A Framework of TPM, SVM and Boot Control for Securing Forensic Logs

Nazanin Borhan
Faculty of Computer Science and Information Technology, University Putra Malaysia

Ramlan Mahmod
Faculty of Computer Science and Information Technology, University Putra Malaysia

Ali Dehghantanha
Faculty of Computer Science and Information Technology, University Putra Malaysia

ABSTRACT

Computer logs files contain the crucial information that is stored and can be an important forensics evidence of attacks and actions of a system. Cyber forensics can be one of the important solutions to systematically gather, process, interpret and utilize digital evidence and log of the activities and events of a system is one of the most important resources of analyzing the evidence for researchers, therefore a secure storage of forensic log is our main focus. In this paper, we propose a Trusted Module Platform (TPM)-based solution along with using Secure Virtual Machines (SVM) to secure the storage of forensic logs of the system for cyber forensics investigation. Since TPM provides protection before system boot process, it heavily limits the number of attacks that may bypass. Also SVM provide a secure environment to test software before installing on the client-machine. To ensure a secure logging system, our model will be using a smart combination of TPM, SVM and secure boot control to provide maximum log protection.

Keywords

Digital Forensics; Secure Log; TPM; SVM.

1. INTRODUCTION

According to [1], the act of two factors is caused prevalent use of computer forensics: First factor is the growing dependence of law enforcement on computing and Second factor is the ubiquity computers that followed from the microcomputer revolution. In this context the computer forensics can be summarized as the process of identifying, collecting, preserving, analyzing and presenting the computer-related evidence in a manner that is legally acceptable by court. Regardless of using actual or Remote Digital Forensics Lab [2] to gather and analyze forensics evidence and logs, Forensics Log is the essential source of data needed to be well-preserved in digital forensic investigation. Electronic records such as computer and network logs are considered as the most important data in digital forensic. This is because electronic records are vulnerable to any change which makes it become less trustable when used in forensic investigation. Moreover, the security level of current software-based protection technique for electronic records is not guaranteed, while the level of protection using hardware based protection would be enhanced in comparison with software based techniques [3]. TPM and SVM are two types of hardware and software-based protections that will be introduced in bringing higher protection to secure logging systems.

The paper gives a summary of related works in Section II, describes forensics logs, introduces the main features of TPM and SVM and how they work in term of their technologies in hardware and software-based protection. Then, in Sects. III and IV, the paper describes more details about current secure logging techniques using TPM and SVM and then combining the implementation of both to achieve better solution of secure log in software and hardware-based protection. Finally, Sect. V concludes the better solution for secure logging system using TPM and SVM.

2. RELATED WORKS

In this section we describe the current trend in researches on protection of Log forensics. Two other fundamental parts of the paper will be described in this section, which are TPM and SVM.

2.1 Log Analysis

Fundamentally, most of the operating systems use log event as part of their security strategy. It means that system might keep a record of login attempts in a hidden directory that is not accessible by the user but accessible for the administrator. System administrator used the logged information to understand how a network intruder gained access to the system or to gain more information on the type of damage from the intruder that is inflicted after he/she gained access [4-5]. Different researches are focused on different approaches to analyze the log evidences and alerts of the system in detecting computer attacks. Four categories are shown in [6] to describe the log analysis and alert correlations are as follows:

- Similarity based approaches [7]; alerts are grouped based on the similarity between alert attributes and use alert clustering to identify “the same attack occurrence”, where expert rules are used to specify the similarity requirement between alerts.

- Predefined attack scenario-based approaches [8]; attacks are detected based on well-defined attack scenarios and this approach is not capable of discovering novel attack scenarios.

- Pre/post condition based approaches [9]; post-condition of one attack is matched to the pre-conditions of another attack. Although specifying pre-conditions and post-conditions for attacks are time-consuming and error-prone, this approach can discover novel attack scenarios.

- The multiple information sources based approaches; this approach is concerned distributed attack discovery. In this category DOMINO [10] is distributed intrusion detection architecture targeting at coordinated attack detection with potentially less false positives.

Modeling and analyzing the invariant properties of a system is not possible in these researches that are mainly based on correlation and intrusion detection. In order to overcome the problem of not existing a general methodology for forensic log analysis some ad hoc analysis techniques are introduces such as log analysis [11] and operating system-specific analysis [12].

To analyze log files an approach in [4-5] is based on computational logic and formal automatic verification. In order to convey the necessary information needed for the analysis the structure of an event was carefully chosen in these studies. In their models a multi-sorted term algebra could be represented an event whose operation symbols are chosen such that they devotedly carry the information kept in the real events that are kept in the log. As discussed in these papers, based on traces and events by using multi-sorted algebra, the system logs are modeled and represented as a tree that the edges of this tree could be extracted events in the form of algebraic terms. Before submitting to a model checker, the hypothesis that originated from a digital investigator is formally expressed in terms of properties and patterns of events. Although their work provides an exact and provable logical support, it does not allow reasoning on verifiable anti-forensic events starting from what is observable.

The basic formal notion used to define the log architecture and their semantics is defined in [13] by describing a framework for the requirement of log architectures and then proposing standards to describe satisfactory log architectures. The first benefit of their framework is to extract the necessities and duties in the logging process, which contributes to the transparency objective. The criteria for acceptable architectures provide guarantees of accurateness and completeness, which contributes to enhancing impartiality and authenticity. They mentioned that their standards can also be used to propose enhancements to derive satisfactory log architecture from non-acceptable log architecture. Conducting in the context of Liability Issues in Software Engineering (LISE), they formally proof the acceptability of the log architecture, its correctness and its consistency. Above from some definition missing such as damages in the logs, their model done well in involving lawyers and computer scientists with the aim to put forward a formal framework to define liability in IT systems in a precise approach and establish liability in failure scenarios.

Digital evidence has lacks of completeness in the area of checking log damaging that makes it problematic to conduct computer forensics processes. Because of the important concept of logs, we always must make sure it is not changed or damaged during producing before we rely to the digital evidence. In [14], researchers use the concept of steganography for logs forensics in order to maintain the completeness and reliability of evidence for future forensic processes and intrusion detection, by keeping and recording all the altered intrusions. Instead of directly sending the generated log files to the log server, they keep the log files in the working computer in their scheme. To solve the problem of individual computers that could have access to the logs after distribution of logs in them, they propose a scheme that applies steganography to achieve protection purpose by hiding text message into an image file. Therefore intruder does not find original files in the case of hacking the log files, so the file can safely remain the same computer with completeness in the next works with forensic investigations. Real time detection of the intruders and sending back the warning message and preservation of the original evidence are some of the advantages of this model that they mentioned in their paper.

Because of the time consuming and troublesome concept of collecting and analysing the network forensics evidences of the attackers, paper [15] is focused on collecting an evidence from Multi Source Log File from multiple network sensors and analyse them using Automated Analysis System of Intrusion Evidences (IEAAS) that is illustrated with Log Collection Agent (LCA) in network sensors and multiple modules in IEAAS. In this method, a platform for dynamic forensics is build up. By using evidence preservation method and performance-improved aggregation algorithm, their approach can improves the usual event correlation and preservation approaches for network forensics.

Instead of traditional approach of gathering information for network forensics, researchers in [16] proposed an automated network forensic approach based on fuzzy logic and expert system. The advantage of this method in comparison of traditional method is their analysis of computer crime in network environment in order to automatically making the digital evidence by using fuzzy logic and expert system for network forensics. They can proved that the results of their experiments in comparison to other common methods is improved, and their system can categorize most types of attack (91.5% accurate classification rate on average) and they can provide analysable and understandable evidence for forensic specialists. Their four stages forensic operations of their model could be mentioned as follows: (1) real-time forensic data acquisition and pre-procession; (2) knowledge base construction and dynamic rule generation; (3) fuzzy linguistic operation of input attack data; (4) compute aggregation fuzzy value and total fuzzy score. In the first stage, In order to real-time detect the network and hosts' intrusion information, they use different popular detection tools such as Snort, Tcpdump, Sniffer, BlackICE, etc., and pre-process and save these information into the database according to the forensic parameters of different data sources. The amount of forensic rules, factors and parameters could determine the memory usage of 3th and 4th stage.

Apart from the usual two classes of normal and abnormal behaviour of attacks, they demonstrated five various types of attacks behaviour in their result: DoS: Denial of service attacks; R2L: Remote-to-local- this attacks gain unauthorized access from a remote machine; U2R: User-to-root. These types of attacks use unauthorized access to gain local super-user (root) privileges. Probe: Probing. This kind of attacks is based on surveillance and probing for information. And finally they demonstrated the improvement in performance of detection rate of their proposed method that called automated Network Forensic approach based on Fuzzy Logic and Expert.

They also mentioned some deficiencies of their work; Apart from some problems in making and classifying the anomalous and novel attacks, because of the large computing cost of new rules, some anomalous and novel attacks cannot been detected successfully in time by using their method.

2.2 Trusted Platform Module

A trusted platform module (TPM) is a hardware chip with the specification from the Trusted Computing Group (TCG) [17], that consists of IT infrastructures and their goal is provide a mechanism for security and integrity of computing Platforms. It basically replaces software protection into hardware based protect by using encryption keys, digital certificates and passwords. Hardware protection is inherently less vulnerable to software-based attacks and authentication processes are conducted through a secure subsystem. The device also enhances the security of web browsers, email programs and other important applications [18-19].

TPM has a secure crypto-processor that generate and store secured keys and password for a very unique identity. TPM chipset is integrated into computer motherboard which means this combination is to enhance tamper resistivity [20]. The chip basically performs as a unique identifier and unique master key which are unable to create by another set of chip in order to secure from external software attacks and physical theft [21].

Integrating trusted computing with other security techniques is discussed in different researches that focused on hardware-based security solutions. For example, to mitigate botnet attacks to internet banking websites, in [22], researchers proposed a solution that integrates the existing technologies using a biometric USB thumb drive which needs to be built with a finger print sensor, RSA SecurID and TPM chip. The proposed solution is based on using biometric fingerprint sensor on a USB thumb for identify the actual owner of the flash that could prevents unauthorized access and theft of the device. The capability of TPM that is embedded inside the USB thumb is to store and protect the finger print of the user for whom the thumb is issued by the e-banking service. All the messages that are exchanged between the bank server and the endpoint device should be encrypted using the built-in TPM on the both ends that is responsible of encrypting, decrypting and verifying the header files of the messages.

2.3 Secure Virtual Machine

As stated by [23], virtualization can promote utilization of hardware by 5 to 20 times and can decrease the number of servers in organizations, therefore reduce the power consumption. Virtual machines are used to test software compatibility on different operating system. Secure Virtual Machine architecture is developed by AMD to provide enterprise-class server virtualization software technology that facilitates virtualization development and deployment [24]. Virtual machines that are running under SVM architecture are provided hardware resources that allow single machine to run multiple operating systems efficiently. Hence, AMD’s SVM provides hardware assists to improve performance and facilitate implementation of virtualization. Besides that, there are a set of hardware extensions that are provided by the SVM processor support. It will enable economical as well as efficiency in the implementation of virtual machine system. The hardware support falls into two categories which are virtualization support and security support. Virtual Machine Monitor (VMM) also consists in SVM and it is one of the important components in SVM that consist software that controls the execution of multiple guest operation system on a single physical machine. VMM works by intercepting and emulating in a safe manner sensitive operation in the guest which could not give a guest access to non-allowed memory.

3. CURRENT SECURE LOGGING THECHNIQUES

3.1 Hardware Based Trust of Logs using TPM

Current solutions to having a trust-relationship between log entries and logging machines are purely software-based or they only authenticate the users of the systems instead of systems themselves [25]. As hardware based attacks require physical access to the concerned TPM and security implications may arise, so they need more determination than software-based attacks and it is important that which exact make and model of TPM is used in the system.

To establish hardware-based trust in the log producing application without manipulation of existing logs, authors in [26] proposed a solution based on TPM and AMD’s Secure Virtual Machine technology (SVM) to establish a root of trust for client syslog daemons. Without considering the effects of software manipulation, their solution is based on enabling a hardware-based trust-relationship between log entries and logging machines which ensures that log data and log producer can be trusted.

3.2 Secure Logging in SVM

The logging system that is currently using by the operating system has two weaknesses that are integrity and completeness. First of all, an attacker can easily turn off the logging event as he/she gained access with the privilege as an administrator; thus the content of the log cannot be trusted completely. Secondly, it is hard to decide the type of information that is needed during the post attack analysis as the attacker may change the logging information and making the log lack of information on how the intruder gained access or the action of the attacker made after he/she gained access. In this case, SVM can solve the vulnerability of logging system that faced by most of the operating system.

Secure Virtual Machine monitor are smaller yet simpler than the guest operating system. Besides that, it is also less vulnerable to attack because of the completeness of logging in virtual machines. In this case, ReVirt which is represented in [27] is a SVM-based logging and replaying system plays an important role. ReVirt performs system logging at the Virtual Machine Monitor (VMM) layer and removing the need to trust the operating system. Besides that, it also records all the operations that are necessary to recreate an entire attack. Logging is done at an instruction-by-instruction level, this processes gives administrator a complete picture of the entire process that happened on the guest virtual machine even in the presence of non-deterministic attacks and execution.

Since ReVirt is a powerful logging tool in SVM, it may leak sensitive information if the logs are not protected in a correct manner. A secure system is building (GPCR04) to minimize the amount of time that sensitive data remain in the system. This system builds to cover or minimize the risk of the log being leak out. Sensitive data such as password and cryptographic keys are deleted right after it is used. If an attacker manage to get hold of ReVirt logs, he/she will gained access to all the sensitive information. However, it is very difficult to remotely control the ReVirt logging virtual machine and one method to minimize this threat is to encrypt the ReVirt’s logs.

4. SECURE LOGGING FRAMEWORK

During digital forensic investigation, log data should be preserved as it is a very important evidence in any forensic investigation. Existing solutions do not provide adequate protection, which could cause exposing the log-producing application to software-based attacks. In this section, solution will be given out as if the previous logging method is not secure enough or still vulnerable to attacks in some of the perspective. Few solutions will be pointed out as a new idea to enhance further the security of TPM in logging.

To enhance the protection towards digital log files, Trusted Platform Module (TPM), Dig-Force [28] and AMD’s Secure Virtual Machine technology (SVM) are used to protect the security of the clients’ systems. TPM is insusceptible to manipulation. As mentioned above, TPM is a special chip which is installed in motherboard to provide hardware authentication. As TPM is less vulnerable to software-based attacks, there is only one way to temper the log information, which is tempering the hardware. The attacker requires physical access to control the hardware such as changing the computer hardware and etc. Therefore, client side software attack can effectively be ruled out. TPM helps to minimize the risk of log information being tempered or compromised. But TPM hardware is only supported by selected software and firmware where the firmware and software are trusted each other.

In addition, SVM is a virtual machine that is used to test software compatibility and it is used for virtualization development and deployment for the purpose of security with using server virtualization software technology. An SVM allows multiple operating systems to be run on a single physical machine. It provides hardware resources to enabled virtual machine architecture to multiple operating systems run in a computer efficiently and yet security is guaranty.

Moreover, trustable syslog can be generated by using the latest Dig-Force2. With its API incorporates with TPM, boot jamming programs can be prevented from executing. Dig-force2 is preventing anyone who tries to bypass authentication of windows vista by using any boot jamming programs [28].

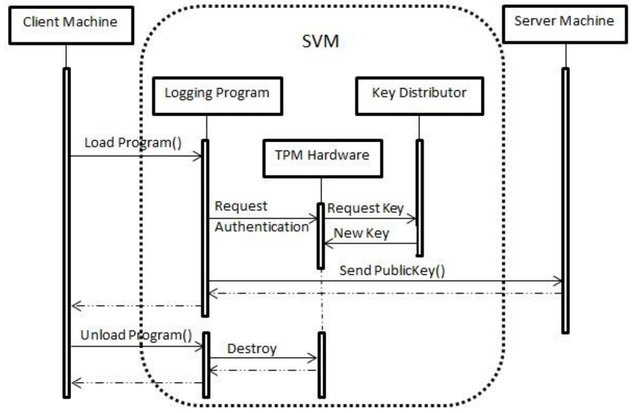

Fig 1: Proposed flow of the system for Client and Server authentication

Figure 1 Shows the flow of process of the proposed model for a simpl client and sever authentication model with Lgging program using SVM and TPM.

Even with the help of authentication status of system, user account and application and validity of the generated records, the Syslog still cannot be trusted as attacker may gain access to edit the log files. As a result, client syslog daemon can be prevented from tempering by getting root of trust from TPM and obeying the chain of trust to gain more secure syslog from client computers. TPM prevents untrusted software and firmware being installed and used in the client machine. Only trusted software and firmware are allowed to be used in client machines. Even if the attackers want to install their own modified software into the client machines, the software will be categorized under untrusted software and the software won't be able to get any access to the client data. With the help of boot control of Dig-Force security, attackers can be prevented hacking the client machine and also it is preventing malicious programs to be started at startup. Therefore, client machines are safe while the machines are booting. In combination with SVM architecture, any program which is going to install in client's machines will be checked in virtual machine. Besides testing the programs' compatibility, new programs will be tested in the virtual machine to check whether it is malicious programs or not, therefore only safe programs allow to install in the client machines.

5. CONCLUSION AND FUTURE WORKS

In this paper we highlighted different techniques of storing forensics logs and had reviewed related works done to overcome the problem of security of them. Various implementations need to be done on a computer system with the combination of hardware and software to achieve a secure logging system. TPM is just one of the tools that might be applied on a computer system to make secure logging possible. However, if TPM is implemented in combination with other techniques such as SVM and boot control, it would have a more secure result to achieve the target of secure logging. This is because TPM is only a hardware based component whereby it is a chipset that integrated on the computer system to prevent physical attacks. SVM and boot control are meant for software based defense against malicious attackers so that if attackers gain access as a guest in a computer system, they do not have privileged user access to the root system and launch various attacks. Therefore those who needed secure systems should take this solution in consideration to bring their security to the next level.

In future we will target the following enhancements:

- Implementing and evaluating model using formal methods – Verification of this model could be achieved by using formal verification techniques.

- Improving current cryptographic method – currently the TPM is using RSA type to encrypt the password stored on the hardware, in the future we hope to replace it with AES and Elliptic-Curve or both in combination to provide maximum protection.

6. REFERENCES

[1] Mohd Taufik Abdullah, Ramlan Mahmod , Abdul Azim Ab. Ghani, Mohd Zain Abdullah, and Abu Bakar Md Sultan. Advances in Computer Forensics. International Journal of Computer Science and Network Security (IJCSNS), VOL.8 No.2, pp. 214-219, 2008

[2] Ali Dehghantanha, Nur Izura Udzir, Ramlan Mahmoud, "Future Digital Forensics Labs," in the 2011 IEEE International Conference on Computer Applications and Network Security (ICCANS), pp. 27-29, Maldives (Accepted)- IEEE index, 2011

[3] J.D. Hietal, "Hardware versus Software", A SANS Whitepaper – September 2007 (Edited May, 2008)

[4] Ali Reza Arasteh, M.D., Assaad Sakha & Mohamed Saleh. Analyzing multiple logs for forensic evidence. Digital Investigation, DFRWS, Elsevier, pp. 82-91, 2007.

[5] Mohamed Saleh, A.R.A., Assaad Sakha & Mourad Debbabi, Forensic analysis of logs: Modeling and verification. Knowledge-Based Systems 20, Sience Direct, Elsevier, pp. 671–682, 2007.

[6] Dingbang Xu and Peng Ning, Alert correlation through triggering events and common resources. Tucson, AZ, USA, 2004

[7] S. Staniford, J.H.J.M., Practical automated detection of stealthy portscans. Journal of Computer Security 10(1/2), 2002.

[8] Debar, B.M.H., Correlation of intrusion symptoms: an application of chronicles. In Proceedings of the 6th International Conference on Recent Advances in Intrusion Detection (RAID), 2003.

[9] Miege, F.C.A., Alert correlation in a cooperative intrusion detection framework. In Proceedings of the IEEE Symposium on Security and Privacy, 2002.

[10] Jha., P.B.V.Y.S., Global intrusion detection in the domino overlay system. In: Pro. of the 11th Annual Network and Distributed System Security Symposium, 2004.

[11] Chuvakin, C.P.A., Security Warrior. O'Reilly, 2004.

[12] Heiser, W.K.J., Computer Forensics: Incident Response Essentials. Addison-Wesley, Boston, MA, 2002.

[13] Daniel Le M´etayer, E.M.M.-L.P., Designing log architectures for legal evidence. Software Engineering and Formal Methods, IEEE Computer Society, pp. 155-165, 2010.

[14] Wang, Y.-T.F.S.-J., Intrusion Investigations with Data-hiding for Computer Log-file Forensics. 978-1-4244-6949-9/10©2010 IEEE, 2010.

[15] Chen Lin, L.Z.G.C., Automated Analysis of Multi-source Logs for Network Forensics. First International Workshop on Education Technology and Computer Science, IEEE Computer Society, pp. 659-664, 2009.

[16] Niandong Liao, S.T.T.W., Network forensics based on fuzzy logic and expert system. Elsevier B.V., Computer Communications, Vol 32, pp. 1881–1892, 2009.

[17] Trusted Computing Group TCG Specification Architecture Overview, Specification Revision 1.4, 2007.

[18] Sadeghi, A.-R., Trusted Computing —Special Aspects and Challenges. In: Proceeding of SOFSEM, LNCS, 4910:98-117, Springer-Verlag Berlin Heidelberg, 2008.

[19] Martin Pirker and Ronald Toegl. Towards a Virtual Trusted Platform. Journal of Universal Computer Science, vol. 16, no. 4, 2010.

[20] Eric D. Bryant, Avni Harilal Rambhia, Mikhael J. Atallah, John R. Rice. Software trusted platform module and application security wrapper. United State Patents, Patent number: US 7,870,399 B2. Issuing date: 11 Jan 2011

[21] David Challener, K. Y., Ryan Catherman, David Safford, Leendert Van Doorn. A Practical Guide to Trusted Computing. IBM Published Book- Pearson Education, Inc Rights and Contracts Department ISBN-13: 978-0-13-239842-8, 2007.

[22] Muththolib Sidheeq, Ali Dehghantanha, Geetha Kananparan, "Utilizing Trusted Platform Module to Mitigate Botnet Attacks" in International Journal of Advancements in Computing Technology (IJACT), Volume 2 Issue 5, pp. 111-117, 2010- Korea- Scopus index,

[23] Yap Tze Tzuen, Ali Dehghantanha, Andy Seddon, and SeyedHossein Mohtasebi, "Greening Digital Forensics: Opportunities and Challenges," In Second International Conference on Recent Trends in Information Processing and Computing (IPC), Vol. 14-15, pp. 35-50, 2011.

[24] F.Felacy Silvia, "Security in Virtual Machine is better than Real Machine", International Journal of Computer Science & Communication, 2010.

[25] C. N. Chong, Z. Peng, and P. H. Hartel, "Secure audit logging with tamper-resistant hardware," Proceeding in 18th IFIP International Information Security Conference (IFIPSEC), vol. 250. Kluwer Academic Publishers, pp. 73–84, 2002.

[26] Benjamin Boeck and David Huemer, A Min Tjoa, Towards more Trustable Log Files for Digital Forensics by Means of "Trusted Computing". 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE Computer Security, pp. 1019-1027, 2010.

[27] X. Zhao, K. Borders, and A. Prakash. Svgrid: A secure virtual environment for untrusted grid applications. In CM/IFIP/USENIX 6th International Middleware Conference, France. 2005.

[28] K. Fujita, Y. Ashino, T. Uehara and R. Sasaki. Using boot control to preserve the integrity of evidence. Advances in Digital Forensics IV, Springer, pp. 61–74, 2008.