IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 4, Ser. III (Jul - Aug 2019), PP 30-35
www.iosrjournals.org

Malware Identification and Characterization through Classification for Runtime intrusion Detection and Clearance

Bhandare Trupti Vasantrao
BE IT, Dept. of Information Technology Engineering, Walchand College of Engineering, Sangli, India
ME Computer, Dept. of Computer Engineering, Smt. Kashibai Navale College of Engineering, Vadgaon Bk, Pune, India
Corresponding Author: Bhandare Trupti Vasantrao

Abstract: *Traditional malware programs are effectively identified and prevented by the current intrusion detection systems. But, the modern malware attacks use various techniques and programs to achieve different behavioural attacks into the systems, where a traditional intrusion system fails to prevent. Because, the behaviour of a program can largely depend on its variable data manipulation which can be an important malware attack technique. Various studies have suggested that classification of malware can be an effective technique for malware identification and detection. This paper proposes an efficient kernel malware identification and characterization through classification for runtime intrusion detection. The approach addresses the challenges in identifying the malware properties and features for identification of malware and the runtime detection of different attacks. The experimental evaluation shows an improvisation in classification accuracy and detection rate at different threshold limit, which proves the effectiveness of the proposal.*

Keywords: *Malware Identification, Classification, Characterization, Intrusion Detection*

Date of Submission: 03-08-2019 Date of Acceptance: 19-08-2019

I. Introduction

Traditional malware detection and analysis methods have been focused on the aspect of the code-centric malicious programs, such as the detection of malicious code injection or an estimate of malicious code sequences. However, in response to these and malware detection, modern malware using advanced techniques such as reusing existing code or obfuscating the code malware to avoid detection. Developers of malware prevention have been continuously exploring various attacks, which are tampering with the system.

Malware categorization and classification are important components in the malware detection mechanism [19]. It is used to distinguish its malware classification prescribed. The malware classification system, appliance or engine is referred to as classification. The malware classification function is commercially used in antimalware products using different machine learning techniques [4]. It was studied in the past that malware classification system is necessary and important to combat malware [1], [13], because, malware classification system and the identification of malware procedural right and effective solution for malware.

Malicious code or malware has long been identified as the world's great threats in computing [14],[17],[21]. All malware has their specific purpose and the target, but the main purpose of creating a data network threats and performance of the computer [15]. In general, incoming files are viewed as malware attacks or suspicious which may damage the computer systems. These files break into vulnerable systems in many ways, such as via the internet, removable media, email and Instant Messaging. Anti-malware products in the host machine will look to identify and trace attacks [11]. These unknown types of attack found will be sent to the anti-malware vendors to analyze and update the malware signature database server for the latest updates available for users. To build such a specific signature accurate classification approach is necessary in order to correctly classify the classes of malware and efficient way for the prediction to remove or clean the intrusion attacks.

In general, the computer program is designed as the code and the data. Thus, malicious attack is seen as a manipulation of the code or data in various attacks. The code was the target of the attack, and so it has been studied intensively using existing monitoring malware. In contrast, there has been little focus on the data in research to prevent the malware. This paper proposed an integrated kernel malware classification and detection framework to efficiently classify the malware and detection. It will contribute a malware classifier for effective detection and evaluate the classifier integrating with a secure commitment process for virtual machine service in an operating system.

The following paper is organized as, section-2, which discuss the related works, section-3, which discuss the propose framework for malware classification and runtime detection, section-4, presents experiment evaluation results, and section-5 presents the conclusion.

II. Related Works

Many malware detection mechanisms depend on the properties of the malware code as an unauthorized code injection [14], [18], [19], and malicious code sequence pattern [15], [16]. Although this approach is effective for classic malware, emerging malicious program introducing advanced techniques such as return or jump-oriented programming [5], error code [17], and code emulation [20], [21] for the malware detection mechanism.

Code injection an attack is an unauthorized enter the code into the system's memory space and send a control code injected [6]. Different types of malware, such as computer worms, virus infection [4],[5], the shell code, and rootkits [14] to use the process to change the behavior of malicious purposes. For example, rootkit load the kernel rootkit code into the kernel memory space using the kernel driver or raw memory device. This malware category can be overcome by the imposition of the integrity of the program code and only allowing the execution of approved and non-tampered code. There are various mechanisms to achieve this is to use the existing space in kernel space and also in the hardware [3].

S. Arvind et al., [18] proposed a stronger and more effective monitoring of the integrity of kernel code known as "SecVisor". They decided kernel contrary integrity by checking the operating code injected at runtime. These methods prevent malware nucleus by allowing only authorized achieve kernel code and non-tempered. Dolan-Gavitt et al., [7] proposed a method which discovers data structures in memory images through the use of constraints. This method uses the value of the properties, such as constants, Bitwise AND values and layout to match the data structure. Using fuzzing technique, they suggested that, reliable value characteristic and generate value properties can be used as a signature generation.

It was found that the traditional classes of malicious malware are mainly based on the goal. For example, the class of malware network carrier to use the system to send a copy of itself to another computer to pass, but do not try to change the target. Therefore, by looking at the class of malware and its objectives, policies and best solution to the malware, which can help the most anti-malware to protect from malware system.

Zhiyong S. et al., [2] propose a commitment virtual machine system to get rid of malicious automatically changes the status when you combine the contents of the OS-level VM host. On the basis of the work role call log data, it uses OS-level knowledge and information about malware behavior to recognize malicious changes. As a result, this method is determined by low productivity and the top is different from the current intrusion detection and recovery of damaged systems, see OS collected one by one, it classifies and groups and sets of malicious things on the basis of group-based cluster. This will reduce the false positive rate by identifying clusters harmful, it is easy to simultaneously watch two malicious programs that are of different types and origin of the processes that exhibit these problems, rather than just a single image yourself as done in existing malware detection.

L. Martignoni et al., [10] and U. Bayer et al., [12] stated that malware variants have many attributes and combinations of syntax but to show the same behavior. It is due to very exclusive malware and the difficulty of their analysis and unique types of malware are no longer to be classified easily the most common classes of malware. Thus, the classification of malware has become more complicated and needed a new classification of malware.

III. Proposed Framework For Malware Classification And Detection

This paper presents a novel integrated framework to address the challenges and enables malware detection approaches based on kernel malware data property analysis. The framework consists of two main components, the first component performs the malware, data classification and its signature generation, and the second component performs a run time malware monitoring and detection using a secure commitment process [2] as shown in figure-1. It performs a classification approach for identification of the malware characterization over kernel data structure [8] and clustering using the data object properties to detect kernel malware.

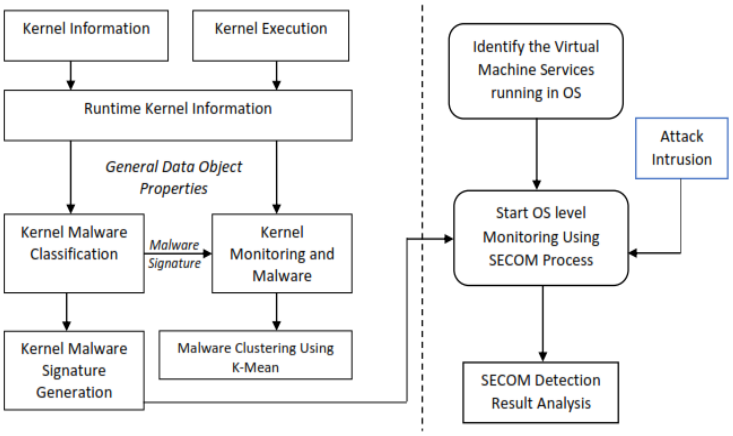

Figure-1: Framework for Malware Signature Generation and Runtime Detection

The propose approach initially, describe the malware signature generation mechanism to obtain the data information and malware characterization over kernel data structure through classification to generate the malware kernel data patterns using the properties of kernel data objects [9]. Later, it discusses the malware detection mechanism which utilizing a malware classifier for labelling and identifying malware intrusion using training datasets knowledge through integrating a secure commitment (SECOM) approach [2] for clearing malware at run-time for the virtual machine services running in OS Level.

3.1. Malware Signature Generation

The malware signature creation process works collected core data information on the core behavior of different type of system activities. The collected data is cleaned using the data cleansing mechanism to eliminate noise in the data. A Rule-based classification approach based on the activities carried out to produce the signature patterns of behavior are called data behavior pattern. The signature is usually a model of sequence command or a series of system calls made for carrying out or completing the work. There will be activity in signature templates stored in the runtime monitoring and intrusion detection systems. To generate a malicious activity malware signature, we run multiple loops as *d* with malware *M*, and apply a set operation on *t* malicious as follows.

Let's assume $A_{M,d}$, and $A_{B,k}$ → represent a data behavior profile for a kind of kernel malware execution *k*, and M_S is called a data behavior signature for a malware *M*, which can be computed using the equation-1 as,

$$M_s = \bigcap_{d\in[1,t]} A_{M,d} - \bigcup_{k\in[1,t]} A_{B,t} \qquad (1)$$

This equation-1, represents that M_S is the set of data behavior that consistently appears in *d* malware runs, but never appears in *t* kind runs. The underlying perception this equation is that the core of malware consistently performs malicious actions during the attacks. This means that we can estimate the harmful behavior by malicious acts intersection. Such behavior does not appear benign, so there benign acts resulting from the union of malware behavior. The main malware instance produces malware signatures to be used for runtime malware detection.

3.2. Malware Detection

Malware Detection detects a malicious program activities carried out in the runtime performance of the core support for malware signature pattern emerges. It characterized the malicious information about the behavior of measuring the behavior of instruction difference relates created a signature model of operation. The reliably characterize the information behavior of the implementation of the core malware dynamic, we will test a number of runs over the core malware signature generated over the runtime malware signature generated for the detection.

Let us consider a likelihood malware program *M* to be present in a tested run *r* which is determined by a deriving set of data behavior elements in M_S which belong to the data behavior profile, A_r. To compute the matching signature for detection we find the set of malware intersection as *D* between M_S and A_r using the equation-2 as given below,

$$D = \{d | d \in M_S \wedge d \in A_r\} \qquad (2)$$

M_S is a data behavior signature of a data behavior profile that is a set of data behavior elements as *d* derived from the intersection and union of data behavior profiles. If D has a variation from a define threshold limit, then it can consider as malware detection. Further analysis was made over the detected malware cluster using k-mean algorithm.

3.3. Securing Virtual Machine Services

Securing the virtual machine (VM) services is an integrated module which performs a runtime malware detection utilizing malware signature. It implements the following process to secure the virtual machine Services.

- ***VM Service Detection:*** Changes in the operating system level to include files, folders and registry entries are created, modified and deleted processes running in the VM. Many companies often have important applications to virtual machines, such as a network server, file server, database server. After leaving the VM valuable data files and applications, such as sales data, customer information, product files, technical files and configuration data generated to be maintained carefully over a secure commitment mechanism.

- ***Secure Commitment Process:*** Safe and secure commitment means to connect only malignant changes in the host environment, but filtering out malicious changes in VM commitments. In order to maintain the integrity of a computer system, the user can operate within a VM to treat emails, download images, edit MS Word files, and web browsing and so on. At the same time, the safety mechanism is the obligation of the host can provide the best results, which can help keep harmful environment.

A protected access virtual machine commitment depends only on OS-level tracking information flows and monitoring the behavior of malware without the technical details of a particular virtual machine requires. Therefore, although it is designed for operating system-level virtual machine, with some changes to apply to other forms of virtual machine malware filtering effect.

IV. Experiment Evaluation

4.1. Datasets

To perform an automatic classification and evaluation of the Malware Characterization we collect datasets from CYBER Systems AND Technology of DARPA Intrusion Detection Scenario Specific [22]. For the training of the malware character classification we process 25 test data files downloaded. We have discovered 20 suspicious malware behaviours on analysis of the datasets.

4.2. Evaluation Measures

To measure the performance of the proposal, we compute the Classification Accuracy by measuring True Positive, False Negative, False Positive, True Negative values. Ideally, the malware detection system must have the attack detection rate (DR) of 100% combined with a false positive (FP) of 0%. Nevertheless, in practice it is very difficult to achieve. The most important parameters involved in the assessment of the effectiveness of systems for detecting malware are shown in Table-1.

Table-1: Parameters for performance Evaluation

Parameters	Definition
True Positive (*TP*) or Detection Rate (*DR*)	Attack occur and alarm raised
False Positive (*FP*)	No attack but alarm raised
True Negative (*TN*)	No attack and no alarm
False Negative (*FN*)	Attack occur but no alarm

For the evaluation test used 37042 trained samples data and 25 malware samples files for the evaluation. The reference value in this experiment is called - the "threshold". Threshold is common control value of the data that has been trained. This work is the first sample prepared learns and control value is set between 50% and 90%, qualified samples. There are 25 files Malware sample data for testing purposes and the same samples were used for each threshold value. In order to evaluate the classifier, accuracy is calculated by using formula as follows,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

4.3. Result Analysis

To analyse the performance of the proposed approach we compared it with existing Naive Bayes and J4.8 classifiers. The NaiveBayes [23] classifier is a learning and probabilistic knowledge based and J4.8 is a extended version of C4.5 algorithm based on tree classifier [24].

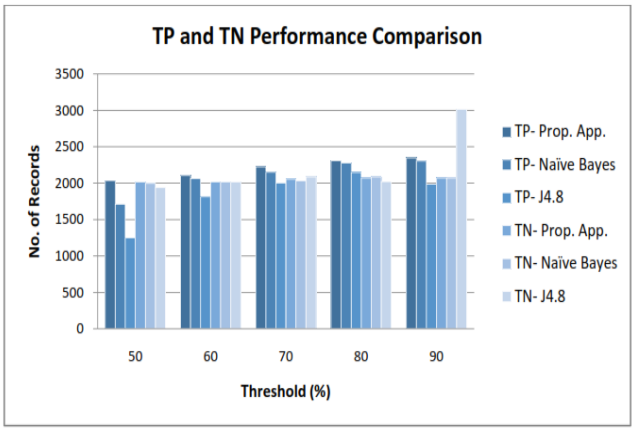

Figure-2: Classification Result Comparison

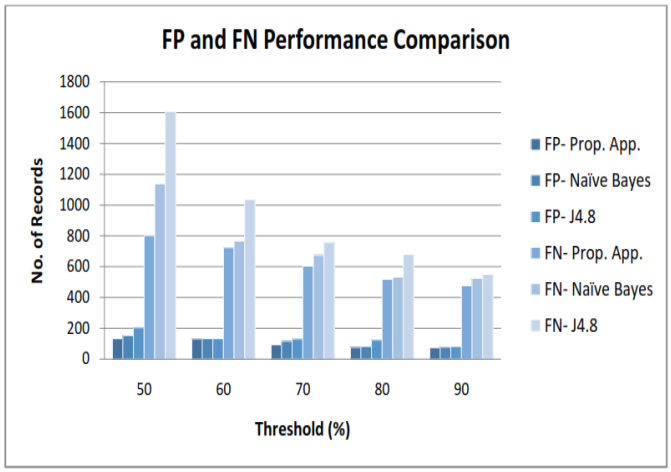

Figure-3: FP and FN performance Comparison

Figure-2, shows a comparison analysis between TP and TN, and Figure-3, shows a comparison between FP and FN performance. TP and TN comparison results shows that proposed approach improvises in the TP rate in compares to others and attains low TN rate. J4.8 shows highest TN rate at 90% detection rate. But it show low FP and FN rate with increasing detection threshold. Proposed approach shows an similar rate of FP rate in comparison but show high TP rate.

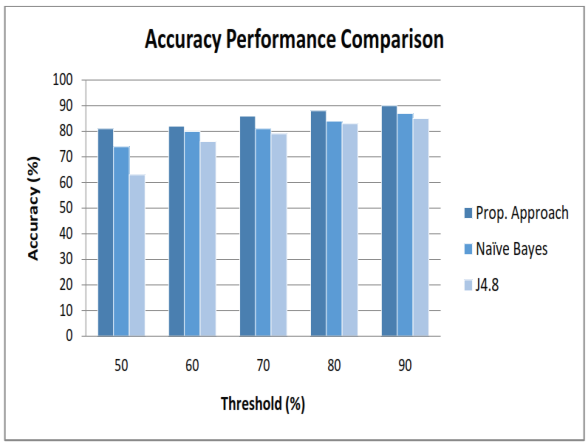

Figure-4: Accuracy Performance Comparison

Figure-4 shows the accuracy comparison results between the classifiers. It shows that with the increasing detection threshold proposed approach shows improvisation in accuracy due to FP and FN decrement which improves the classifier accuracy in compare to others. The classifiers shows an optimal accuracy between 70-90% of threshold limit where all maintains an optimal TP and TN, and a low FP rate.

V. CONCLUSION

Many malware detection mechanisms rely on malware such as code injection unauthorized code and patterns of malicious code sequences. Although this approach is effective in classic malware, new malware is introducing advanced technology, such as a return or jump programming, code obfuscation, and code emulation to avoid these malware detection mechanisms. In this work, we have presented a new approach for classifying kernel malware based on the properties of kernel data objects. The proposed work con composed of two components as Malware Signature Generation and Malware Detection which helps to design a classifier which enable to detect accurate malware on the kernel access patterns. A secure commitment for VM service is integrated for the run time malware detection. Experiment result shows higher accuracy with increasing the classification threshold in compare to the existing classifiers. In can improve by applying the static binary analysis techniques which provide an infrastructure for analyzing binary in future works.

References

[1]. Junghwan R, R Riley, Z Lin, X Jiang, and D Xu, "Data-Centric OS Kernel Malware Characterization" IEEE Transactions on Information Forensics And Security, Vol. 9, No. 1, January 2014.

[2]. Zhiyong Shan, Xin Wang, and Tzi-cker Chiueh, "Malware Clearance for Secure Commitment of OS-Level Virtual Machines", IEEE Transactions on Dependable and Secure Computing, Vol. 10, No. 2, March/April 2013.

[3]. Junghwan R. and Dongyan Xu., "LiveDM: Temporal Mapping of Dynamic Kernel Memory for Dynamic Kernel Malware Analysis and Debugging". Technical Report CERIAS TR 2010-02, Purdue University, West Lafayette, Indiana, 2010.

[4]. K. Rieck, T.Holz, C.Willems, P.Dussel, and P.Laskov, "Learning and Classification of Malware Behavior", Proc.Fifth Int'l Conf. Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), pp. 108-125, June 2008.

[5]. Ping Chen, Xiao Xing, Bing Mao, Li Xie, Xiaobin Shen, and Xinchun Yin, "Automatic Construction of Jump-oriented Programming Shellcode". In Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS'11), 2011.

[6]. J.Li, Z.Wang, T.Bletsch, D.Srinivasan, M.Grace, and X.Jiang, "Comprehensive and Efficient Protection of Kernel Control Data", IEEE Transactions on Information Fo-rensics and Security, 6(2), June 2011.

[7]. B. Dolan-Gavitt, A. Srivastava, P. Traynor, and J. Giffin, "Robust Signatures for Kernel Data Structures", In Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09),2009.

[8]. Z. Lin and J. Rhee and X. Zhang and Dongyan X. and Xuxian J., "SigGraph: Brute Force Scanning of Kernel Data Structure Instances Using Graph-based Signatures", In Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS'11), San Diego, CA, February 2011.

[9]. M. Carbone, et al., "Mapping kernel objects to enable systematic integrity checking", Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009.

[10]. L. Martignoni, Paleari, R., Bruschi D.,"A Framework for Behavior-Based Malware Analysis in the Cloud". In 5th International Conference on Information Systems Security. Vol. 5905, pp. 178-192. Berlin, Heidelberg: Springer-Verlag, 2009.

[11]. Z. Shan, X. Wang, and T. Chiueh, "Tracer: Enforcing Mandatory Access Control in Commodity OS with the Support of LightWeight Intrusion Detection and Tracing", Proc. Sixth ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 135-144, Mar. 2011.

[12]. U. Bayer, Habibi, I., Balzarotti, D., Kirda, E., Kruegel C.,"A View on Current Malware Behaviors", In Usenix Workshop on Large-scale Exploits and Emergent Threats. Berkeley, CA, USA: USENIX Association, 2009.

[13]. C. Xuan, John A. Copeland, and R. A. Beyah, "Toward Revealing Kernel Malware Behavior in Virtual Execution Environments", In Proceedings of 12th International Symposium on Recent Advances in Intrusion Detection (RAID'09), pages 304–325, 2009.

[14]. R. Riley, X. Jiang, and Dongyan Xu.,"Guest-Transparent Prevention of Kernel Rootkits with VMM-based Memory Shadowing", In Proceedings of 11th International Symposium on Recent Advances in Intrusion Detection, 2008.

[15]. D. Balzarotti, M. Cova, C. Karlberger, C. Kruegel, E. Kirda, and G. Vigna, "Efficient Detection of Split Personalities in Malware", In Proceedings of the 17th Annual Network and Distributed System Security Symposium,2010.

[16]. C. Kolbitsch, P. Milani, C. Kruegel, X. Zhou, and X. Wang, "Effective and Efficient Malware Detection at the End Host", In Proceedings of the 18th Usenix Security Symposium (Security'09),2009

[17]. R. Riley, X. Jiang, and D. Xu, "An architectural approach to preventing code injection attacks," IEEE Trans. Dependable Secure Comput.,vol.7, no. 4, pp. 351-365, Dec. 2009.

[18]. A. Seshadri, Mark Luk, Ning Qu, and Adrian Perrig, "SecVisor: A Tiny Hypervisor to Provide Lifetime Kernel Code Integrity for Commodity OSes", In Proceedings of 21st Symposium on Operating Systems Principles, ACM, 2007.

[19]. X. Jiang, X. Wang, and Dongyan Xu., "Stealthy Malware Detection through VMM-based 'Out-of-the-Box' Semantic View Reconstruction", In Proceedings of the 14th ACM Conference on Computer and Communications Security, October 2007.

[20]. W. Sun, Z. Liang, R. Sekar, and V.N. Venkatakrishnan, "One-Way Isolation: An Effective Approach for Realizing Safe Execution Environments", Proc. 12th ISOC Network and Distributed Systems Symp., pp. 265-278, 2005.

[21]. C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, et al., "StackGuard: Automatic adaptive detection and prevention of buffer-overfow attacks," in Proc. 7th USENIX Sec. Conf., pp. 63-78, Jan. 1998.

[22]. US-CERT. Vulnerability Notes Database. http://www.kb.cert.org/vuls/.

[23]. Panda M. and M R Patra, "A Comparative Study Of Data Mining Algorithms For Network Intrusion Detection", First International Conference on Emerging Trends in Engineering and Technology, pp 504-507, IEEE, 2008.

[24]. S. Thaseen and Ch. Aswani K, "An Analysis of Supervised Tree Based Classifiers for Intrusion Detection System", International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), IEEE, Feb-2013.