Keylogger Detection using Memory Forensic and Network Monitoring

4 authors:

Md Bayzid Ahmed
eGeneration
2 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Mohiuddin Shoikot
East West University (Bangladesh)
3 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Jafrul Hossain
2 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Anisur Rahman
Daffodil International University
2 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Keylogger Detection using Memory Forensic and Network Monitoring

Md Bayzid Ahmed
Computer Science & Engineering
East West University
Dhaka, Bangladesh

Mohiuddin Shoikot
Computer Science & Engineering
East West University
Dhaka, Bangladesh

Jafrul Hossain
Computer Science & Engineering
East West University
Dhaka, Bangladesh

Anisur Rahman
Computer Science & Engineering
East West University
Dhaka, Bangladesh

ABSTRACT

Human society is moving towards a life that is fully govern by automated system where every important event of our life is locked and protected by a 'String', known as password. Password protection is in high demand and researchers shown fervent interest to accomplish the same. Besides, the process of stealing information also evolving. Keystrokes monitoring by using keylogger is an advanced way to steal passwords and valuable data. As keylogger is an unprivileged program running on user-space, it could be injected through many different ways into a computer. Usually, keylogger is untraceable by the user and also undetectable by various known anti-viruses. Many cyber security specialists have proposed different methods for detection of this malicious program which includes API based detection method and network traffic monitoring system. But, with evolving technology, attackers have developed a new level of keylogger which is no longer easily detected though those conventional methods. This new level of keyloggers is capable of communicating with the eavesdropper without sending any attached file and uses volatile memory as a buffer. In this paper, we have proposed a memory analysis based detection method. This proposed method is capable of detecting such different type of logger and also works for on traditional one. With this method any regular user can detect any suspicious activity. And also it does not need any special permission from operating system. It was tested on Linux and Windows OS with satisfactory level of success.

General Terms

Your Memory forensic, packet tracing, network, mailing agent, architecture of keylogger, memory space, string matching, text searching.

Keywords

Keylogger, volatility, user space, memory forensic, wireshark, SMTP, HTTP, key strokes.

1. INTRODUCTION

Malwares are that software's that used to steal sensitive information, impede computer operation and gain access to private computer systems. In brief, we can say that malwares are willfully programmed to harm or exploit people's computers particularly which are connected to the web. Very few spying software are as popular as a keylogger in recent time. To know the term keylogger and how it works, it is necessary to know and understand the architecture of the operating system. Keystroke logger is a software or hardware based program/device which keeps the log or record of each and every keystroke that a user makes from a keyboard. Keystroke logging is a very common approach to steal valuable information from someone's computer. When a keylogger is installed into the end computer, the keyboard activity or keystrokes made by that host computer will be compromised. By doing this the attacker can easily capture the sensitive information (e.g., Username and password) of a particular user. Even though keylogger has been studied well previously but still in recent year's hackers use it for stealing valuable personal information. It has been drawing worldwide attention in recent years. In the year 2005 more than eight hackers accused of planning to hack a Japanese bank and steal more than 423 million dollars by installing a keylogger which reminds us of the strength of keylogger [1]. A majority of key logger shows no sign of any intrusion within the system allowing them to gain keystrokes without having knowledge of its actions except for the user who installed it. In this work, we are mainly focusing on software keylogger. As advanced level keylogger is undetectable through regular anti-virus, so detection mechanism for novel keylogger is in demand. This paper proposes such mechanism, that is capable to detect any kind of keylogger installed in the system. Almost all the keylogger creates a log file in the host computer to store the typed strokes, but we have found such a novel one. This kind of keylogger does not create any log file, it can use RAM as a buffer. This new feature makes a keylogger a very dangerous one. And the proposed work is also motivated by this particular feature. Following proposed work is executable by any user. And it is easy to work with.

2. BACKGROUND WORKS

A. Keylogger

Keyloggers are initially developed in the mid of 1970s. Then it was developed and used by the Soviet Union. They deployed a hardware keylogger targeting typewriters. It was termed as "selectric bug" and it measured the movements of print head of IBM Selectric typewriters via subtle influences on the regional magnetic field caused by the rotation and movements of the print head. Perry Kivolowitz was the man who wrote the early keylogger on November 17, 1983. Day by day the technology improves and keyloggers become stronger and also there are many different kinds of keyloggers in the market which are developed using different technologies.

B. Features of keylogger

Different keyloggers have different features. As we are improving in technologies, keyloggers are also enriching with so many different features. When it runs in Linux server environment then only keystrokes are logged. When it works with the windows environment then a lot more then keystrokes are logged. Here is a list of keyloggers features. Key stoke logging, keyboard record, application tracking, website visited, email log delivery, FTP delivery, invisible mode, easy to install and automatic setup.

C. Types of Architecture

The main problem of a keystroke logger is it is very difficult to categorize it by its pattern. There are so many different key loggers in this world and every individual key logger has its own pattern. On the other hand, if we take a look at computer viruses then we can notice that every virus has its own signature or pattern. So a number of models and techniques have been proposed to address the general problem of viruses or malicious software's. However, the specific problem of detecting keyloggers all existing solutions are unsatisfactory.

Software keystroke loggers are computer applications which is programmed to log the keystrokes generated by keyboard. It collects keystroke events, stores them in a remote location, and then transmits to the attacker who installed the keylogger. We found many different software keyloggers by searching for "keyloggers" using search engine. But according to their internal architecture we can categorize them in several categories.

Kernel based keyloggers obtain root access to hide itself in the operating system and intercept keystrokes that pass through the kernel. This kind of keyloggers are very powerful because they reside at the kernel level which makes them very difficult to detect [3]. Keyloggers that developed using this method also can act as a keyboard driver. To program a keylogger like this way is very complex to code.

API based keyloggers uses a hooking mechanism to capture the keyboard data. API based keyloggers hook keyboard APIs inside a running application. This keyloggers registers the keystrokes like normal applications. Windows APIs like GetAsyncKeyState(), GetForegroundWindow(), etc. are used to poll the state of the keyboard or to subscribe to keyboard events [2][3]. This method is the most common method that used to develop keyloggers.

Form grabbing based keeps the log of web form submission or retrieve authorization and login credentials from web. After filling and submitting the form when the user press of click enter button then the keylogger recognize it as a submit event and keeps logs of all the form data.

Memory Injection based keyloggers works by interchanging memory tables that are linked with system functions and the web browsers. By fixing the memory tables or injecting directly into memory, this method will be employed by malware authors to bypass Windows UAC (User Account Control). The Zeus and SpyEye Trojans use this technique completely.

3. RELATED WORKS

Many anti-viruses included keylogger detection mechanism in their program. But the problem is almost all of them uses a repository of signature that matches renowned and vastly used keylogger. User-space keylogger is developing almost every day with a unique pattern, that's why signature matching does not work anymore. Mugdhakolte proposed in "Unprivileged detection of user-space keyloggers" [4] process analysis based approach to detect such malicious program. Sending string through a developed user-space program to all the other running program and monitor their allocated memory size changing pattern,keylogger can be detected. Only keylogger will capture them. But this approach needs real-time monitoring and that is very difficult and time-consuming. The author developed a user-space running program to detect and warn from keylogger. The user has to run that program for a full system scan to detect the malware. But the problem arises within two scanning gap period. The victim still unprotected within that time.

Keylogger uses I/O for storing typed keystrokes through a keyboard. Stefano Ortolani proposed a novel technique for detection of keyloggers through I/O monitoring [5]. Within a controlled way sending strokes into the system and constantly monitor the behavioral changes made by I/O provides a better solution. Windows do not provide this kind of permission to the user level. It needed privileged API permission to get proper information. After injecting string through unprivileged API command, a pattern has been generated from the output. That help to find out suspicious activity.

Many cybercrime investigator and researcher came up with many processes to detect keylogger. But most of them are not able to construct any particular way to remove that. Mahak Arora came up with a unique software named "KeyLog Detector" [6] with a solution for removal. That program is able to find any threat or any kind of suspicious activity in the computer system, through a full phase scanning. If the scan delay is 5 secs, then the program will be able to find keylogger. But the problem remains same, within two scanning gap period system can be compromised.

One of the detection mechanism includes network traffic analysis. R SreeramSreenivas studied for finding a behavioral pattern in remote keylogger [7]. All the remote keylogger is capable of communicating with the attacker through information passing. They have to send files which contains typed keystrokes. He found a pattern that explains, keylogger has to communicate with through sending files continuously maintaining a fixed time duration. And also the file size is fixed within a range. With this pattern using traffic monitoring, any suspicious communication could be detected. But that does not help to find out the real culprit. Analysis proves an existence but that is unable to locate which process is behind it.

The best approach for message communication between sender and receiver is to send the data in such a way that without the original receiver no one should be able to read that. Stefano Ortolani proposed in his Keylogger Detection and Containment [8] that, injection of dummy keystrokes with original data create a possible solution for live together with an installed keylogger. He proposed a system that capable of creating dummy keystrokes from a specified repository called noise factory and inject with original strokes in a kernel. In order to mimic human-like strokes, a normalizer is being used. For injecting dummy strokes it uses Keybd_event(). Before reaching the graphical routines included in USER32.dll his proposed system is able to remove the noise from data. With this, all malicious applications are just eavesdropping on a noisy string. With a 10% CPU load, this process is an almost 100% successful one. His technique allows legitimate applications to recover the data transparently.

4. METHODOLOGY

Software based and API based keylogger hide itself using rootkit. Attacker can inject any keylogger with any regularly used application. When a user installs that particular application in host PC, attached key logger will be installed automatically. As the malware is working under another application, that won't be visible in task manager. In case of remote keylogger, process will take all the key strokes and store them in a log file. After every fixed time period that log file will be sent to a remote server by using a mail agent.

Keylogger is capable of creating such mailing agent in host pc. That particular log file will be attached to the mail. In maximum case mailing agent uses port 80 or port 587 to communicate. Usually created log file is also a hidden one. And file size is also maintaining a range. There is another software named "SYSDIG", using this software any kind of memory writing or hidden activity. But working with SYSDIG need higher level of knowledge about computer architecture. It will be very difficult to use this procedure by any normal user or nontechnical user.

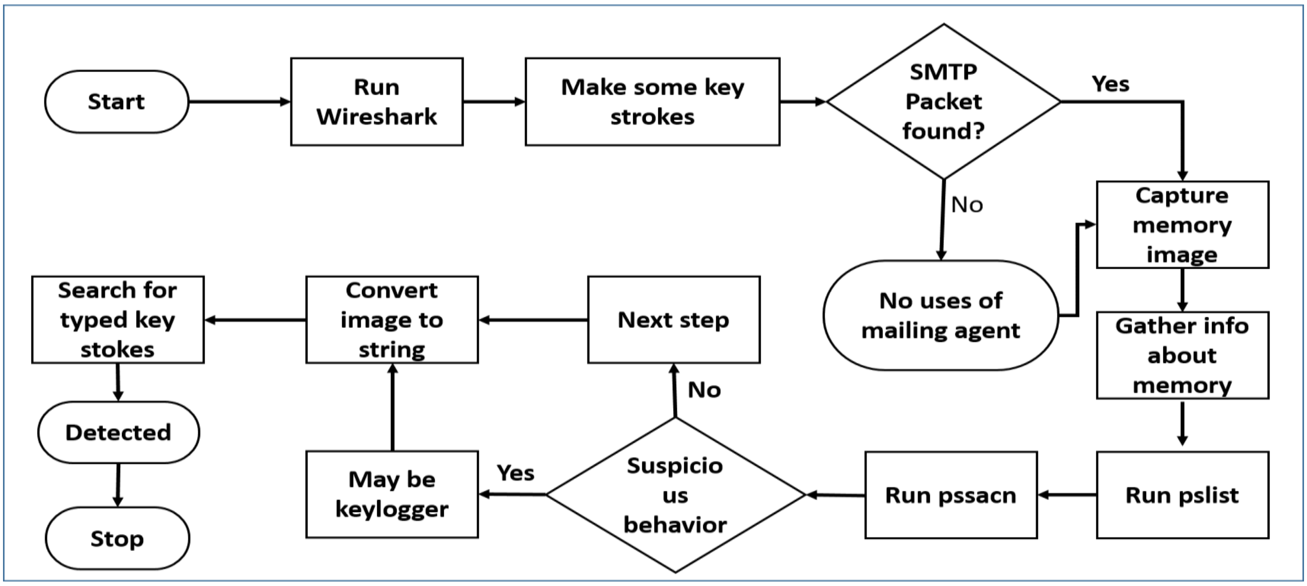


Figure 1:Flow chart - Keylogger detection using memory forensic and network monitoring.

This proposed method is a combination of Memory Forensic and Network Monitoring. As mentioned earlier there is such keylogger that does not need to create log file in host PC, it uses volatile memory as a buffer and after every specific time period it sent the stored keystrokes to a remote mail server. For finding the process behind this activity RAM data analysis is an option. And also by traffic monitoring detection of such suspicious program is possible.

Here is the pseudocode for the process to capture the keylogger:

1. Run only wireshark and no email.
2. Type something and remember that keyword.
3. Search for SMTP packet.
4. If no SMTP found, keylogger isn't using mail agent and GOTO 5.
5. Capture memory image.
6. Run "python vol.py –f ["File Name"] imageinfo". Capture profile name.
7. Run "python vol.py –f ["File Name"] profile=[profile name] pslist"
8. Run "python vol.py –f ["File Name"] profile=[profile name] psscan"
9. If that doesn't match that software may be keylogger or virus.
10. Run "cat [Filename]|strings|grep "[the word or sentence that was typed earlier]"
11. Try to find keyword like "key.backspace, key.enter, key.delete , [backspace],[delete]".

If any of these keyword with those word that was searched found in a string that means the existence of keylogger has found.

4.1 Memory Forensic

Memory Forensic, usually referred as memory analysis, is a process to analyze or deep digging in volatile memory of any system [9]. It allows investigator to search for suspicious malware in any computer or server. Since there exist some kind of malware that, they can hide themselves in such a way, they are completely untraceable. And also they could be the reason of compromised OS. Memory forensics are in highly demanded in this kind of situation. With this approach so many information could be retrieved through memory dump which is,

- All the running process information
- All executable file info with their history
- Network scanning (Packet, port, traffic)
- User logging info
- File data and their creation information

Monitoring on running processes is very difficult to analyze. As it relies on application it could create false data. For the whole process, by using any memory analysis software, a snapshot will be taken. Which is called the memory dump [10].

Advance level malware use modification of OS kernel for being undetected. For this reason, they have to use rootkits. There are two types of rootkits one is user level and another one is kernel mode rootkits. User level rootkits use modification of system libraries. In other cases, rootkits implement windows hooking, which is referring to execution in kernel internals. Through hooking mechanism access of API is possible, which should be permitted from user level [11]. With this process newly developed malware and eavesdropping viruses are making itself undetectable. They have to use volatile memory for their execution inside the system. Memory forensic is a very strong platform to identify this kind of suspicious object with a very deep searching mechanism. Volatile memory analyzing has been evolved to an advanced level. With an augmented model of memory forensic capture and subsequent analysis of a suspect memory is possible [12,13].

To take a memory snapshot of the victim's computer using DumpIt is the best option. A convenient tool from Comae Technologies, makes this very easy, even if the person in front of the affected computer isn't technical. This tool is a part of the free Comae Memory Toolkit. (An earlier version of this tool was distributed by MoonSols, which no longer makes it available.)

So, DumpIt has created a snapshot of the memory which was a RAW image file. Analysis of this raw image using Linux in more convenient. It can be analyzed in windows also. For this method a free memory forensic tool named as The Volatility Framework was used to examine the memory file's contents for malicious artifacts. Using Volatility scan the full process to get all the visible and hidden process with their ID. So now if a process was found that was not in pslist but in psscan that means this process is trying to hide something which is phishing. That process has to be monitored. However, there is another way to this. Use commend "python vol.py [File name].raw profile= Win10x64_10586 psxview". This command will show which Offset, Name, PID, pslist, psscan, thrdproc, pspcid, csrss, session,deskthrd&ExitTime. There will be a pattern for every process if any process does not match with this pattern that means that process is phishing. That process is trying to hide something.

We converted this raw image into strings because there will be binaries in this raw image which is difficult to breakdown. After converting into string we have done string matching with the word or sentence that was typed earlier. Using this command: "cat [Filename]|strings|grep "[the word or sentence that was typed earlier]""

With this command we can see those word or sentence along with Key.backspace or Key.delete or [backspace] or [delete].

4.2 Network Monitoring

We used Wireshark platform which is vastly popular for network monitoring and it is free & friendly to use. We ran Wireshark in host computer for 1hr without sending any email and type word or sentence with pressing the Backspace or Delete key in that computer and remember what was typed. We checked all the packets that has been generated within this 1hr through filtering HTTP or SMTP packets.

We had to look all the HTTP packets if there is an attached file in the packet. Then we know that there is some process who is sending files to hacker.

Now what if we didn't get any HTTP packets with file attached but we got SMTP packets. SMTP packets is used by mailing agent to send an auto generated e-mail. Gmail use SMTP to send e-mail from a computer to its server.

When packet is generated it uses SSL/TLS encryption. Without this encryption google will not accept any e-mail. This packet has to go to Port 587 and outgoing mail server is: smtp.gmail.com

If we want to look at these packets we can't see anything because it's encrypted. But we know that we did not used SMTP to email anyone. We use Browsers to mail which use HTTP packet. Finding a SMTP packet while running traffic monitoring software create suspicion about having malicious mailing agent in host PC.

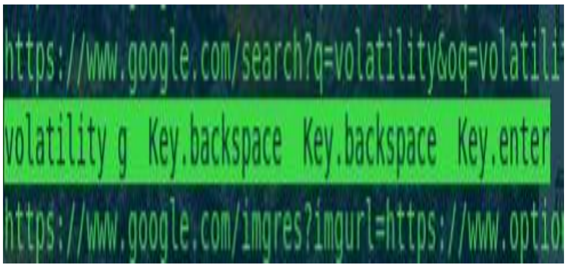
5. RESULT & ANALYSIS

To analyses the converted raw image, we used two operating systems. We used windows 10 to create ram image. We gather information using volatility.

Table 1. Testing system configuration and consumed time.

PC	RAM	OS	TIME
#1	8GB	Windows (64bit)	4.5 hrs.
#2	4GB	Windows (64bit)	55 min

Huge time difference between PC 1 &PC 2 because 8GB RAM contains more data than 4GB. So capturing the image information is the most time consuming process. After that run pslist, psscan, psxview, grep for string matching. When we used grep we converted the whole data into string with "string|grep". After that we search the desire keyword and got matched. While we were running this, we typed "volatility g" then pressed backspace twice for removing the g and the space between volatility and g then pressed enter. We were searching in google. After matching the strings, we found exact same data was in RAM. As we were searching this data should be in URL but not in RAM. Data was exactly same alongside with the buttons that was pressed. Two Backspace and one Enter. A snapshot is given in below.

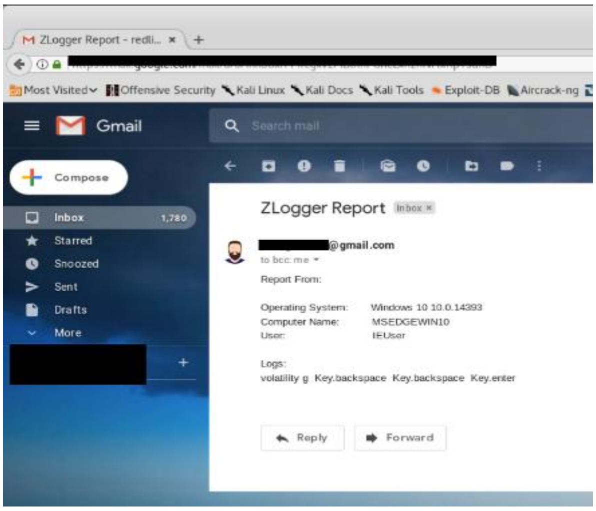


```
https://www.google.com/search?q=volatility&oq=volatili
volatility g  Key.backspace  Key.backspace  Key.enter
https://www.google.com/imgres?imgurl=https://www.optio
```

Fig 2: Matched string with the user made key strokes

This matching was running in Kali Linux in PC 1. For string matching we choose Linux system because it's easy to do string matching in here.

We also got email report from ZLogger. Snapshot is given below:



ZLogger Report Inbox ×

[redacted]@gmail.com
to bcc: me

Report From:

Operating System: Windows 10 10.0.14393
Computer Name: MSEDGEWIN10
User: IEUser

Logs:
volatility g Key.backspace Key.backspace Key.enter

Fig. 3. Emil report from ZLogger.

The whole process is tested on 3 different keylogger.

Table 2: Testing and results

Name	**Type**	**File sending method**	**Features**	**OS**	**Detection**
BeeLogger	SW	Gmail	Keyboard stokes	Cross platform	Yes
Zlogger	SW	Mailing agent	Keyboard strokes	Cross Platform	Yes
Radium-Keylogger	SW	Mailing agent	Keystrokes and screen shots	Windows	Yes

This three keylogger collected from github. We ran the test indivisually and successfully detect them.

6. CONCLUSION

Real time network monitoring can create an option to identify the running malicious process faster. If we can identify the culprit process earlier then we can work on removal. Still there is no valid process for removal of keylogger, many researchers have proposed detection mechanism but at the end the only solution is to format the system.

And also for searching in RAM will be easier if we use any searching mechanism or string matching program. The goal was to create a method for keylogger detection and make it easier for non-technical person. This proposed method does not need any special technical knowledge. Any regular person will be able to handle it.

7. REFERENCES

[1] Chien-Wei Hung, Fu-Hau Hsu*, Shih-Jen Chen, Chang-Kuo Tso, Yan-Ling Hwang, Po-Ching Lin, and Li-Pin Hsu, A QTE-based Solution to Keylogger Attacks, The Sixth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2012), Rome, Italy, August 19 - 24, 2012

[2] Rao, Ahsan. (2017). Detection of unprivilliged keylogger.

[3] Creutzburg, Reiner. (2017). The strange world of keyloggers - an overview, Part I. Electronic Imaging. 2017. 139-148. 10.2352/ISSN.2470-1173.2017.6.MOBMU-313.

[4] “Unprivileged detection of user space keyliggers” by Mugdha Kolte from MITCOE International Journal of Innovation Research in Science, Engineering and Technology

[5] Stefano Ortolani, Cristiano Guiffrida, Bruna Crispo: Bait Your Hook: A Novel Detection Technique for key loggers; S. Jha, R. Sommer, and C. Kreibich (Eds.): RAID 2010, LNCS 6307, pp. 198–217, 2010. c_Springer-Verlag Berlin Heidelberg 2010

[6] Mahak Arora, Kamal Kumar Sharma, Sharad Chauhan: Cyber Crime Combatting Using Keylog Detector Tool; Mahak Arora et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 3, Issue 2, June 2016, pp. 1-5

[7] R Sreeram Sreenivas, Dr R Anitha; Detecting keyloggers based on traffic analysis with periodic behavior; PSG College of Technology, Coimbatore, India

[8] Keylogger Detection and Containment by Stefani Ortolani(PHD Thesis) from Vrije Universiteit Amsterdam.

[9] Graeme Massina (2018, February 26), Computer Forensics: Memory Forensic, Retrieved from https://resources.infosecinstitute.com/category/computerforensics/introduction/areas-of-study/digital-forensics/memory-forensics/

[10] Eliézer Pereira (2017, June 1), RAM Memory Forensic Analysis, Retrieved from https://www.cybrary.it/0p3n/ram-memory-forensic-analysis/

[11] Limon, Gabriela. (2010). Forensic physical memory analysis: an overview of tools and techniques.

[12] Vidas, Timothy. (2006). The Acquisition and Analysis of Random Access Memory. J. Digital Forensic Practice. 1. 315-323. 10.1080/15567280701418171.

[13] Case, Andrew and Golden G. Richard. “Memory forensics: The path forward.” *Digital Investigation* 20 (2017): 23-3