ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

Automatic Detection System for User Space Keylogger

Ankit Waikar[1] , Siddhi Gaikwad[2], Yash Nikam[3] , P.P.Bastawade[4]

Student, Dept. of Computer Engineering, AISSMS College, Pune, India.[1]

Student, Dept. of Computer Engineering, AISSMS College, Pune, India.[2]

Student, Dept. of Computer Engineering, AISSMS College, Pune, India.[3]

Lecturer, Dept. of Computer Engineering, AISSMS College,Pune, India.[4]

ABSTRACT*:* Keyloggers are a invasive software often used to harvest secret information. One of the main reasons for this fast growth is the possibility for unprivileged programs running in the user space to secretly steal and record all the keystrokes typed by the users on a system. The ability to run in unprivileged mode makes possible their implementation and distribution. but, at the same time, allows one to understand and imitate their behavior in detail. Overviewing this characteristic, we proposed a new spying as well as catching technique that traps carefully typed keystroke sequences in input and track the behavior of the keylogger in output to detect it among all the other running processes.Our technique is an unprivileged application, hence matching the ease of deployment similar as of a keylogger executing in unprivileged mode. We have examined the implicit technique against the most common keyloggers. This confirms the practicatlity of our approach in practical scenarios.According to extensive experimental results confirm that our technique is robust to both false positives and false negatives in day to day environment.

KEYWORDS: Invasive Software, Keylogger, Security, Black-box, PCC.

I. INTRODUCTION

KEYLOGGERS are deployed on an machine with an intentionally monitor the user activity by logging keystrokes and eventually send them to the person who deployed it [1]. While they are often used for legitimate purposes (e.g., surveillance/parental monitoring infrastructures), keyloggers are often a hidden unknown hazardous exploited by attackers to steal personal information. Many credit card detail and passwords have been stole using keyloggers [2], [3], which makes them one of the most harmful types of spyware known to date. Keyloggers can be implemented as small hardware devices or in software. Software-based keyloggers can be further differentiate based on the permissions they require to execute. Keyloggers implemented by a kernel module run with full permissions in kernel space. Conversely, an unprivileged keylogger can be deployed by a user-space process. It is necessary to notice that a user-space keylogger can easily depended on documented sets of unprivileged APIs which are commonly available on modern operating systems. This is not the case found for a keylogger implemented as a kernel module. In kernel space, the programmer must be rely on kernel-level facilities to cut off all the messages release by the keyboard driver, without doubt requiring a measurable amount effort and knowledge for an effective and bug-free implementation A user can say that the keylogger is a harmless software and being deceived in executing it. On the contrary, kernel-space keyloggers needs a user with super-user permission to knowingly install and execute unsigned code within the kernel, a practice often not permitted by modern operating systems such Windows Vista or Windows 7. Today more than 95% of the existing keyloggers are running in user space.[4] In this paper, we propose a new approach to detect and kill keyloggers running as unprivileged user-space processes which can be harmful .Our technique isbeing entirely implemented in an unprivileged process. As a result, our solution is much more portable, unintrusive, simple to install, and yet much more effective. In addition, the proposed detection technique is fully blackbox, i.e., based on behavioral characteristics

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

similar to all keyloggers. In other words, our technique does not depend on the internal structure of the keylogger or the particular set of API'sbeing used. For this reason, Our solution is general applicability. We have prototyped our approach and tested it against the most common freekeylogger.

II. PROBLEM DEFINITION

Keyloggers are implemented on a system to purposely monitor the user activity, by logging keystroke and delivering them to a third party. This System propose a new approach to detect keyloggers running as unprivileged processes. To match the model, this technique is fully implemented in an unprivileged process. As a result, the solution is portable, easy to install, and yet very effective. The proposed detection technique is completely black-box which is based on behavioral characteristics of keyloggers. In other words, this technique does not rely on the internal structure of the keylogger or the particular set of APIs used for this reason; the solution is of general applicability. This system prototypes and is evaluated it against the most common free keyloggers. This approach has proven effective in all the cases. Key loggers are increasing rapidly and are the number one threat on the internet.

III. ARCHITECTURE OF SYSTEM

In this System, A keylogger can be lodged in a hardware device which plugs it into the keyboard port on computer. Some keyloggers are hidden inside of keyboards of themselves. Hardware keyloggers cannot be detected by software, but they have the disadvantage of gaining the physical access to a computer. If you pretend that the hardware keylogger is present on your system, examining keyboard's connection , or replacing the keyboard will also solve the problem. Form-filling software such as Rob form stores passwords, credit card info, and other information in a database, then enters it into Web forms as needed. This proposed system eliminates the user's need to type any such data on the keyboard, and can prevent keyloggers from recording it. However, there are other forms of software which can catch data posted to forms by form-fillers. The Speech-to-text software or virtual keyboards can eliminate the keyboard connection, too. Since , the text has to get to its terminal , and that path may be unprotected to canny keystroke loggers. An anti-keylogger software attempts to detect or disable keylogging programs. Anti-keyloggers scan your hard drive for the digital signatures of known keyloggers, and look for low-level software "hooks" that indicate the presence of a keystroke grabber. Anti-keyloggers are most effective against keyloggers than antivirus programs because they often don't identify keyloggers as an malware

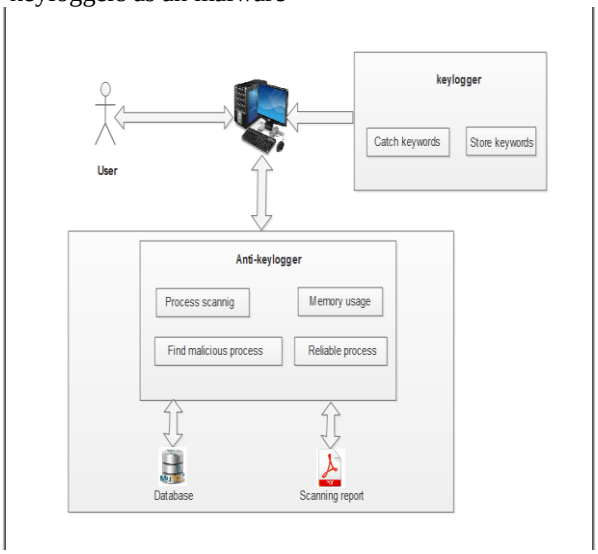

Fig: *System Architecture*

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

Methods used for detection

Keylogger can detected by using two methods:

1. Signature based detection
2. Heuristic based detection

the thing is that keylogger developers usually depend on a known methods to make their vicious code, and that allows researcher to find and detect them quickly. Such methods are for example:

- Using Send Keys
- Using Monitoring Process Usage
- Using Reliable and Malicious Database

Monitoring Process Usage:

Running key loggers temporary store all its received keys in its process memory.
Due to this its process memory keeps changing. So our software will keep record of all the process and memory being used.

- Sent Keys:

Lies within the Windows Script Host's object model is a tiny but important method called sends Keys that permit you to the send keystrokes to the active window just as if you have typed them on the keyboard .

- Monitoring Process Usage:

Running key loggers temporary store all its received keys in its process memory.
Due to this its process memory keeps changing.As we keep track of process and memory

- Reliable and Malicious Database:

Our Anti-keylogger application will keep our own Databases for all vicious or malicious Process and Reliable Process. After executing our Application it checks databases against all users running processes. If some records are been recorded than the result is displayed otherwise further steps are executed.

IV. LITERATURE SURVEY

The approach of this paper is focused on designing a detection technique for unprivileged user-space keyloggers. Unlike other categories of keyloggers, a user-space keylogger is a background process which registers operating system supported hooks to surreptitiously spy every keystroke issued by the user into the current foreground application. Our goal is to prevent user-space key loggers from stealing confidential data originally intended for a (trusted) authentic foreground application. Malicious foreground applications surreptitiously logging user-issued keystrokes and application-specific keylogger are outside our threat model and can be unidentified using our detection technique. This model is based on these observations and explores the possibility of isolating the keylogger in a controlled environment, where its behavior is directly exposed to the detection system. Techniques are used for controlling the keystroke events that the keylogger receives in input, and constantly monitoring the I/O activity generated by the keylogger in output. To assert detection, leverage the intuition that the relationship between the input and output of the controlled environment can be modeled for most keyloggers with very good approximation.
The key advantage of the approach is centered on a black-box model that completely ignores the keylogger internals.

V. IMPLEMENTATION

The figure shows the result of (a) The loging page which verify's the authentication of user (b) The Reslts before scanning the system using the software (c) The results generated after scanning the system using the software if a keylogger is found then the system asks the user whether to kill the keylogger or the suspicious process.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 9, Issue 3, March 2020

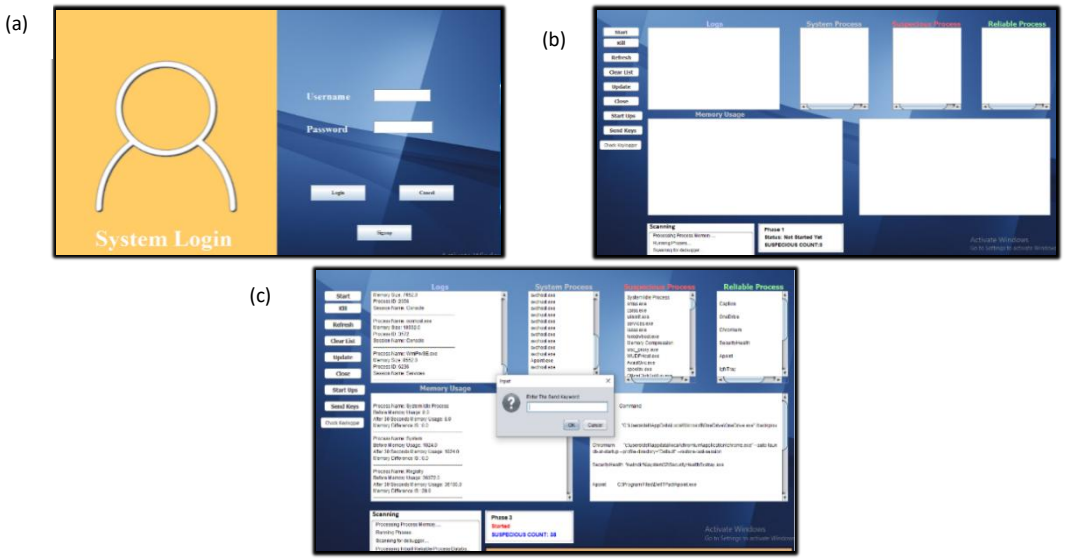

Fig.Implementation of project (a)Log in page used for Authentication of user(using valid credential) (b)Before scanning the device using the software (c)After scanning the device using the software

VI. CONCLUSION

Through this paper, we presented an unprivileged black-box approach for precise detection of the most common used keyloggers, (user-space keyloggers). We analyzed the behavior of a keylogger by related input (i.e., the keystrokes) with the output (i.e., the I/O)patterns produced by the keylogger). We increased our model with the ability to naturally inject the keystroke patterns. We also discussed the problem of selecting the best input pattern to improve our detection rate.

REFERENCES

[1] T. Holz, M. Engelberth, and F. Freiling, "Learning more about the underground economy: A case-study of keyloggers and dropzones," Proc. of the 14th European Symposium on Research in Computer Security, pp. 1–18, 2009.
[2] San Jose Mercury News, "Kinkois spyware case highlights risk of public internet terminals," http://www.siliconvalley.com/mld/siliconvalley/news/6359407.html.
[3] N. Strahija, "Student charged after college computers hacked," http://www.xatrix.org/article2641.html.
[4] N. Grebennikov, "Keyloggers: How they work and how to detect them," http://www.viruslist.com/en/analysis?pubid= 204791931.
[5] Security Technology Ltd., "Testing and reviews of keyloggers, monitoring products and spyware," http://www.keylogger.org.
[6] 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.
[7] M. Vuagnoux and S. Pasini, "Compromising electromagnetic emanations of wired and wireless keyboards," Proc. of the 18th USENIX Security Symposium, pp. 1–16, 2009.
[8] J. Rutkowska, "Subverting vista kernel for fun and profit," Black Hat Briefings, 2007.
[9] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," The American Statistician, vol. 42, no. 1, pp. 59–66, feb 1988.
[10] Sarah Young (14 September 2005). "Researchers recover typed text using audio recording of keystrokes". UC Berkeley NewsCenter.
[11] Sharon A. Maneki. "Learning from the Enemy: The GUNMAN Project" Archived 2017-12-03 at the Wayback Machine. 2012
[12] Christopher Ciabarra (2009-06-10). "Anti Keylogger". Networkintercept.com. Archived from the original on 2010-06-26