Identification and Prevention of Intrusion Attacks on Computer Systems Using Keystroke Computing Approach

[1]Dawodu, A. A., [2]Otukelu, O.N., [3]Odusanya, O.A. & [4]Onanuga, A.G.
[1&3]Department of Computer Science and Statistics, D.S Adegbenro ICT Polytechnic, Itori, Ogun State, Nigeria
[2]Department of Computer Engineering, D.S Adegbenro ICT Polytechnic, Itori, Ogun State, Nigeria
[3]Department of Computer Science, Ogun State College of Health Technology, Ilese, Ogun State, Nigeria
E-mail: alandawodu@gmail.com; tuxyt2000@yahoo.com; oloriebiodus@gmail.com
Phone: +2348055141696

ABSTRACT

In a modern computer, the interpretation of a pressed key is generally left to the software. Keylogging is one of the most popular spying software in the computer history. A computer keyboard distinguishes each physical key from every other and reports all keystrokes to the controlling software. Physical keyboard is used to type text and numbers into a Word Processor, Text Editor or other programs. In a modern computer, the interpretation of keystrokes is generally left to the software. A command-line interface is a type of user interface operated entirely through a keyboard. This research examined the identification and prevention of intrusion attacks on computer systems using keystroke computing approach

Keywords: Identification, Prevention, Intrusion Attacks, Computer Systems and Keystroke Computing Approach

iSTEAMS Proceedings Reference Format
Dawodu, A. A., Otukelu, O.N., Odusanya, O.A. & Onanuga, A.G. (2019): Identification and Prevention of Intrusion Attacks on Computer Systems Using Keystroke Computing Approach Proceedings of the 15th iSTEAMS Research Nexus Conference, Chrisland University, Abeokuta, Nigeria, 16th – 18th April, 2019. Pp 141-146. www.isteams.net - DOI Affix - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V15N1P15

1. INTRODUCTION

Virtual technologies are acting differently when interpreting a keystroke from user keyboard, and that depends on how the virtual machine sees it's hypervisor and how the hypervisor handling and using the hardware resources, such as keyboard. The keystrokes entered on the keyboard will be necessary to detect, since one of the purpose is to log the keystrokes performed by the attacker. In computer environment, it exists both in hardware keyloggers and software keyloggers. The hardware keyloggers can only log from the only one physical machine, the hardware keylogger is installed on. The software - keylogger can log local and remote users. It will be necessary to use a software keyloggers for log intruders from all over the world.

2. RELATED WORKS

Although research regarding keyloggers and their probable impact are likely to people, started with astonished insecure networking applications such as Telnet, file transfer protocol, which simply passed all confidential information (including user's passwords) in the clear form over the network. This situation was attacked through broadcast-based networks that were commonly (e.g. Ethernet) which allowed a malicious user to eavesdrop on the network and collect all communicated information.

In February 2005, Joe Lopez, a businessman from Florida filed a flint against Bank of America after unknown hackers stole $90,000 from his Bank of America account. An investigation showed that Mr. Lopez's computer was infected with a malicious program. Backdoor coreflood, which records every keystroke and sends this information to malicious users via the internet. This is how the hackers got hold of Joe Lopez's username and password. In February 2014, an article at www.nrk.no states that the Norwegian Police Security Service (PST) asked politicians for permission to install ways to monitor data keyboards of people they have in the spotlight. This could be achieved by installing a proper keylogger secretly on the remote machine to log keystrokes. Keystrokes logging has become an established method used by hackers for fetching passwords and other confidential data. Not only for hackers, but also for others such as; system administrators for systems, detecting suspicious users. Also, in research for different cases such as for parents for monitoring children for detecting special behaviours and criminals. Keystroke logging can also be a very useful method to detect attacks and their attack mechanisms, when setting up keylogger. However, some keyloggers today work on clean platform formed on bare metal machines and could maybe not work in platforms build on a virtual platform environment, since the hardware keyboard could be interpreting differently with a bare mental system. This interpreting issue of keyboard stroke signals may cause problem when trying to keylog the attackers in a money pot in a virtual environment.

The situation to develop a kernel keylogger that works on virtual machines in any environment is a big motivation for this.

A keylogger known as keystroke logging or Keylogging is a hardware or a software program that records a lot of user inputs and user activity. The real time activity of a computer user including the keyboard strokes that is pressed, websites visited, programs running, instant messages as well as other computer related activities. The user might know it, or the keylogger is hidden from the user for malicious purposes. If a keylogger is installed on a system, it can be configured to start every time the computer turns on. There exist two types of keloggers: software keylogger and hardware keyloggers. The Hardware keylogger is a device that is connected between the keyboard and the I/O on the computer's hardware for logging keystrokes entered in the computer. Some of the hardware keyloggers work at BIOS level while some are based on keyboard level. The hardware keylogger does not require any driver or software and will work with all Linux based operating systems as well as with Windows operating systems. Hardware keyloggers are used for keystroke logging by means of a hardware circuit that is attached somewhere in between the computer keyboard and the computer, typically in line with the keyboards cable connector.

However, hardware based keylogger do not depend upon any software being installed as they exist at a hardware level in a computer system. There are also USB based hardware keyloggers as well as ones for laptop computers. A hardware keylogger has an advantage over a software keylogger solution; it is not dependent on being installed on the target computers operating system and therefore will not interfere with any programs running on the target machine or be detected by any software. More so, a software keylogger is installed on a computer, directly or by remote installation. The software keylogger is invisible to the human eye, while hardware keylogger is easy to spot if a user checks is connected to the computer. Software based keyloggers use the target computers operating system in various ways, including; imitating a virtue machine, hypervisor based, or virtual machine manager, acting as the keyboard driver (kernel based), to watch keyboard store.

Within software key logger, there are also two different types:

i. User level logger
ii. Kernel level key logger

A Kernel level- based key logger is a program on the machine that gets administrator permissions and hides itself in the OS and start interrupting keystrokes because keystrokes always go through the kernel. A keylogger using this method can act as a keyboard device driver and thus gain access to any information typed on the keyboard as it goes to the Operating System. A User level keylogger are the easiest to create, but also the easiest to detect. This is the most common method used when creating keyloggers. The keylogges sets on global hook for all keyboard events and for all the threads in the system. Normal keylogging application store their data on the local hard drive, but some can be configured automatically and transmit data over the network to a remote computer file server or web server.

3. INSTANT BASED LEARNING MODEL USING TIMING ATTACK SYSTEM ARCHITECTURE.

1. Timing attack system model:

There are some assumptions that were used in determine the IBL model timing attack system architecture. The users may and may not be familiar with the way keyboard typing were done. The statistics showed that an average length of 1000 characters that was based on random sampling and is chosen by the text person with no restriction to what he/she typed. In fact, does not matter whether the text typed is not correct or not, it can be repeated as many as possible which is against the rule and the following can happens:

- The network latency is not constant, it can either below or high.
- The network latency may be bigger than the interkeystroke timing due to problem with transmission medium itself or propagation time In the secure shell.
- The adversary can eavesdrop the users encrypted communication.

However, and attacker can take advantage by observing the data stream (i.e attackers can measure the arrival time of the password keystrokes packets, and get the interkeystrokes timing of the super user password .

2. Instance based learning model timing attack system :

The timing attack system is a system that uses instance based learning model to measure inter arrival time of a keystroke, analyses of keystrokes and infers the keystrokes press in order to determine the password transmitted in a secure shell session.

The component of attack system schematics are ;

a. Information analysis module, is also known as “the parser” and then processes the sniffer’s output file, searches for a SU command and thus extracts the password packet arrival-time intervals (i.e it detects if a password or normal input is typed). In fact, if the user types a password, the corresponding keystroke timings are forwarded to the IBL algorithms.

b. Network sniffer module, which is a sub-unit monitors network data. Infact, sniffers normally act as network probes or snoops and thus examine network traffic, allow a copy of the data without redirecting / altering it (i.e wire shark).

c. IBL algorithm (IBLA): The IBLA module processes the timing and produces a least of the most likely passwords according to the statistics. Infact, the modules are independent and not automated. i.e. time will activated normally in correct order for the full attack. Also, the modules calculate possible password candidates using the highest or latency that match the query latency.

d. Keystrokes timing characteristics modules (KTCM): it specifies the characteristics of a special users general timing characteristics which helps to determine users pattern on the network.

e. Round Trip Times (RRT) Module**:** is a module that measures the length of the time it takes for a signal to be sent, in a addition, the length of time it takes for an acknowledgement of that signal to be received to both host.

Instant Base Learning (IBL) Module: In a machine learning, IBL or memory based learning is a family of learning algorithm, that instead of performing explicit generalization, in which we compare new problem instances with it instances seen in training, that have been stored in memory. It is called instance based because it constructs hypothesis directly from the training instances themselves. I.e. hypothesis complexity can grow with the data.
The major advantage of IBL has over other method of machine learning is its ability to adapt it model to previously unseen data.

Keystroke Timing as TBL
In a system like this, each character pair is a non-observable state. The latency between two keystrokes is the observed output. Each of the state corresponds to a pair of character, so that typing the sequence K_0, K_1, …….Kr is a process that goes through T states, q_1, q_2, ……q_r we denote yt the observed latency of state qt.
However, to model the process as IBL there are two characteristics to be met;

1. The characters are uniformly distributed to that the probability of transition from the current state to another state depends only on the current state. This assumption usually holds "good" password.
2. The probability distribution of the latency is dependent only on the current state but in some cases it does not hold due to typing latency of the character which changed based on individual pattern.

4. IMPLEMENTATION

Web Application Server
A web server is a program that, using the client/server model and the WWWebs hypertext protocol (HTTP), serves the file that forms the web pages to web users (where computers contain HTTP clients that forward their request). Web servers often comes as part of a larger package of internet and internet related programs for the File Transfer Protocol (FTP) files, and building and also publishing web pages. Apache Tomcat, formerly called Jakarta Tomcat, is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat will implement the java servlet and the java page (JSP) specifications the sun micro system (new oracle) and provides a "pure Java" HTTP web server environment for Java code to run. Apache Tomcat tools for configuration and management, but cab also be configured by editing XML configuration files.

Web Server Setup
The description uses the variable In the form;
Name & CATALINA_BASE to refer the base directory against which most relative part are resolved. If Tomcat has not been configured for multiple instance by setting a CATILINA_BASE directory then & CATALINA_BASE will be setup to the value of CARALINA_HOME, the directory into which Tomcat has been installed. However, to install and configure SSL support on Tomcat, the following steps were carried out as follow;

1. Create a key store file to store the server private key and self-signed certificate by executing the following command:

Windows:

```
%JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA
```

Figure 4a: Command for creating self-signed certificate on windows.

Unix:

```
$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA
```

Figure 4b: Command for creating self-signed certificate on unix. and specify a password value of "programit" or any password of choice.

Uncomment the "SSL HTTP/1.1 Connector" entry in $CATALINA_BASE/Conf/server.xml and modify as described below Tomcat currently operates only on JKS, PKCS11 or PKCS12 format keystores. The JKSformat is Java's standard "Java Keystore" format, and is the format created by the keytool command-line utility. This tool is included in the JDK. The PKCS12 format is an internet standard, and can be manipulated via (among other things) OpenSSL and Microsoft's Key-Manager. Each entry in a key-store is identified by an alias string. Whilst many key-store implementations treat aliases in a case insensitive manner, case sensitive implementations are available. The PKCS11 specification, for example, requires that aliases are case sensitive. To avoid issues related to the case sensitivity of aliases, it is not recommended to use aliases that differ only in case. To import an existing certificate into a JKS keystroke, please red the documentation (in your JDK documentation package) about key-tool. Note that OpenSSL often adds readable comments before the key, key tool does not support that, so remove the OpenSSL comments if they exist before importing the key using the tool. To import an existing certificate signed by your own CA into PKCS12 keystore using OpenSSL, the following comman would be executed:

```
Openssl pkcs12 -export -in mycert.crt -inkey mykey.key \ -out mycert.p12 -name tomcat –CAfile
myCA.crt \ -canname root -chain
```

To create a new key-store from scratch, containing a single self-signed Certificate, execute the following from a terminal command line:

Windows:

```
$JAVA_HOME%\bin\keytool -genkey -alias
```

Figure 6a: Creating a keystore from scratch on windows.

Unix:

```
$JAVA_HOME/bin/keytool -genkey -alias
```

Figure 6b: Creating a key-store from scratch on Unix.

Web Application Home Page.

Figure 7 shows the web application home page. This page contains a selection of text that contains some basic information about the application, but most importantly it offers links to the "key stroke analysis" and "Performance" pages. These pages are used to accomplish the key stroke attack analysis and also implement its performance. The web application home page and subsequent pages used secured http (https) protocol which provides a secured communication between the client and the server.

5. DISCUSSION

Several theories, approaches, techniques, models and methodologies were studied before Instance Based Learning model was used for this research. The research work discussed the keystroke timing analysis attacks on SSH using Instance Based Learning Model. The researchers also analysed if it were possible to perform the timing analysis attack as it was described in the work. The research work discovered that, there are countermeasures against timing analysis implemented in the latest version.

The implementation of an Instance Based Learning Model for Timing Analysis of Keystrokes and Timing Attacks on Secure Shell provide a fix for the attack that makes this countermeasure non-effective by providing a Round-Trip-Times (RTT) Model subsystem in the attack system., but with minimal time taken in breaking the passwords and provide better performance in term of efficiency using various parameters such as Sequence Time Threshold (STT) which is the number of time of a latency sequence between characters pair that can be used to identify a user on the system. An attacker can measure the Round-Trip-Times (RTT) o both hosts, he eavesdrops. If these times are exact enough, he can reveal the even presented timing information by subtracting the RTT's from the eavesdropped timings. So the attacker can distinguish, whether the echo-mode is activated or not.

6. CONCLUSION

In this paper, it has been discovered that Secure Shell (SSH) is not secure as people popularly believed because there are serious security threats emanated from the protocol when carrying out timing analysis of keystrokes and timing attacks. Implementation of an Instance Based Learning (IBL) Model for Timing Analysis of Keystrokes and Timing Attacks on Secure Shell learns when a password is typed, how long the password is and reveals the password irrespective of character combination that form the password.

REFERENCES

1. Song, D. and Tian, X. (2001). Timing Analysis of Keystrokes and Timing Attacks on SSH, 10th USENIX Security Symposium.
2. Noack, A. (2007). Timing Analysis of Keystrokes and Timing Attacks on SSH Revisited, Horst G"ortz Institut f'ur IT- Sicherheit Ruhr-Universit"at Bochum.
3. Lustig, M. and Shabtai, Y. (2001). Keystroke Attack on SSH, Final Project Report at Technion IIT.
4. Asonov, D., and Agrawal, R. (2004) Keyboard Acoustic Emanations. In Proceedings of the IEEE Symposium on Security and Privacy.
5. Berger, Y., Wool A and Yeredor A. (2006). Dictionary Attacks Using Keyboard Acoustic Emanations: In Proceedings of ACM Conference on Computer and Communication Security (CCS).
6. Shah, G., Molina A. and Blaze, M. (2006). Keyboards and Covert Channels. 15th USENIX Security Symposium.
7. Gagliardi, F (2011). Instance-based classifiers applied to Medicak databass: Diagnosis and Knowledge Extraction, Artificial Intellingence in Medicine," 123-139.doi:10.1016/j.artmed.
8. Hoyge, M., Hughes C., Sarfaty, J., and Wolf, J.(2001). Analysis of the Feasibility of Keystroke Timing Attacks Over SH Connections. Research Project at University of Virginia.
9. Menezes, A., Oorschot P. and Vanstone, S.(1997). Handbook of Apllied Cryptography. CRC press (1997).
10. Shell Protocol", Asian Journal of Computer and Information Systems (ISSN: 2321 – 5658) Volume 01 – Issue 04.