Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 5, May 2015, pg.914 – 922

RESEARCH ARTICLE

Adulterate -Bot Detection & Espionage Monitoring System

Prakash Chandra Shukla[1], Asst. Prof. Yashpal Singh[2], Prof. S.Niranjan[3]

[1]GANGA INSTITUTE OF TECHNOLOGY AND MANAGEMENT (CSE) MTECH, (MDU); JHAJJAR

[2]DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

[3]DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

[3]GANGA INSTITUTE OF TECHNOLOGY AND MANAGEMENT (MDU); JHAJJAR

[1] Shuklaprakash23@gmail.com; [2] niranjan.hig41@gmail.com; [3] yashpalsingh009@gmail.com

Abstract:- ***All companies systematically monitor the computers, internet, or emails use of its user's employees. There are over hundred's different products available today that will let organizations see what their users do at work on their "personal" computers, in their email, and on the internet. The Espionage & Monitoring System program logs all Keystroke Logging along with the name of the application in which the keystrokes were entered. Using Espionage & Monitoring System, we prevent the miscellaneous use of system. Using this we capture all information in text and image form. Bot softwares are implanted on a machine to intentionally monitor the user activity by logging keystrokes and eventually delivering them to a third party . Espionage & Monitoring System are often maliciously exploited by attackers to steal confidential information. The Bots are covert security threat to the privacy and identity of users. The attackers are exploring different techniques of Bot using hardware loggers, software Bots and screen capturing, recording every things through software to steal the user sensitive data. To overcome this problem, we have proposed a model. In this model solution to Espionage & Monitoring System and Screen Recording Software has been proposed by using the concept of fabricated password on untrusted machine. It deceives the untrusted system's key logging and video capturing software. The main feature of this model is that it has a hardware recognition to retrieve the key .This key is required by the Temporary Filter layer (TFL) as an intermediary to change into the trusted password after bypassing all the capturing techniques and returning the original password to the required website.***

Keywords:- ***Onscreen Keyboard, Screen Recording, Anti-key , Thumb drive, Email monitoring, Internet monitoring, Computer monitoring, Chats/IM is monitoring, Network monitoring, Document monitoring, Web site monitoring, Productivity monitoring.***

INTRODUCTION

The threats of Bots are increasing day by day and the threat is becoming potent as people are unable to detect the presence of Bots in the system. Moreover the threat has taken a severe form after the introduction of screen recording software which paralyzes the anti Bot mechanisms like the virtual keyboards which are pertinent today. It is a major threat because most of the access control such as login ids which are entered through keyboard gets stored or recorded. Hence makes the control mechanisms ineffective. Espionage & Monitoring System are mainly classified into two categories: Hardware Espionage & Monitoring System and Software Espionage & Monitoring System.

A. Hardware Espionage & Monitoring System

Hardware Espionage & Monitoring System is mainly a small electronic device used for capturing the data in between a keyboard device and I/O port. When they are mounted in a computer system they start capturing the keystrokes in their inbuilt memory.

These Espionage & Monitoring System can be plugged inside the keyboard port, or directly inside the keyboard or at the end of the keyboard cable. The main privilege of hardware Espionage & Monitoring System is that it does not use any computer resource so it becomes quite infeasible for the anti-viral software or scanners to detect.

The keystrokes logs are stored in encrypted form in its own memory instead of the computer's hard disk .The major disadvantage of hardware Espionage & Monitoring System is that they necessitate physical installation in the keyboard or Computer case.

B. Software Espionage & Monitoring System

Software Espionage & Monitoring System logs and monitors the keystrokes and data within the target operating system, store on hard disk or in remote locations, and send them to the attacker. Software Espionage & Monitoring System is mainly based on the operating system. The Major Problem of Data Theft due to use of the Bots were minimized by the use of various anti - Espionage & Monitoring System mechanisms. Virtual keyboard only operates through mouse clicks so the key strokes are not captured. The virtual keyboard uses the concept of random shuffling of keys; hence it is not having a definite structure. Therefore the key presses if captured cannot be used because of the random changing of the key locations.

C. Screen Recording Software's:-

Screen Recording Softwares are prevalent because they are used to capture whatever on the screen for monitoring for the purpose of the Educational Demonstrations. This could be used negatively because the software could be used to capture the screen and mouse movement, so it is using the virtual keyboard to avoid Bots are no safer. This software records the screen activities which includes key presses trough virtual keyboards. Whatever activities are done on the screen, it is recorded and hence the passwords could be easily being captured. The model we have proposed deceives the Bot and

screen recording software and there by passing the malicious techniques and help the access control techniques to work properly. The model is to such a problem by passing of hardware and software Espionage & Monitoring System as well as the screen recording softwares.

LITERATURE SURVEY

Different works deal with the detection of key loggers. The simplest approach is to rely on signatures, i.e. Fingerprint of a compiled executable. Many commercial anti-malware adopt this strategy; show that code obfuscation is a sound strategy to elude detection. In the case of user-space Espionage & Monitoring System, we do not even need to obfuscate the code. The complexity of these Espionage & Monitoring System is so low to the source code are trivial. While ours is the technique to solely rely on unprivileged mechanisms, several approaches have been recently proposed to detect privacy-breaching malware, including Bots. One popular technique that deals with malware in general is taint analysis. It basically tries to track how the data is accessed by different processes by tracking the propagation of the tainted data. Moreover, furthermore, all these approaches require a privileged execution environment and thus are not applicable to our setting.

1) What Espionage & Monitoring System Are?

Espionage & Monitoring System the user's input is a privacy-breaching activity that can be per pet rated at many different levels. When physical access to the machine is available, an attacker might wiretap the hardware of the keyboard.

```
# using System;
# using System.Collections.Generic;
# using System.Linq;
# using System.Windows.Forms;

# Namespace Adulterate_Bot_Detection
```

2. Defenses against Bots:-

In the past years many defenses were proposed. Unfortunately, positive results were often achieved only when focusing on the general problem of detecting malicious behaviors. Detection of Bots behavior has notably been an elusive feat, the applications that legitimately intercept keystrokes in order to provide the user with additional usability - related functionalities (for example, a shortcut manager)

3. METHODOLOGY

3.1 Introduction

Our effort is explicitly focused on designing a detection technique for Type I and Type II user-space Espionage & Monitoring System. Unlike Type III Espionage & Monitoring System, they are both background processes which register operating-system-supported hooks to surreptitiously log every keystroke issued by the user into the current foreground application. Our aim is to prevent user-space Espionage & Monitoring System from stealing confidential data originally intended for a (trusted) legitimate foreground application. The key advantage of our approach is that it is centered on a black-box model that completely ignores the Bots internals. I/O monitoring is a non-intrusive procedure and performed on multiple processed simultaneously. Our technique can deal with a large number of Bots transparently and enables a fully-unprivileged detection system able to all the processes running on a particular system in a single run. In the following, we discuss how our approach deals with these challenges.

3.2 Injector

The injector is to inject the input stream into the system, mimicking the behavior of a simulated user at the keyboard. The injector must satisfy several requirements.Tt should only rely on unprivileged API calls. Second, it should be capable of injecting keystrokes at variable rates to match the distribution of the input stream. Finally the resulting series of keystroke events produced should be no different than those generated by a user at the keyboard. In all Unix supporting the same functionality is available via the API call X Test Fake Key Event, part of the XTEST extension library.

3.3 Monitor

The monitor acts for recording the output stream of all the running processes. As done for the injector, we allow only unprivileged API calls. The strategies to perform real time monitoring with minimal overhead and the best level of resolution possible. Finally, we are interested in application-level statistics of I/O activities, to avoid dealing with system level caching.

3.4 **Translator**:-

The pattern translator is to transform an AKP into a stream and vice-versa, given a set of target configuration parameters. A pattern in the AKP form can be modeled as a sequence of samples originated from a stream sampled with a uniform time interval.

3.5 Detector**:-**

The detection algorithm lies in the ability to infer a cause effect relationship between the keystroke stream injected in the system and the I/O behavior of an Espionage & Monitoring System process between the respective patterns in AKP form. While one must examine every candidate process in the system, the detection algorithm operates on a single process at a time, identifying whether there is a strong similarity between the input pattern and the output pattern obtained from the analysis of the I/O behavior of the target process.

3.6 **Generator**:-

It is designed to support several possible pattern generation algorithms, The pattern generator can leverage any algorithm producing a valid input pattern in AKP form. We present a number of pattern generation algorithms and discuss their properties. An important issue to consider is the effect of variability in the input pattern.

4. EVALUATION:-

T o demonstrate the approach and evaluate the proposed detection technique, we implemented a prototype based on the ideas described in this chapter . Our prototype is entirely written in C# and runs as an unprivileged application for the Windows OS. It also collects simultaneously all the processes' I/O patterns, thus allowing us to analyze the whole system in a single run.

5. Model System

The main purpose of the model is to bypass the Espionage & Monitoring System and Screen recording software by Two Factor Authentication and hence securely use password in un-trusted machines

A. Terms Used in the Model Following terms used in our model:

Trusted Systems:-

Trusted systems are systems with proper security configuration, updated patches, updated antivirus, configured firewall etc. These systems are those which are accessed by only one user or the persons trusted by him like his family members, for example personal Computer, laptop etc

Un-trusted System:-

Un-trusted systems are systems which are accessed not by the user but by many other persons like computer system in cyber cafes, computer in public places. These systems are termed as untrusted systems which may have Bots, screen recording software or other malicious programs. The Un-trusted system has been divided into the following two zones:

a. **Trusted Zone**:-This is considered free from presence of any malicious software like Adulterate -Bots. The trusted zone is a safe place where the windows procedure operates and where the web-browser is present.

b**. Un- trusted Zone**:-The malicious activities such as existence of Adulterate -Bots and screen recording software are operational here in this zone. This is considered unsafe for the critical data as it might get captured. This is the zone where protection of the critical data is required.

6. Onscreen Keyboard:

IT is a software component that allows a user to enter characters. The onscreen keyboard is generally a visual representation of the real keyboard on the standard output. An onscreen keyboard can usually be operated with multiple input devices, which may include an actual keyboard, a computer mouse, an eye mouse, and a head mouse. [Secure Authentication using Dynamic Virtual Keyboard Layout].

7. **Filter Layer**:

This is a layer which is only operational when needed by user in the un-trusted machine. It is present in between the un-trusted and the trusted zone. This is the main operational layer which converts the fabricated password to the original password before reaching the window procedure. The existence of this layer will only be during the critical data transfer such as the password transfer.

8. **Hooks:**

An application can register (hook) itself into a point so that any message flowing in windows message mechanisms is passed to the hooked application before going to the original target that receives the message.[4]The two types of hooks :-

A. Global Hook: Global hooks monitor system-wide message.

b. Local Hook: Local hooks monitor application specific messages.

Window procedure:-It is the active window for which the key press is intended to. [4] Here for our purpose the active window will be the web-browser with the webpage where the user requires the access codes and credentials to be reached safely.

WYSINT:-What you see is not true is the concept used in this model where the recording and capturing agents are deceived by making them capture something what is not useful.

Original Password: The password which is used in trusted systems.

Fabricated Password: The password used in un-trusted systems.

Pen drive/Thumb drive/USB Mass Storage Device.

Unique id: This is the key used for identification of the Hardware Device.

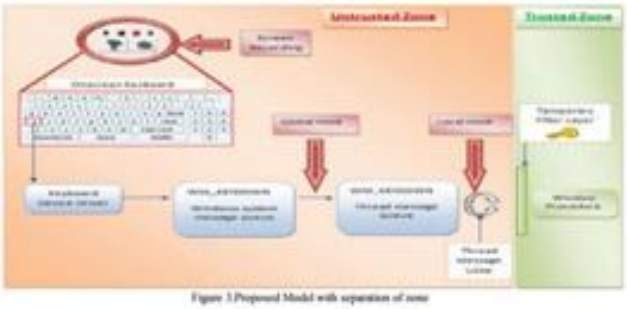

Figure 3 Proposed Model with separation of zone

Figure 4. Creation and Operation of Temporary Filter Layer (TFL) in Un-trusted Machine

Algorithm

a. Key generation In this phase a unique key is generated for each user based on its original password , fabricated password and pen drive's unique ID , this key will be used to login on un-trusted machine .The key generated will be random in nature. The key generated by this algorithm is a combination to two strings CKEY and PKEY.CKEY is stored in an array, which stores difference in ASCII value of OPASS and FPASS. PKEY is stored in an array, which is obtained by applying operation P on ASCII value PID of pen drive .Required key TKEY is stored in an array which is the sum of corresponding elements of CKEY and PKEY.

i. Creation of CKE:-

1. OPASS – Stores the ASCII value of the original Password intermediary CKEY which will lead to format ion of final key i.e. TKEY

2. 2. FPASS – Stores the ASCII value of the fabricated Password
3. 3. CKEY – Difference of corresponding elements of OPASS and FPASS

Example

i. Key generation

Unique ID of the Plugged in Pen drive is converted into the corresponding ASCII value i.e. PID which is further divided into two sub array. These two sub arrays are added to form PKEY. Now, the difference between ASCII value of original password i.e. pra_123 and fabricated password i.e. prakash results in the CKEY. The corresponding element of CKEY is added to the PKEY to get the final key TKEY.

ii. Key retrieval

As Unique ID of the Plugged in Pen drive is converted into the corresponding ASCII value i.e.PID which is further divided into two sub array. These two sub arrays are added to form PKEY. This PKEY is subtracted from the TKEY, which is retrieved by the application from pen drive, it will result in CKEY. This CKEY is added to FPASS to get the OPASS. The FPASS is the ASCII value corresponding to fabricated passwordi.e. PRAKAS. The OPASS is the ASCII value corresponding to original password. This ASCII value is converted back to character to get original password i.e. pra_123

Figure 7. Key Generation Algorithm Example

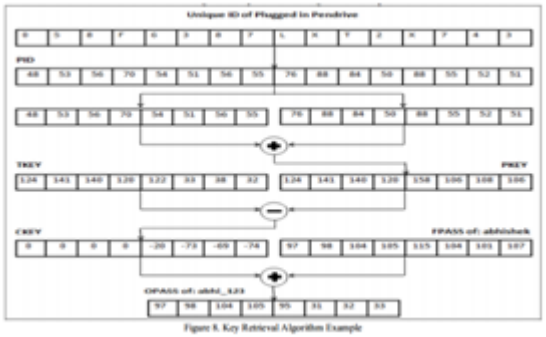

Figure 8. Key Retrieval Algorithm Example

CONCLUSIONS

The issue of Espionage & Monitoring System and screen recording software are addressed and a new USB mass storage device authenticated anti key-Espionage & Monitoring System technique is proposed. This new technique not only provides the protection against screen recording software, Bots. The model developed in an increasing the client side security in a client –server architecture and help in battering Espionage & Monitoring System and screen recording software in any of its forms. The model follows two factor authentications these factors are:

Something the user knows (fabricated password); and something the user has (USB mass storage device with the key).This model could be implemented in the windows operating system platform and further it could be used in the other platforms as well. The model could replace the virtual keyboard technology used in banking portals by integrating the temporary filter layer and applying the two factor authentication so that the clients could securely enter the critical information. Further research could be carried out to design a single password solution. In which user has to remember a single simple password for multiple websites to keep the critical information secure. This research presented Key Catcher, an unprivileged black-box approach for accurate detection of the most common key loggers, i.e., user-space key loggers. In addition, we augmented our model with the ability to antiracially inject carefully crafted keystroke patterns, and discussed the problem of choosing the best input pattern to improve our detection rate. We successfully evaluated our prototype system against the most common free key loggers [10], with no false positives and no false negatives reported.

REFERENCES:-

[1] G. Canbek, "Analysis, design and implementation of Keyloggers and anti-Keyloggers" Gazi University, Institute Of Science And Technology, M.Sc. thesis (in Turkish), Sept. 2005, pp. 103

[2] Williams, "I know what you did last logon: Monitoring software, spyware, and privacy,"Microsoft Security News. vol. 4, no. 6, June 2007.

[3] M. Kotadia, "Keyloggers spying at work on the rise, survey says," CNET News.com, May2006;http://news.com.com/Keylogger+spying+at+work+on+the+rise,+survey+says/2100-7355_3-6072948.html accessed on 27 Jan 2012.

[4] S. Seref and C. Gurol "Keyloggers increasing threats to computer security and privacy" IEEE Technology and society magazine, 2009, pp.10-17.

[5] www.keylogger.org accessed on 9 Dec 2011

[6] F.S. Lane, "The naked employee: How technology is Compromising workplace privacy" AMACOM Div American Mgmt. Assn., 2003, pp.128-130.

[7] S. Gong "Design and Implementation of Anti-Screenshot Virtual Keyboard Applied in Online Banking "E-Business and E-Government (ICEE), 2010 International Conf., 7-9 May 2010, pp-1320-1322

[8] L. Valeri "Screen Recording System for Windows Desktop" Russian-Korean International Symposium Science and Technology conf., 2004, pp.107-109

[9] M Agarwal , M Mehra "Secure Authentication using Dynamic Virtual Keyboard Layout" ICWET – TCET, Mumbai, India, 2011

[10] L. Keun-Gi, "USB PassOn: Secure USB Thumb Drive Forensic Toolkit", Future Generation Communication and Networking, 2008. FGCN '08. Second International Conf

[11] S. Sagiroglu and G. Canbek, "Keyloggers," IEEE Technology and Society Magazine, vol. 28, no. 3, pp.10–17, fall 2009.

[12]ThinkGeek.com,"Spykeylogger,"2010(accessedMay8,2010),http://www.thinkgeek.com/gadgets/sec

[13] G. Hoglund and J. Butler, Rootkits: Subverting the Windows Kernel. Addison-WesleyProfessional,2005.

[14]C.WoodandR.K.Raj,"Samplekeyloggingprogrammingprojects,"2010(accessedMay8,2010)http://www.cs.rit.edu/~rkr/ keylogger2010.

[15] Bauer, Michael D., Chapter 10 (System Log Management and Monitoring) of Building Secure Servers with LINUX, O'Reilly, 2002.

[16] Babbin, Jacob et al, Security Log Management: Identifying Patterns in the Chaos, Syngress, 2006

[17] Stout, Kent,"Central Logging with a Twist of COTS in a Solaris Environment.", SANS Institute, March 2002, URL: http://www.sans.org/rr/papers/52/540.pdf

[18] Stout, Kent,"Central Logging with a Twist of COTS in a Solaris Environment.",SANS Institute,March2002,URL:http://www.sans.org/rr/papers/52/540.pdf

[19]Mendez,William,"WindowsNT/2000EventLogs.",SANSInstitute,April2002,URL:http://www.sans.org/rr/papers/67/290.pdf

[20] T.Olzak, "Keystroke logging (keylogging)," Adventures in Security, April 2008 (accessed May 8, 2010),http://adventuresinsecurity.com/ images/Keystroke_Logging.pdf.

[21] S.Shah,"Browser exploits-attacks and defense,"London, 2008(accessed May 8, 2010), http://eusecwest.com/esw08/esw08-shah.pdf.

[22] P. Mell, K. Kent, and J. Nusbaum, "Guide to malwareincident prevention and handling," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. 800-83, November 2005.

[23] B. Whitty, "The ethics of key loggers," Article on Technibble.com, June 2007 (accessed May 8, 2010), http://www.technibble.com/the-ethics-of-key-loggers/.

