International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 3 Issue 4 April 2016

An Approach to Disclose the Existence of Keylogger

Sneha Ann Chandy[1], Anitha Jose[2]
Department of Computer Science and Engineering, College of Engineering, Kallooppara[1, 2]
chandysneha@gmail.com[1], joseani79@gmail.com[2]

Abstract: *The number of malignant applications targeting internet banking transactions has incremented dramatically. This represents a challenge not only to the customers who utilize such facilities, but also to the institutions who offer them. These malignant applications make utilization of two kinds of assailment vector - local attacks which take place on the local computer, and remote attacks, which redirect the victim to a remote site. Keystroke capturing is one among such attacks. Evasive software keyloggers conceal their malicious behaviours to defeat run-time detection. This paper proposes an algorithm known as Dendritic Cell Algorithm (DCA) that uses an induction-correlation framework to detect the presence of Keyloggers. It also encrypts the log file which contains all the keystrokes captured making it useless when viewed by attacker thus providing added protection.*

Keywords: *keylogger, keystroke simulation, dendritic cell algorithm (DCA), induction, correlation.*

1. INTRODUCTION

Recently, financial services providers are faced with complex challenges that affect their very survival in a high churn market. Protecting sensitive and critical data, no matter where it resides should be a core requisite of every company's security strategy. Number of users who uses internet services such as online banking, social networks, e-mails, etc has increased, the number of fraudulent activities also increased. Commonly used attacks are phishing, malwares, keystroke capturing/logging, SQL injection etc...

Keyloggers are gaining more attention in recent days. Keystroke logging, also known as keylogging or keyboard capturing, is the action of recording (or logging) the key struck on a keyboard, typically in a covert manner so that the person using the keyboard is unaware that their actions are being monitored.

There are mainly two types of keyloggers, Hardware keyloggers and Software keyloggers. Hardware keyloggers are small electronic devices that are used for capturing the data in between a keyboard device and an I/O port. Software keyloggers are softwares that track systems, accumulate keystroke data within the target operating system, store them on disk or in remote locations, and send them to the assailant who installed the keyloggers in authentic time or later. The main advantage of software keyloggers is that they can run for an indefinite duration and the accumulated information will be transmitted to the attacker. It eliminates the need of personally obtaining the information.

This paper proposes an algorithm known as Dendritic Cell Algorithm to detect the presence of keyloggers within a system and a method to obviate keylogger attack. It uses an induction-correlation framework which correlates some behaviour exhibited by each running applications in a system.

2. PROPOSED SYSTEM

Unlike other type of malicious program, keyloggers present no threat to the system itself. They can cause a serious threat to users. They can be used to intercept password and other confidential information entered via the keyboard. Keeping a keylogger off your machine is trillion time more important than the strength of any of your password. Antivirus software's could detect and block many kinds of keyloggers, but there is no assurance that it gets everything.

There are various methods to detect keyloggers:

- Signature based analysis
- Heuristic analysis
- Immune / Behaviour analysis

Anti-virus software's uses signature based method but they only scan for known signatures. They have nothing to do with novel keyloggers as well as private keyloggers. Heuristic method is based on the piece-by-piece examination of a virus, probing for sequences of instructions that differentiate the virus from normal programs. It has some impuissance. The length of time scan takes is longer than other types. An increased number of false positives can occur depending upon data.

To overcome these problems, security experts are trying to use behaviour based detection techniques, that analyze API calls of a process to classify it as a keylogger or not.

Existing methods relies on single behaviour that has a high rate of false positives (FP) [6]. The ability to correlate multiple behaviours (keystroke tracking, file access and network communication) helps to reduce FP rate to a great extend. This paper proposes an immune inspired algorithm known as dendritic cell algorithm (DCA) for the purpose of improving the detection performance. Rather than relying on a single type of API (Keylogging APIs), it has the ability to correlate multiple types of API (keystroke tracking, file access and network communication APIs).

The DCA is based on an abstract model of the dendritic cells which are natural intrusion detection agents of the human body [1]. These cells collect antigens and signals (environmental conditions of the antigens), and combine the evidence of damage (signals) with the acquired suspicious antigen to provide information about how perilous a particular antigen is [1]. DCA not only detects an anomaly, but also the culprit responsible for it.

Following are the 5 input signals. 1) PAMPs and safe signal-2, derived from the frequency of invocations of keystroke tracking functions, 2) danger signal-1, the time difference between two consecutive WriteFile calls, 3) danger signal-2, the relation between different categories of function calls, 4) safe signal-2, the time difference between two outgoing consecutive functions. Antigens are defined as the process (identified by Process ID) which causes the calls. These antigens are correlated with the input signals by the DCA, resulting in a pairing between signal evidences and antigen suspects, and in the end, the identification of the keylogger process.

2.1 Keylogger detection

All keyloggers works in a like manner. They all firstly track keystrokes and then wrote them to a file or send them to a destination via the Internet.

This paper proposes solution to detect software keyloggers on a host. It consists of two steps: the induction of the keyloggers and the correlation of the behaviours exhibited by them [3]. In real environment, the frequency of keystrokes may not be high, and thus the behaviours of the keylogger are not evident enough. Therefore, by designing a keystroke agent application to frequently generate random keystrokes will purge this problem. As a result, the performance of the keylogger will be more obvious in the stimulation of a large number of random keystrokes in a short time. The keystroke agent holds the simulated keystrokes within a hidden application to avoid them passing to the other applications. It will ensure that the normal applications remain unaffected by the synthesized keystrokes. The framework of our approach is shown in fig. 1.

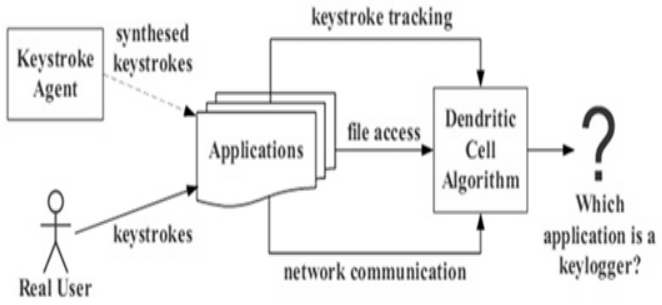

Fig.1 The induction-correlation framework for software keylogger detection by the DCA.

Induction Phase: Induction phase stimulates virtual keystrokes to induce keyloggers. Fig.2 indicates a Windows NT operating system generates a keyboard interrupt when a key is pressed. The keyboard driver transforms this interrupt to a system defined message and then puts it into the system level message queue [1]. The operating system passes this message to the application level message queue of that

specified focused application [1]. In this process, keyloggers employ very low level operating system calls, such as GetKeyBoardState or GetAsyncKeyState, to intercept keystroke messages [1]. So the keyloggers perceive everything whenever a key is pressed.

The keystroke agent simulates keyboard event completely by invoking system kernel (keybd_event). A keylogger tracks keystrokes from all applications, it will not be able to distinguish the simulated keystrokes from the authentic keystrokes. When keystrokes are generated frequently, the keylogger has to perform more file access and communication behaviours in order to log/ send plentiful keys.

Correlation phase: DCA correlates API calls generated by all running applications to identify the keylogger application. In order to obtain the API calls, implement a hook program to monitor three types of function calls:

- Keystroke Tracking: GetKeyboardState, GetAsyncKeyState and GetKeyNameText
- File Access: CreateFile, OpenFile, ReadFile and WriteFile
- Communication: send, recv, socket, sendto and recvfrom

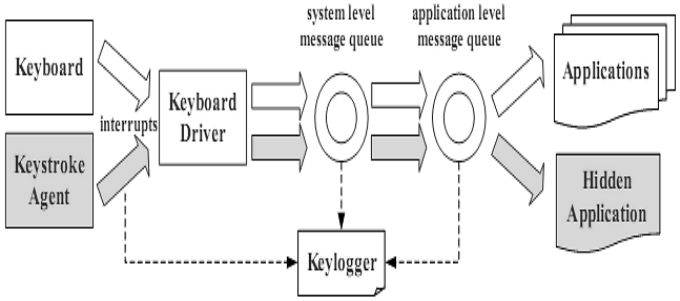

Fig. 2 How keystrokes are handled by a Windows NT operating system and intercepted by a keylogger.

Though these API functions are often employed by keyloggers to implement their keylogging, it may form part of legitimate usage. Therefore, a keenly intellective correlation method is required to determine the invocations of such functions are anomalous or not.

Five signals, namely one PAMP signal (PAMP), two danger signals (DS) and two safe signals (SS), are used for the input of the DCA.

PAMP is a signature based signal derived from the rate of keyboard tracking function calls. An immense number of these function calls signify the potential existence of a keylogger.

Danger signal is a quantification of an attribute which increases in value to designate an abnormality. Low values of this signal indicate that it may not be anomalous. DS-1 is derived from the time difference ($\Delta t1$) between two consecutive WriteFile function calls. Because a keylogger preserves the keystrokes captured to log files perpetually, a minute $\Delta t1$ will be observed. DS-2 is derived from the correlation between different categories of function calls.

Based on the behavioral characteristics of keyloggers, it engenders this signal when file access or communication functions are invoked shortly after the invocations of keyboard tracking functions.

Safe signal is a confident indicator of normal or steady state system behaviour. This signal is used to counteract the effects of PAMP and danger signals. SS-1 is derived from the time difference ($\Delta t2$) between two outgoing consecutive communication functions including send, sendto and socket functions. SS-2 is derived from the small amount of the keyboard tracking function calls within a specified time-window. Legitimate applications such as notepad or WordPad invoke much fewer keyboard tracking functions than keyloggers. So, small amount of invocations is considered to be safe in the host.

Antigens are potential culprits responsible for any observed changes in the status of the system. As any process executed one of the selected API functions, the process id (PID) which causes the calls and thus generates signals is defined as antigens. Observing which processes are active when signal context is danger, the DCA can find the existing keylogger in the system.

2.2 Keylogger Prevention

After the detection of keylogger the work of prevention is started. It inspects every log file which contains the activities

made by the keylogger software. It runs an encryption program on the log file so that the person cannot interpret its contents who receive/view the log.

3. CONCLUSION

This paper proposed an induction-correlation framework for keylogger detection. Keystroke simulation raises the frequency of the keystrokes, and thus induces keyloggers to produce more malicious behaviours. Then the amplified behaviours are correlated by the DCA to find the keylogger process at the earliest to reduce the loss of privacies. It uses an encryption algorithm to encrypt the recorded keystrokes present in the log file.

ACKNOWLEDGMENTS

We would like to thank, first and foremost, Almighty God, without his support this work would not have been possible. We would also like to thank all the faculty members of college of engineering Kallooppara for their immense support.

REFERENCES

[1]. Jun Fu and Huan Yang and Yiwen Liang and Chengyu Tan, "Enhancing Keylogger Detection Performance of the Dendritic Cell Algorithm by an enticement Strategy", Journal Of Computers, Vol. 9, No. 6, 2014.
[2]. Rajeshree Khande and Dr. Yashwant Patil, "Online Banking In India: Attacks And Preventive Measures To Minimize The Risk", IEEE, 2014.
[3]. Jun Fu, Yiwen Liang, Chengyu Tan, Xiaofei Xiong, "Detecting Software Keyloggers with Dendritic Cell Algorithm", International Conference on Communications and Mobile Computing, 2010.
[4]. Greensmith, Uwe Aickelin, and Jamie Twycross, "Articulation and Clarification of the Dendritic Cell Algorithm", 2009.
[5]. Suchita Yadav, Ravi Randale,"Detection and Prevention of Keylogger Spyware Attack", International Journal of Advance Foundation And Research In Science and Engineering (IJAFRSE), Vol. 1, 2015.
[6]. Jun Fu, Huan Yang, Yiwen Liang , Chengyu Tan, " Mimicking User Keystrokes to Detect Keyloggers with Dendritic Cell Algorithm ", International Workshop on Cloud Computing and Information Security, 2013.
[7]. Roger Mayer,"Secure Authentication on the Internet", SANS Institute, 2007.
[8]. Serge V.Krasavin,"Keyloggers-content monitoring exploits", SANS Institute,2005.
[9]. Qinghua Zhang, "Study on Fraud Risk Prevention of Online Banks", International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009.
[10]. Ioannis Koskosas, "The Pros and Cons of Internet Banking : A Short Review", Business Excellence and Management, Vol 1, 2011.
[11]. Ruby Shukla, Pankaj Shukla, "E-Banking : Problems and Prospects", International Journal of Management and Business Studies, Vol 1, 2011.
[12]. http://www.sans.org/readingroom.