyloggers will contain the password, but embedded in so much random junk that discovering it is infeasible, but this method use simple mechanism and attacker may detect password [6]. Up-to-date antivirus and anti-adware tools can help for dealing with keyloggers greatly but we will show that these software can fail against advanced keyloggers [7]. In many case hackers use a wide range of programing language for making the keyloggers and antivirus software's can detect them by analysis the behavioral of suspicious files. But in this paper we will introduced undetectable keyloggers then will explain the process of making this keyloggers.

- *Master Student, Department of Computer and Information Technology, Foolad Institute of Technology, Foolad shahr, Isfahan 8491663763, Iran, mdt@dr.com.*
- *Faculty Member, Department of Computer and Information Technology, Foolad Institute of Technology, Foolad shahr, Isfahan 8491663763, Iran, mdjazi@cc.iut.ac.ir*
- *Msc, Faculty of Geography, University of Bucharest, Romania; Secondary School Gura Calitei, com. Gura Calitei, Vrancea, Romania, ciobotaruanamaria@inbox.lv*
- *Faculty Member, Department of Computer and Information Technology, Foolad Institute of Technology, Foolad shahr, Isfahan 8491663763, Iran, elaheh.barati@gmail.com*

2 Our Research About Undetectable Keyloggers

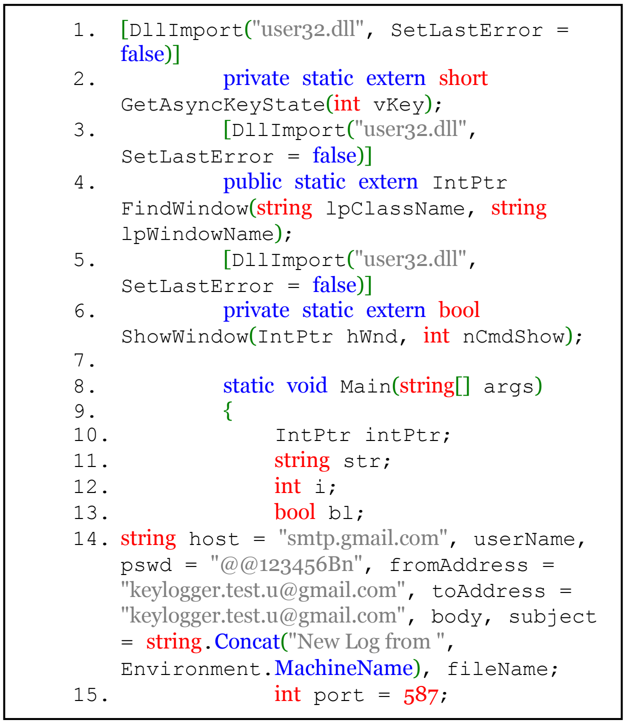
We used a keylogger that is written with C# language because many hacker use it and have many function that suitable for this work such as connecting to mail services. For encoding the source codes and string we will use smart assembly and Multimedia Builder. Smart assembly is popular software in encoding domain. For testing the level of encoding we will test keylogger with BinText Tools that it extract all text from any file and we can see un-encoded text and for final testing the keylogger again popular antivirus we use online labs such as Jotti and VirusTotal.

2.1 Materials of Our Research

The keyloggers that most Up-to-date antivirus and anti-adware tool cannot detect them are named as "undetectable keyloggers". At first we make a keylogger with C# language, then use smart assembly tools for string encoding the keylogger and Multimedia builder software for embedding the keylogger in another file and at last we test popular antiviruses against keylogger.

2.2 Making the Keylogger

We used a keylogger that is written with C# language. This keylogger can send captured information to predefined email address and use a Gmail account for sending email (figure 1). Also it save the captured data in txt file in the place that keylogger located. All source code have been append.



```
[DllImport("user32.dll", SetLastError = false)]
        private static extern short GetAsyncKeyState(int vKey);
        [DllImport("user32.dll", SetLastError = false)]
        public static extern IntPtr FindWindow(string lpClassName, string lpWindowName);
        [DllImport("user32.dll", SetLastError = false)]
        private static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

        static void Main(string[] args)
        {
            IntPtr intPtr;
            string str;
            int i;
            bool bl;
string host = "smtp.gmail.com", userName, pswd = "@@123456Bn", fromAddress = "keylogger.test.u@gmail.com", toAddress = "keylogger.test.u@gmail.com", body, subject = string.Concat("New Log from ", Environment.MachineName), fileName;
            int port = 587;
```

Fig. 1. Source code of keylogger.

2.3 Encoding the Strings of Keylogger

The encoding stage involve some task that describe below. This stage is critical because if this stage done good many security software cannot detect the keylogger. We use smart assembly (a tools that programmer use to secure their code against cracker) for encoding. Figure 2 show the encoding process.

Assigning a strong name key: this task protect keylogger from assembly edition and text extraction.

Pruning code: pruning task removes code that will never be executed at runtime. Pruning also removes metadata such as design attributes and the names of events and properties thus it is hard for people to use reverse engineering and analyze the keylogger structure. Also this task improve the performance and speed of keylogger and reduce the size of keylogger up to 30%.

Obfuscating code with name mangling: obfuscating task change class name and methods to unreadable name and increase security of code thus it hard for security analyzer to understand the structure of keylogger.

Control flow Obfuscating: convert source code to spaghetti code [8]. It convert code to complex and unstructured code. It is difficult to read or follow by a people and security software because it cannot be organized and in many time and vice-versa.

References dynamic proxy: this task create a proxy for external call and hide all call to external of codes. This dynamic proxy secure keylogger from security software and increase undetectable rate, so security software cannot track keylogger and after detect it.

Encoding strings: this task protect form passwords (such as passwords, query and information) by encoding them.

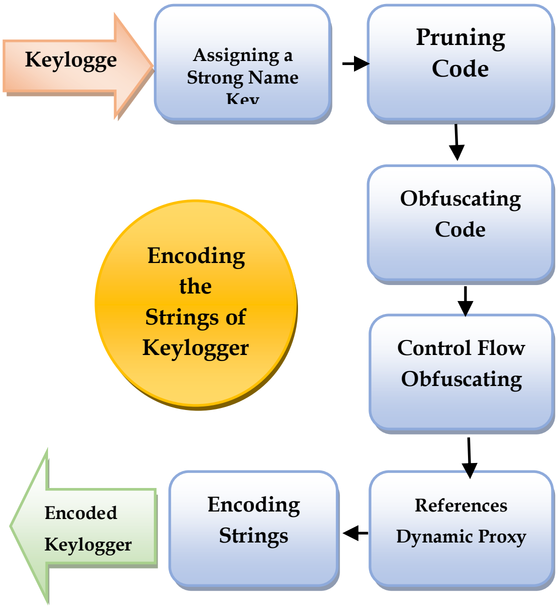


Fig. 2. Encoding the strings of keylogger.

2.4 Embedding the Keylogger in another File

This task is a sample of social engineering attack [9]. On this task, the keylogger will embedded in another file then will set for embedded keylogger like a deceptive icon. Also, on this task the extension of keylogger file will change to another extension (such as jpg, mp4) by Extension spoofing that use security hole in the Windows operating system and allow attacker to change extension of keylogger file to any extension. To do this stage we can use Multimedia Builder software.

3 TESTING SECURITY SOFTWARE'S AGAINST UNDETECTABLE KEYLOGGER

For testing security software against undetectable keyloggers, we use best antivirus according Av-Comparative report at March 2014 [10]. The participated antivirus in At Av-Comparative test have shown below.

- AhnLab V3 Internet Security 8.0.8.2
- Avast! Free Antivirus 2014.9.0.2013
- AVG Internet Security 2014.0.4335
- AVIRA Internet Security 14.0.3.350
- Baidu Antivirus 4.0.9.57999 (EN)
- Bitdefender Internet Security 17.26.0.1106
- BullGuard Internet Security 14.0.278.3
- eScan Internet Security 14.0.1400.1572
- Emsisoft Anti-Malware 8.1.0.40
- ESET Smart Security 7.0.302.26
- F-Secure Internet Security 14.99.103
- Trend Micro Titanium Internet Security 7.0.1206
- Fortinet FortiClient 5.0.8.344
- Kaspersky Internet Security 14.0.0.4651 (e)
- Kingsoft Internet Security 2013.SP6.0.030511
- Lavasoft Ad-Aware Free Antivirus+ 11.1.5354.0
- McAfee Internet Security 16.8.708
- Microsoft Security Essentials 4.4.304.0
- Panda Cloud Free Antivirus 2.3.0
- Qihoo 360 Internet Security 4.9.0.4109 (EN)
- Sophos Endpoint Security and Control 10.3.1

According this report best antivirus of 2014 based on detection rates and false alarms are shown at Table 1. We also use this antivirus for testing them against undetectable keyloggers. Table 2 show the scan result in every stage. In scanning progress antivirus software cannot detect anything but 360 Internet Security detect the keylogger after running it by behavioral analysis.

TABLE 1
REPORT BEST ANTIVIRUS OF 2014 BASED ON DETECTION RATES AND FALSE ALARMS.

Rank	*Antivirus name*	*Rank*	*Antivirus name*
1	Kaspersky Lab	9	Qihoo (en)
2	F-Secure	10	McAfee
3	eScan	11	Panda
4	Fortinet	12	AVIRA
5	Emsisoft	13	Tencent
6	Bitdefender	14	Trend Micro
7	Lavasoft	15	ESET
8	BullGuard	16	ThreatTrack Vipre

TABLE 2
SCAN RESULT IN EVERY STAGE OF MAKING UNDETECTABLE KEYLOGGER.

Antivirus software	***Stage of work***		
	Making the keylogger	***Encoding the strings of keylogger***	***Embedding the keylogger in another file***
Kaspersky Lab	Found nothing	Found nothing	Found nothing
F-Secure	Detected	Found nothing	Found nothing
eScan	Detected	Found nothing	Found nothing
Fortinet	Found nothing	Found nothing	Found nothing
Emsisoft	Detected	Found nothing	Found nothing
Bitdefender	Detected	Found nothing	Found nothing
360 Internet Security	Detected	Detected	Found nothing
McAfee	Found nothing	Found nothing	Found nothing
Panda	Found nothing	Found nothing	Found nothing
AVIRA	Detected	Detected	Found nothing
Tencent	Found nothing	Found nothing	Found nothing
Trend Micro	Found nothing	Found nothing	Found nothing
ESET	Detected	Found nothing	Found nothing
ThreatTrack Vipre	Found nothing	Found nothing	Found nothing

4 CONCLUSIONS

In this paper we discuss the problem of keyloggers and we show that keyloggers can be undetectable from Up-to-date antivirus and anti-adware tool and existing technique can be fail against advanced keyloggers. We explain stage and task of creation of undetectable keyloggers. We describe a new challenge that must attract by Antivirus Company's.

5 ACKNOWLEDGEMENTS

I would like to acknowledge to teacher Phd. Andronache Ion from "Constantin Brancoveanu" Technical School from Braila, Romania.

6 APPENDIX

A. Source code of keylogger

Source code available at:
http://codeviewer.org/view/code:414d

B. Macro of MMB for Embedded file

Run("<Embedded>/capsulate2.exe","")
ExitTimer("2500")

C. Online Result of Testing Antivirus against Keylogger

For stage 1 (making keylogger):
File size: 9728 bytes
File type: PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows
MD5: 3cc8c9d692e5f6cc7f067fc407385453
SHA1: 12abcdfd9496cd85aa38dfc5595fa7eeceb8d9a3
Test result available at:
https://www.virustotal.com/en/file/5113f0d8b85ebddce253b19950def8ea9b1242d0875a2d8ac3db82a96086d439/analysis/1403780773/

http://virusscan.jotti.org/en/scanresult/d3887d823d77d9512a696162b08d81b0efe31b0a/eb2617cb37d467de1de23be38db56b665c64180d

For stage 2 (Encoding the strings of keylogger):
File size: 31232 bytes
File type: PE32 executable (console) Intel 80386 Mono/.Net assembly, for MS Windows
MD5: 82e5749a6f17512f66dfa6b48e3e8545
SHA1: 67c855bf85d1f13b2804b873ba1adb986a97aef9
Test result available at:
http://virusscan.jotti.org/en/scanresult/487ec86e33ab69b54a1118ebb6bb280e618dd13d

https://www.virustotal.com/en/file/7bf3ce276988fad774503655c5e2e8ed5274f4bed70f7dc85ae582963d3fed48/analysis/

For stage 3 (Embedding the keylogger in another file):
File size: 546525 bytes
File type: PE32 executable (GUI) Intel 80386, for MS Windows, UPX compressed
MD5: b338052386ca0179a72963aaee03b71a
SHA1: 09332e9c855b8e42a500e7a5159fc907c70e1f75
Packer (Avast): UPX
Test result available at:
http://virusscan.jotti.org/en/scanresult/b6453d9730db07a8d01bba8d1194c8396b6041c6

https://www.virustotal.com/en/file/11e32ff708bd964388fdef85362be5465791d3e062dca0ff5d19e963de227c73/analysis/1403781650/

C. Behavioral analysis of Keylogger in stage 3

- *File Info*

Name	Value
Size	546525
MD5	b338052386ca0179a72963aaee03b71a
SHA1	09332e9c855b8e42a500e7a5159fc907c70e1f75
SHA256	11e32ff708bd964388fdef85362be5465791d3e062dca0ff5d19e963de227c73
Process	Exited

- *Values Changed*

Name	Type	Size	Value
LM\Software\Microsoft\Direct3D\MostRecentApplication\Name	REG_SZ/REG_SZ	22/22	"msoobe.exe"/"sample.exe"
LM\Software\Microsoft\DirectDraw\MostRecentApplication\ID	REG_DWORD/REG_DWORD	4/4	0x3b7d853e/0x473b220c
LM\Software\Microsoft\DirectDraw\MostRecentApplication\Name	REG_SZ/REG_SZ	22/22	"msoobe.exe"/"sample.exe"

- *Files Created*

Name	Size	Last Write Time	Creation Time	Last Access Time	Attr
C:\Documents and Settings\User\Local Settings\Temp\MMBPlayer\capsulate2.exe	31232	2009.01.09 10:37:28.484	2009.01.09 10:37:28.484	2009.01.09 10:37:28.484	0x20

- *Processes Created*

PId	Process Name	Image Name
0x4b4	capsulate2.exe	C:\DOCUME~1\User\LOCALS~1\Temp\MMBPlayer\capsulate2.exe

- *Threads Created*

PId	Process Name	TId	Start	Start Mem	Win32 Start	Win32 Start Mem
0x260	csrss.exe	0x674	0x75b44616	MEM_IMAGE	0x0	MEM_PRIVATE
0x2b0	lsass.exe	0x4b0	0x7c810856	MEM_IMAGE	0x75738e06	MEM_IMAGE
0x2b0	lsass.exe	0x4c0	0x7c810856	MEM_IMAGE	0x77e76bf0	MEM_IMAGE
0x4b4	capsulate2.exe	0x5d0	0x7c810867	MEM_FREE	0x408f42	MEM_IMAGE

- *Verdict*

Auto Analysis Verdict
Undetected

- *Mutexes Created or Opened*

PId	Image Name	Address	Mutex Name
0x4ac	C:\TEST\sample.exe	0x438723	n 4.9.8.13
0x4ac	C:\TEST\sample.exe	0x6c983dc9	DirectMusicMasterClockMutex
0x4ac	C:\TEST\sample.exe	0x6d9afec2	DDrawWindowListMutex
0x4ac	C:\TEST\sample.exe	0x6d9affde	__DDrawExclMode__
0x4ac	C:\TEST\sample.exe	0x6d9b000a	__DDrawCheckExclMode__
0x4ac	C:\TEST\sample.exe	0x73786c76	DDrawWindowListMutex
0x4ac	C:\TEST\sample.exe	0x73786c84	DDrawDriverObjectListMutex
0x4ac	C:\TEST\sample.exe	0x73786dd7	__DDrawExclMode__
0x4ac	C:\TEST\sample.exe	0x73786e05	__DDrawCheckExclMode__
0x4ac	C:\TEST\sample.exe	0x73f11693	DirectSound DllMain mutex (0x000004AC)
0x4ac	C:\TEST\sample.exe	0x73f1ef39	DirectSound Administrator shared thread array (lock)

- *Events Created or Opened*

PId	Image Name	Address	Event Name
0x4ac	C:\TEST\sample.exe	0x77a89422	Global\crypt32LogoffEvent
0x4ac	C:\TEST\sample.exe	0x77de5f48	Global\SvcctrlStartEvent_A3752DX

Test result available at:
[*http://camas.comodo.com/cgibin/submit?file=11e32ff708bd964388fdef85362be5465791d3e062dca0ff5d19e963de227c73*](http://camas.comodo.com/cgibin/submit?file=11e32ff708bd964388fdef85362be5465791d3e062dca0ff5d19e963de227c73)

7 References

[1] M. Bailey, J. Oberheide, J. Andersen J, Z.M. Mao, F. Jahanian and J. Nazario, “Automated Classification and Analysis of Internet Malware”, Springer-Verlag Berlin Heidelberg, LNCS 4637, pp. 178–197, 2007.

[2] Fielding and M.J. Connor,” First Response Computer Virus Blocking”, Patent Application Publication, pp. 1-10, Sep. 2, 2004.

[3] N. Joukov and T. Chiueh, “Internet Worms as Internet-Wide Threat”, Stony Brook University, pp. 1-26, 2003.

[4] S. Ortolani, C. Giuffrida and G. Crispo, “Bait Your Hook, A Novel Detection Technique for Keyloggers”, Springer-Verlag Berlin Heidelberg, pp.199-207, 2010.

[5] Obiniyi and M.A. Umar, “Random Number Based Dynamic Securer Web Login”, The International Journal of Engineering and Science, Vol 2, pp.19-25, 2013.

[6] Herley and D.F. Encio, “How To Login From an Internet Cafe Without Worrying About Keyloggers”, Symposium on Usable Privacy and Security, (SOUPS) ’06.

[7] S. Chahrvin, “Keyloggers – your security nightmare”, Computer Fraud & Security, Vol7, pp. 10–11, 2007.

[8] T. Mikkonen, A. Taivalsaari ,”Web applications: spaghetti code for the 21st century”, Sixth International Conference on Software Engineering Research, Management and Applications, 319-328, Washington, DC, USA, 2008.

[9] Hadnagy, *Social Engineering: The Art of Human Hacking*, Wiley Publishing, USA, 10-39, 2010.

[10] Anti-Virus Comparative, File Detection Test of Malicious Software Report, pp. 4-9, 2014.