A Novel Approach of Unprivileged Keylogger Detection

Ahsan Wajahat, *Azhar Imran, Jahanzaib Latif, Ahsan Nazir, Anas Bilal
School of Software Engineering,
Beijing University of Technology
Beijing, 100124, China
ahsan.sunny56@yahoo.com, azharimran63@gmail.com, zaib.chauhan@hotmail.com,
ahsan_ravian@hotmail.com, a.bilal19@yahoo.com

***Abstract*— Nowadays, computers are used everywhere to carry out daily routine tasks. The input devices i.e. keyboard or mouse are used to feed input to computers. The surveillance of input devices is much important as monitoring the users logging activity. A keylogger also referred as a keystroke logger, is a software or hardware device which monitors every keystroke typed by a user. Keylogger runs in the background that user cannot identify its presence. It can be used as monitoring software for parents to keep an eye on children activity on computers and for the owner to monitor their employees. A keylogger (which can be either spyware or software) is a kind of surveillance software that has the ability to store every keystroke in a log file. It is very dangerous for those systems which use their system for daily transaction purpose i.e. Online Banking Systems. A keylogger is a tool, made to save all the keystroke generated through the machine which sanctions hackers to steal sensitive information without user's intention. Privileged also relies on the access for both implementation and placement by Kernel keylogger, the entire message transmitted from the keyboard drivers, while the programmer simply relies on kernel level facilities that interrupt. This certainly needs a large power and expertise for real and error-free execution. However, it has been observed that 90% of the current keyloggers are running in userspace so they do not need any permission for execution. Our aim is focused on detecting userspace keylogger. Our intention is to forbid userspace keylogger from stealing confidential data and information. For this purpose, we use a strategy which is clearly based on detection manner techniques for userspace keyloggers, an essential category of malware packages. We intend to achieve this goal by matching I/O of all processes with some simulated activity of the user, and we assert detection in case the two are highly correlated. The rationale behind this is that the more powerful stream of keystrokes, the more I/O operations are required by the keylogger to log the keystrokes into the file.**

Keywords—Keylogger userspace; privileged; keystroke simulation; security.

I. INTRODUCTION

Keylogger sometimes refers as keystroke logger which is a surveillance type of the software or spyware that has an ability to s all the keystroke pressed by the user. Generally, keylogger is used for monitoring and malicious activity purposes. It is not a new device or software for the users. The first keylogger was installed in US embassy consulate buildings of various countries in 1970 for capturing the information to be used in a malicious manner by Petersburg [1]. There are several types of keylogger present in markets like hardware keylogger and software keylogger. In software, there are two main types of kernel mode keylogger and userspace.

It has been observed that 90% of current keylogger exist in userspace mode [2]. Userspace keylogger does not require any permission or authentication from the user for implementation [3]. Once you access the file it automatically runs and it will hide in your machine and inject the keystroke and you will never know that your information is being captured from some source [4]. An average skill full programmer can develop userspace keylogger. There are several reasons why the userspace keylogger is mass market version. Userspace keylogger relies on documented API's, commonly available on modern operating system i.e. Windows 7, 8, Mac OS 10 and plus etc. The user can mistakenly run the keylogger as an inoffensive piece of software and being cheated in running this file. In the result, the intruder will receive all the keystroke that pressed by the user and easily get his personal information and all other data. On the other side, kernel keylogger runs on the kernel mode. It has required all the permission for implementation execution from the user. The reported incidents due to keylogger are given in Table I.

Systematically, small dongles are added in the middle of Keyboard and the motherboard to save the entire logs that operator pressed keystrokes (must require for physical access). Meanwhile, keylogger software's support the hardware devices for the implementation of keyloggers [13]. Such software's are installed on the target machine which is responsible to detect the user actions by hidden and saved all the keystrokes (made at that time) along with some conditional statements by transferring to the third party. Recently all the new operating systems are designed to show familiarities with the groups of unprivileged application program interface (API) that possibly be utilized by client space projects to intrude on all the client keys [5].

Presently, different kinds of keyloggers run where any approval is not compulsory for the position and/or implementation. A common user remains unaware of keyloggers and overlooks by approaching it meaningless software and keyloggers implementation may be cheated by the intruders. If we compare it with Kernel keyloggers, it certainly needs a large power and expertise for a real and error-free execution [6]. It also reliant on access for both implementation and placement by Kernel keyloggers, the entire message transmitted from the keyboard drivers, and the programmer simply rely on kernel level facilities that interrupt.

978-1-5386-9509-8/19/$31.00 ©2019 IEEE

Table I. Reported Incidents of Keylogger

Year	Loss	Information
2009	$415K	The Criminals from Ukraine has stolen a country treasure's login credentials using keylogger. The fakes initialized a list of wire transfer each below than $10K limit that would have generated a closer examination by the local authorities.
2006	$580K	The Offenders delivered a Trojan Horse via email to some users of the online private bank (Nordea). Clients were betrayed to download and execute a "spam fighting" application which ensued to install a keylogger.
2006	$1M	User personal information, bank credentials and bank code of personal accounts was stolen using privacy-breaching malware. All their information
2006	$4.7M	A group of fifty-five peoples with juveniles had been arrested during the installation of a keylogger on a computer owned by unwitting Brazilians in the zone of Campina Grande. This keylogger was used to leak all necessary information including bank credentials to this gang.
2006	$13.9M	Though ineffectively attempts were made by the criminals to access "Sumitomo Mitsui Banks" computer system in London. This intrusion was possible just because of keylogger installation and recording administrative authorizations.

The 21st-century technology focusing keylogger as a very fast growing class of unauthorized software that has been used for gathering private information without authorizing of users. It is reasoned that unprivileged program running in user space which records all the keystroke is used by the user. These keystrokes are saved either in a log file or in any FTP server. It is being noticed that the lack of system protection includes un-updated anti-viruses and firewall makes the keyloggers very easier for plantation and execution. The intruder strategies to design the keylogger are based on infecting medium, type of target machine, a lifetime of keylogger etc. It is recently reported that the growth of keyloggers has been raised in various kind of crime wars. The growth of keylogger variants included in crimeware is discussed in Fig.1.

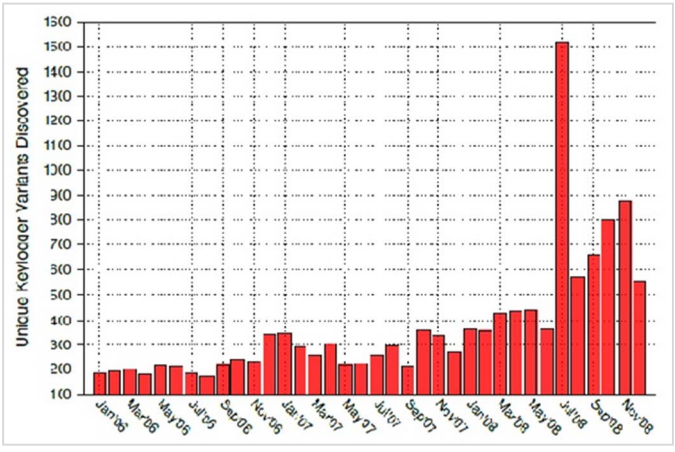

Fig. 1. Growth of Keylogger Variants in Crime Ware

II. LITERATURE REVIEW

A keylogger is a dangerous tool that can do numerous tasks. Normal security measure interface cannot defend the computer from keylogger attacks. Home to machine interface should be measured against a keylogger. The positive use of keylogger for employee and computer proprietors could be in some situation to improve security. But the effect of this positivity must be stable against the negative effect on user owner and employee [7]. Keylogger a tool made to save all the keystroke generates through the machine and offers the hacker the capability to take a huge amount of sensitive information without knowing the owner [16]. The general aim of this plain detect keylogger software and prevent sensitive information outflow. The keyloggers are distinguished utilizing Black-box strategy. Discovery methodology depends on behavioral attributes which can be connected to all keyloggers and it doesn't depend on the basic qualities of the keylogger. This anticipates means to create location framework on cellular telephones in light of machine learning calculation to identify keylogger applications [8].

Cyber-world is a vulnerable to different attacks, malware attacks are nasty one it is very hard to find and then protect. A keylogger contains both the script and spyware in a single program. The keylogger functionality is that it can store all the key has been pressed by the user and then save it in a log file, and then send this spyware email log file to the desired address.it is very dangerous for that system which is used in the regular business purpose i.e. banking system. Forbid from these attacks is very necessary [9].

For detecting a keylogger on a system a method is described. One descriptive personification creates; in the memory of the computer and hides in the window makes an exclusive, unpredictable data pattern. A user scans running Unpredictable data pattern and performs a secondary scan of a suspect process, the suspect process having an associated buffer that contains the unique, unpredictable data pattern [10].

Kernel malware usually makes insistent control flow adaptations e.g. hooks installation, in order to gain and preserve control [15]. The malware designers started to target function pointer in kernel data structure due to the fear of detection, particularly for vigorously assigned from the memory pools and heaps. The attack surface is very large and the function pointer modification is stealthy thus the attack is appealing to malware developers. More than 18,000 function pointers exist within the windows kernel [14]. Furthermore, to prove that this threat is truthful for the license-based operating system, we applied two possible attacks for windows by deploying two function pointers separately. Then, author purposed a novel proactive hook detection technique and develop a prototype, called Hook scout [11]. In present-day years, basic secret word based validation frameworks have logically demonstrated Insufficient for some sorts of genuine gadgets. Therefore, a great deal of examination begins working and Focus their endeavors on the outline of biometric verification frameworks. This pattern has been more accelerated along the approach of cell phones, which proposes a considerable measure of sensors and capacities to apply an assortment of portable biometric verification frameworks.

There is additionally some biometric route for verification yet assault turn out to be more refined and numerous other biometric methods have at last demonstrated lacking in face of cutting-edge aggressors practically speaking [12].

III. Research Methodology

In our methodology, we had arranged nonspecific keystrokes to convince keyloggers, keeping in view the end goal to figure out how to reenact keystroke which is been observed by the keylogger. To start with we should see how an OS create as well as handle keyboard events and second we additionally should be clear that keylogger capture keystroke in this procedure. Keystrokes generated by a keyboard is shown in Fig.2.

Windows XP 2006 OS makes a console hinder when a key was hit. Then an interrupt message has been sent out from the keyboard drivers and stores it into the system level message queue. Following the engaged application when the keyboard interrupt was produced, the working structure forward the message to the next application-level line of that particular desired application. Now it's the obligation of that application to control this key in like manner if the event with the working framework does not discover a particular Centered application. It just cancels that key. keyloggers utilize low-level operating system calls for example "GetKeyboardState" or "GetAsyncKeystate" to receive keystroke message or identify keyboard intrudes directly. So the keylogger is able to see everything whenever a key is pressed.

In our methodology, we used the source code of some well-known typical open source keylogger that is running on the platform of the Windows operating system such as key email version 7.0, Spybot version 1.2 and Morsa-keylogger version 1.8. We compiled and execute these source codes to get the result about their run-time feature. We found the relationship between dissimilar behaviors based on static and dynamic analysis. This behavior is generally generated by the keyloggers. Moreover, we disclosed the dodging mechanism which is often used by the static examination. We also observed that all keyloggers work on the same pattern. All keylogger firstly followed keystrokes and after that store them in a file or forward them to intruder over the internet by using electronic mail, or through FTP server. The primary distinction between these keyloggers is the planning that activated record access and correspondence exercises are performed i.e.

- The keystroke interrupted reached a certain amount.
- Special keys i.e. 'Enter' key were pressed.

After the execution of all these compiled source codes, we notice the keylogger such as key email using the Trigger condition produce more communication behaviors and file access. But these behaviors are comparatively less experimental when the various keyloggers such as Morsa and Spybot keylogger with the trigger condition were implemented. Thus, we concluded that two or three trigger conditions are used which provides capabilities to keylogger to escape the association based detection. Luckily our source code on C++ for keylogger is like a double edge sword. Other than concealing keylogger practices it effortlessly uncovered the nearness of the keylogger when high recurrence of keylogger particularly exceptional keystroke is squeezed by the client. That is the reason we utilize the keystroke recreation to build the discovery execution of the code. We exhibited the keylogger behavior by recording the input from keystrokes and by means of the output in which the I/O patterns formed by the keylogger. Moreover, we extend the proposed model with the capability to artificially inject keystroke patterns. We also explained the issues of selecting the finest input pattern to refine detection rate efficiently. Then, we presented the execution of our detection method on Windows, which is the most susceptible operating system for keylogger threats. To create an operating system independent architecture, we also provide the details of implementation for other operating systems. We effectively tested our proposed method on various public keyloggers. Here we can see the flowchart of our detection scheme in Fig.3.

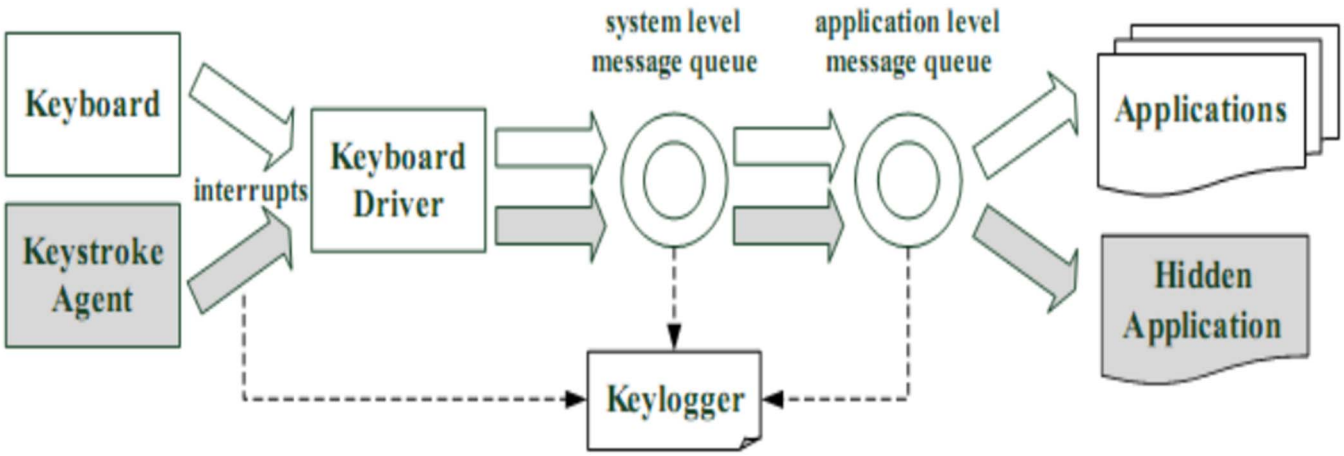

Fig. 2. Proposed Methodology of Unprivileged Keylogger Detection

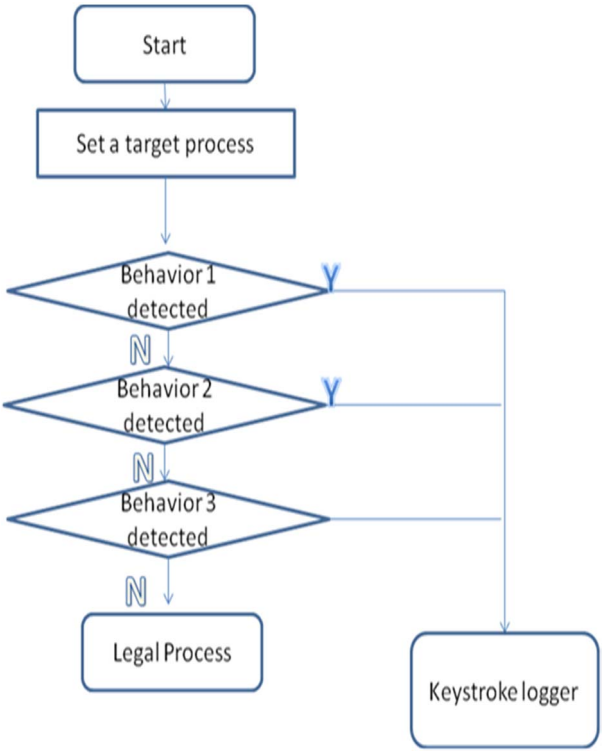

Fig. 3. Detection Scheme Flow Chart

IV. Experimental Design

As we already said our Research is explicitly focused on detecting the userspace keylogger for checking the ability and detecting the userspace keylogger. We experimented some of the top keylogger used as a monitoring software that is freely available on the internet just like Refog keylogger free, 5.4.1, Best-free keylogger 1.1, IwantSoft keylogger 3.3, and Actual keylogger 2.3 etc. Firstly we search out their source code and run them in C++ to find out what pattern and method they used to take keystrokes as well as how they transmit the information via the internet. We also observed using static examination that all keyloggers worked in the same way. All keylogger firstly followed keystrokes and after that store them in a file or forward them to intruder over the internet by using electronic mail, or through FTP server. The following snapshot given in Fig. 4 explains how a typical keylogger saves the keystrokes.

In Fig.4, it can be seen that keylogger save each and every keystroke we pressed. We typed "Information technology" the keylogger not only save the character are typed by the user. But can also save the special character that has been pressed from the keyboard.

To analyses the detection technique we implement prototype-based coded in C++ coded in 30 LOC, it can run in Windows OS as an unprivileged behavior based technique. It likewise gathers all the while all the procedures' I/O designs, in this way permitting us to dissect the entire framework in a solitary run. We use Dev C++ and visual studio for Running these line of code for experiment purposes.

Fig. 4. Generating Keystrokes

We used this vb.net code for monitoring our disk I/O via WMI this code tell us the all input and output services running on the machine. The refresher portion is really only needed if you're going to make multiple calls. Avoid having to execute the get Object code over and over again. The "no key" is the output of the WQL query.

Select Average disk queue length from win32_performattedData_Perfdisk_logicaldisk. We used this code in both normal when keylogger not installed and also used for troubleshooting when keylogger is installed on the machine. This C++ Code informs us the keystroke is typed by the user is directly communicating with the system level message queue or any interruption occurring if any interruption occurring then high chances that keylogger is installed in the machine.

V. Results

In the present study, it was revealed that the API functions invoke the I/O process to make screenshots during long sentence typing. Whereas, the keyloggers were properly installed in a computer/machine the API functions response could be noted very clearly. Conversely, the computer without keyloggers the API functions directly communicate with the motherboard of a system. It has been observed that the proposed technique outperformed others in unprivileged keylogger detection. The mechanism of response appeared either the key was pressed had involvement of system message queue or responding by the keyloggers. In result, we determined the keyloggers installation confirmation.

The graph shows the keylogger used in the evolution and summarization of obtained results as in Fig. 5 (a) and (b). All the keyloggers were detected without generating any false positive. A long window of observation can be detected using actual keylogger 2.3. In the sample, a general keylogger was used to save keystroke in memory and send to the disk regularly at different intervals. In some cases, it was noted that the keystrokes were saved in memory, however, they were sent to the disk as soon as the keylogger found the change in focus. Such response appeared in Virtuoza free keylogger 2.0. Withal, to deal with this regular procedure, when a sample is injected our detection framework implements a change in focus each time.

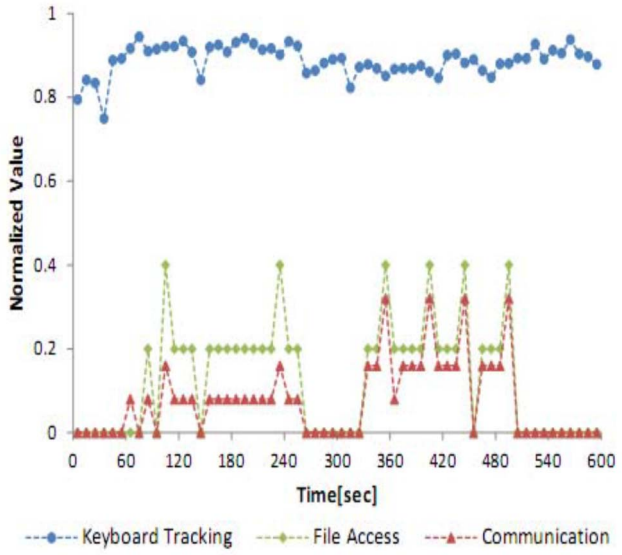

Fig. 5. (a) API Function without Keystroke Simulation

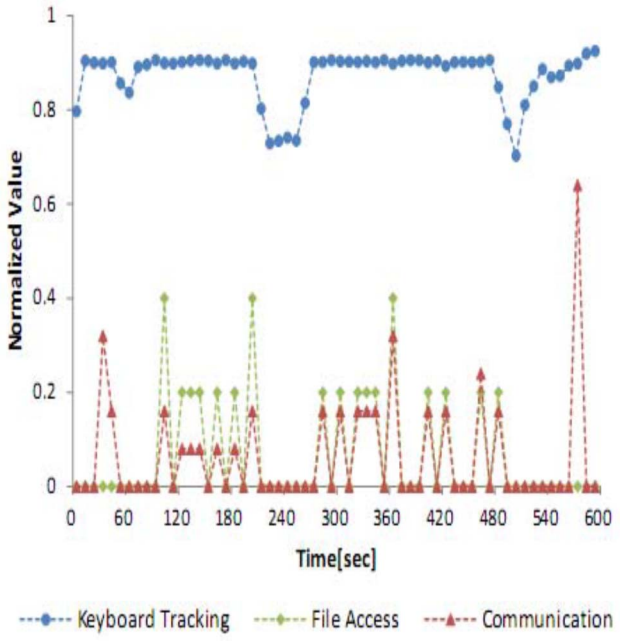

Fig. 5. (b) API Function with Keystroke Simulation

A. Detection Results

Table II shows the list of the Keyloggers used for evaluation purpose. Notes are also given to track buffering either time-based or focus based. The keystroke agent we executed in our research can make Spybot achieve more communication behaviors and file access as given in Fig. 5. The Spybot detection performance of the DCA has been improved to some extent using simulation method, as described in Table II.

Table II. Detected Keyloggers

Keylogger Type	Version	Finding	Records
Refog	5.4.1	✓	Focus-based buffering
Best free	1.1	✓	---
I want soft	3.0	✓	---
Actual	2.9	✓	Focus-based buffering
Revealer	1.4	✓	Focus-based buffering
Virtuoza	2.0	✓	Time-based buffering
Quick	5.0	✓	---

VI. CONCLUSION

The present study was focused on the detection of most common unprivileged userspace keylogger. In this research, we presented a C++ Code that permits the client to live respectively with keylogging malware without placing his protection in danger. In our study, we have displayed the keylogger response by matching the input from keystrokes and with the output i.e., I/O designs that are delivered by the keylogger. We effectively evaluated our codes in contrast to the most commonly recognized free keyloggers without any false positives and/or negatives false were reported. Our codes officially allow the legal API to retrieve their actual data while revealing the keyloggers to actual noisy stream. We have been achieved state of the art results using the proposed method.

REFERENCES

[1] http://www.cryptomuseum.com/covert/bugs/selectric/index.htm.

[2] N. Grebennikov, "Keyloggers: How they work and how to detect them," http://www.viruslist.com/en/analysis?pubid= 204791931.

[3] San Jose Mercury News, "Kinkois spyware case highlights the risk of public internet terminals," http://www.siliconvalley.com/mld/siliconvalley/news/6359407.htm.

[4] N. Strahija, "Student charged after college computers hacked," http://www.xatrix.org/article2641.html.

[5] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," ACM Trans. on Information and System Security, vol. 13, no. 1, pp. 1–26, 2009.

[6] J. Rutkowska, "Subverting vista kernel for fun and profit," Black Hat Briefings, 2007.

[7] Seref sagiroglu and gurol canbek increasing Threats to Computer Security and PrivacyDigital Object Identifier 10.1109/MTS.2009.934159.

[8] Gunalakshmii IEEE Transactions on Parallel and Distributed Systems 25(1):53-63DOI: 10.1109/TPDS.2013.27

[9] A framework for detection and prevention of novel keylogger spyware attack

[10] Jun Fu and Huan Yang Enhancing Keylogger Detection Performance of the Dendritic Cell Algorithm by an Enticement Strategy The 28th Research Institute of China Electronics Technology Group Corporation.

[11] Tyagi, Gaurav, Khaleel Ahmad, and M. N. Doja. "A novel framework for password securing system from key-logger spyware", 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014.

[12] Valeriu - Daniel Stanciu, Riccardo Spolaor, Mauro Conti, Cristiano Giuffrida In Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy (CODASPY '16) March 9-11, New Orleans, LA, US.

[13] Stefano Ortolani, Cristiano Giuffrida, And Bruno Crispo, Senior Member,IEEE; IEEE Transactions On Dependable And Secure Computing, Vol. 10, No. 1, January-February 2013.

[14] Evangelos Ladkis1,Lazaros Kormilas2; You Can Type But You Can't Hide: A Stealthy GPU Based Keylogger,1Institute Of Computer Science, Foundation for Research And Technology,Hellas, Greece; 2Columbia University, USA,.

[15] Stefano Ortolani1,Bruno Crispo2,Bait Your Hook : A Novel Detection Technique For Keyloggers,1Vrije University. De Boelelaan 1080,1081HV Amsterdam, Netherland; 2University of Trento , vai Sommarive.

[16] N. Patterson and M. Hobbs, "Virtual World Security Inspection", Journal of Networks, Vol. 7, No. 6, 2012, pp. 895–907.

[17] T. Holz, M. Engelberth, and F. Freiling, "Learning More about the Underground Economy: A Case-Study of Keyloggers and Dropzones", in Proc. of 14th European Symposium on Research in Computer Security, 2009, pp. 1–18.