PARM: A NOVEL POSITIVE ASSOCIATION RULE MINING ALGORITHM FOR DISCOVERING MALEVOLENT APPLICATIONS IN WINDOWS OPERATING SYSTEMS

Chandrasekar R, N. Deepa

Assistant Professor, School of Information Technology & Engineering,
VIT University, Vellore-632 014, Tamil Nadu, India.
chandrasekar.r@vit.ac.in, deepa.rajesh@vit.ac.in

***Abstract*-The most important vulnerability to the current World Wide Web is the malevolent applications. Generally, these applications are used for interrupting the normal functioning of a system and accessing unprivileged and confidential data and other wicked activities. Malevolent applications were primitively designed to spread from one host to another, but in recent past their behavior has converted to complex, highly developed, sophisticated nature to pinch personal and confidential data. Also, some of these applications can be more dangerous by infecting organizations and steal identities. An application can be efficiently categorized as malevolent or normal application by observing the characteristics of the application while it is executing in the host. The majority of the present methods for discovering malevolent applications make use of the information present in the system calls. The projected work discovers malevolent application by using the order in which the system calls are being made by the application. A 5^{th} order Markov chain is chosen for representing the transition of system calls. This attribute set is used for differentiating malevolent and normal applications. Positive Association Rule Mining (PARM) uses the attributes that are available in the dataset and also results in higher detection rate and detection time than traditional data mining methods like Decision Tree (DT), Support Vector Machine (SVM) and Naive Bayes (NB). Not all but only the core system calls are monitored to sustain high detection rate and detection time. The efficiency of PARM is increased by avoiding redundant rules. The performance of PARM is evaluated by measuring the detection rate and detection time and comparing them with those of some of the present data mining based systems for discovering malevolent applications. PARM has been implemented and observed that it performs better than the existing techniques for discovering malevolent applications.**

Keywords: Positive Association Rule Mining, Security, Markov Model, Frequent Itemset

I. INTRODUCTION

Malevolent applications enter a host system without the permission and knowledge of the user [1]. One of the biggest threats in the current arena of computing is the malevolent application. Its growth is continuous in volume and complication. Many organizations are trying to provide solution for this problem, yet many malevolent applications had increased in the recent past. The main portal of entry for these kinds of malevolent applications is the internet. After entering the host system, the malevolent application reduces the efficiency of that system by discovering the vulnerabilities present in the system and then performing wicked activities in that system. The malevolent applications have certain common features among them [2]. These applications perform more than one type of action and the program consists of multiple modules. Malware is available and user-friendly. Malevolent applications are easy to use. They can infect a wide variety of hardware and software. Malevolent applications help to earn lot of illicit money. A malevolent application analyzer maps the given set of applications into one of the following categories namely malevolent and normal [3].In other words: Malevolent Analyzer (Application) = Malevolent, if p contains illicit code, Normal, otherwise. The analyzer analyzes the application to check if the application is malevolent or normal. The analyzer discovers the malevolent application based on its signature. The machine code of a specific virus is known as signature. The signatures of the malevolent programs in the database are compared with the file system and removable devices as well as within other memory devices. Static, Hybrid and Dynamic signature-based detections use the above for analysis [4]. The paper is structured as follows. Section II presents the literature survey. The proposed design, methods and illustrations are presented in Section III, IV and V respectively. Section VI presents the experimental results. Section VII discusses the conclusion of the paper.

II. LITERATURE SURVEY

A tool was introduced by Faraz, Haider, Zubair, Muddassar [5]. It analyzes the order in which the system calls in Windows is made and processes it using usual machine learning algorithms to find the abnormal program. They performed experiments and have deduced that using system calls increases the accuracy of the system. They have used only a reduced and significant set of system calls and yet have achieved good performance. A method called Object-oriented utility based association rule mining algorithm was introduced by Yi-Dong, Zhong and Qiang [6]. The purpose and the usefulness of the patterns are modeled in this approach. Because the purpose and usefulness of the patterns are taken as the main factor, this method totally differs from most of the so far proposed methods. An intelligent detection system was proposed by Yanfang, Dingding, Tao, Dongyi and Qingshan [7, 8]. They used association rule mining algorithms for classifying exe programs using their system calls. A huge number of exe programs were gathered from a security organization and they were used for comparing the various methods. Their experiments conducted using the system shows that the performance of the system using association rule mining is better than the performance showed by various security software and other existing systems. The post processing method was thoroughly studied by Yanfang, Tao, Qingshan and Youyu and association rules were proposed for discovering the abnormal programs. They proposed a technique to discover the abnormal program from the gray list. They used the above studied processing method. This technique was incorporated into their former system and the new system was developed, CIMDS [10]. They performed various experiments and proved that the performance of their system was better than the performance of the existing methods that used the same technique. In contrast, the novelty of the projected malevolent application detector is that it is dynamic, behavior-based and it uses positive association rule mining technique and also discovers malevolent applications while they are executing. 5^{th} order Markov chain is used for designing the system call order and also analyses only a fixed set of system calls.

III. PROPOSED DESIGN

Design of projected malevolent application discovery system is shown in Fig 1. The malevolent applications are discovered while they are getting executed, which makes this as a dynamic technique for discovering malevolent applications. The order in which the system calls are made is captured while the application is executing and the Positive Association Rules are applied [11]. A 5^{th} order Markov chain is utilized to design the order of the system calls. The system (PARM) consists of 2 modules: learning and discovery stage.

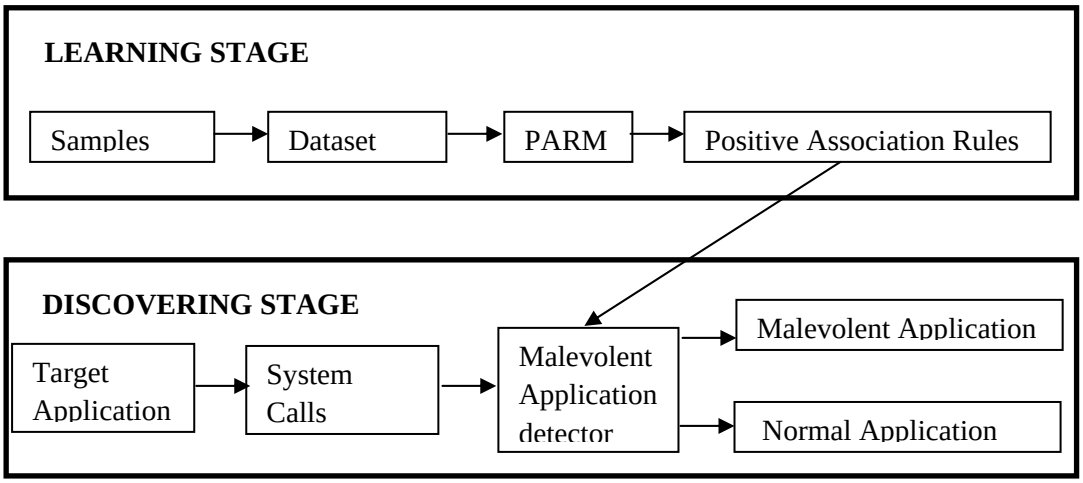

Fig 1. Design of the PARM system

The learning stage does the following functions: gathering the samples, analyzing the samples, preparing the dataset from the samples, generating the positive association rules and obtain frequent patterns from the dataset transaction along with support and confidence. The dataset is a collection of malevolent applications as well as normal applications. Malevolent applications have been obtained from a publicly available database [12] and normal applications have been gathered from newly installed Windows operating systems. The Windows operating systems include Windows XP, Windows Vista and Windows 8. The order in which each application, both normal and malevolent, makes the system calls is hooked and monitored for all the applications in the sample set. This series representing the series of system call along with the label (malevolent or normal) is stored in a database. This database forms the dataset. The lengthy transactions from the dataset are broken down into 6-grams i.e. they are modeled as the 5^{th} order Markov chain and their corresponding labels are retained. The support and confidence of each 6-gram is calculated. The 6-gram along with its label, support and confidence forms the rule and is stored in the frequent itemset. The rule from the frequent itemset which is having the threshold confidence and support are filtered into another database which forms the reduced frequent itemset.

The definition of the Support(S) and the confidence(C) are shown below.

S = [Count (X U {label}, S) / N] x 100%

C = [Count (X U {label}, S) / Count (X, S)] x 100%

Where,

X is the 6 gram, Count (X, S) is the no. of records in the S containing X, Count (X U {label}, S) is the no. of records in the S in which (X U {label}) holds true, label represents malevolent or normal, S is the dataset, N is the no. of records in S

The discovery stage performs the following operations: the target application will be executing, while the PARM hooks and monitors the series of system calls made by the target application. Now the series of system calls of the target application would be ready. The malevolent application detector analyses this series using the frequent itemset generated during the learning stage and decides whether the executing application is malevolent or normal.

IV. MATERIALS AND METHODS

The method for hooking and monitoring the order of system calls made by each application and then discovering positive association rules is called "PARM Learner" shown in Fig 2 and the method for discovering whether the applications is normal or malevolent, based on the output of PARM Learner, is called PARM Detector shown in Fig 3. PARM Learner hooks and monitors the order in which the system calls are made by the application in the gathered samples (both malevolent and benign). This series of system calls are converted into 6 grams. The support and confidence of the 6 grams are calculated and this along with their corresponding label forms the rule, which is put in the frequent itemset. The rules that don't have the threshold support and confidence are removed from the frequent itemset. PARM Detector hooks and monitors the order of system calls made by the target application during its execution. This series is converted into 6 grams. The confidence of the 6 grams in the target application series is substituted and total confidence pertaining to malevolent and normal labels are calculated separately. If the average of confidence of 6 grams of malevolent label is greater than that of normal class, then the target application is malevolent, else it is normal.

Input: Application sample
begin
1. Hook system calls and monitor application
2. Split sequence into 6 grams
3. Name them with their corresponding label
4. **for** (each 6 grams)
5. Calculate support and confidence
6. Discard 6 grams without minimum confidence and support

Fig 2: Parm Learner

***Input*:** Association rule and target application
begin
1. Hook system calls and monitor application
2. Split sequence into 6 grams
3. ***for*** (each 6 grams)
4. Calculate confidence of being normal and malevolent
5. A = avg confidence of malevolent 6 grams
6. B = avg confidence of normal 6 grams
7. ***if*** A > B ***then***
8. target application is MALEVOLENT

Fig 3: Parm Detector

V. EXAMPLE SCENARIO

The design and methods projected in the recent sections are illustrated with an example. In this illustration, English alphabets from A to Z are considered as system calls. The training samples consist of 2 malevolent and 2 normal applications. Each application in the training sample, when executed, yields a process which is given as input to PARM Learner method. This method hooks the process and monitors the system calls of the executing process, shown in Table I. PARM mines frequent patterns and generate the rules. All the 6 grams of the system call sequence of each training application in training samples is obtained and labeled with its corresponding label (malevolent or normal) as shown in Table II. The support and confidence of all itemsets in dataset are calculated which yields the frequent itemset, shown in Table III. The itemsets in frequent itemset with support value less than 50% (say) and confidence value less than 70% (say) are removed from the frequent itemset. The remaining rules form the reduced frequent itemset, shown in Table IV.

TABLE I
System Calls of Samples

System Call	Application	Label	System Call	Application	Label
ABCDEFGH	Malevolent_1.exe	Malevolent	RSTUVWXY	Normal_1.exe	Normal
JKLMNOPQ	Malevolent	Malevolent	EFGHIJKL	Normal	Normal

TABLE II
Dataset

6 Grams	Label	6 Grams	Label	6 Grams	Label	6	Label
ABCDEF	Malevolent	RSTUVW	Normal	JKLMNO	Malevolent	EFGHIJ	Normal
BCDEFG	Malevolent	STUVWX	Normal	KLMNOP	Malevolent	FGHIJK	Normal
CDEFGH	Malevolent	TUVWXY	Normal	LMNOPQ	Malevolent	GHIJKL	Normal

TABLE III
Frequent Itemset

6 Grams	Support (say)	Confidence (say)	Label	6 Grams	Support (say)	Confidence (say)	Label
ABCDEF	30	70	Malevolent	RSTUVW	60	40	Normal
BCDEFG	40	60	Malevolent	STUVWX	35	65	Normal
CDEFGH	50	50	Malevolent	TUVWXY	45	55	Normal
JKLMNO	90	75	Malevolent	EFGHIJ	55	45	Normal
KLMNOP	70	30	Malevolent	FGHIJK	65	35	Normal
LMNOPQ	80	80	Malevolent	GHIJKL	75	85	Normal

TABLE IV
Reduced Frequent Itemset

6 Grams	Support	Confidence	Label
LMNOPQ	80	80	Malevolent
JKLMNO	90	75	Normal
GHIJKL	75	85	Normal

PARM Detector classifies target application as malevolent or normal using frequent itemset and system call sequence of target application. The above sequence is broken into 6 grams and using the frequent itemset, the confidence value of all 6 grams is substituted. The average confidence value of all 6 grams malevolent label and normal label are calculated separately. Thus if the average confidence value of 6 grams in malevolent label is greater than that of normal label, the target application will be classified as malevolent else the target application will be classified as normal, as shown in Table V.

TABLE V
Detection

System call of Target Application	Confidence		Result
	Normal (%)	*Malevolent (%)*	
JKLMNOPQ	75.00	80.00	Malevolent
GHIJKLM	00.00	80.00	Normal

VI. IMPLEMENTATION, RESULTS AND DISCUSSIONS

The projected malevolent application detection system has been evaluated using detection rate (DR) and detection time (DT) as defined in [10]. Detection Time is defined as the time taken to classify the process as benign or malware. Detection Time is the time (in seconds) needed for discovering the label of the target application. Detection Rate is defined as the ratio of TP to sum of TP and FN.

$$DR = [TP / (TP + FN)] * 100\ \%$$

Where,

True positive (TP) is the number of malevolent application classified as malevolent and False negative (FN) is the number of malevolent application classified as normal.

Windows executable applications both In normal and malevolent categories were gathered from a public database [12]. The normal application executables were collected from newly installed Windows operating system. The methods were coded in Microsoft Visual C++. The IAT hooking technique [13, 14, 15] was used for hooking and monitoring the system calls made by the application samples. Around 500 applications were selected as samples out of which around 150 were normal applications and the rest 350 were malevolent applications. DR and DT of existing systems were tabulated in [10]. The DR of the projected malevolent discovery system is 12% more than that of existing systems using training set and 14% more than that of existing systems using testing set. The DT of the projected malevolent discovery system is 60% less than that of existing systems using training set and 40% less than that of existing systems using testing set. Figs 4 and 5, showing the DR and DT of various systems, illustrates that the proposed projected malevolent discovery performs better than several existing systems.

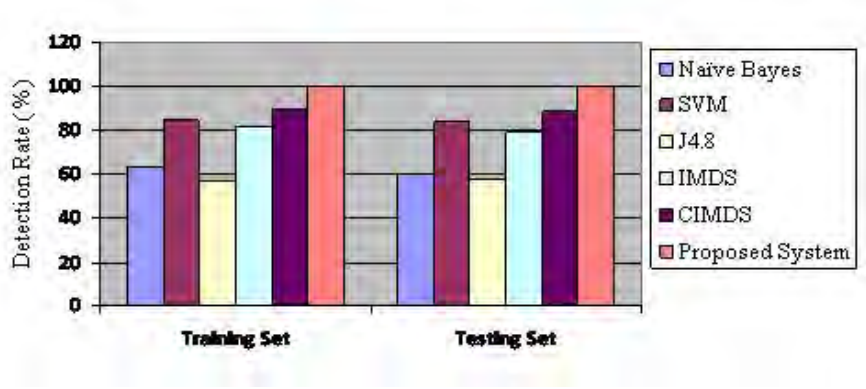

Fig 4. DR of different malware detection systems

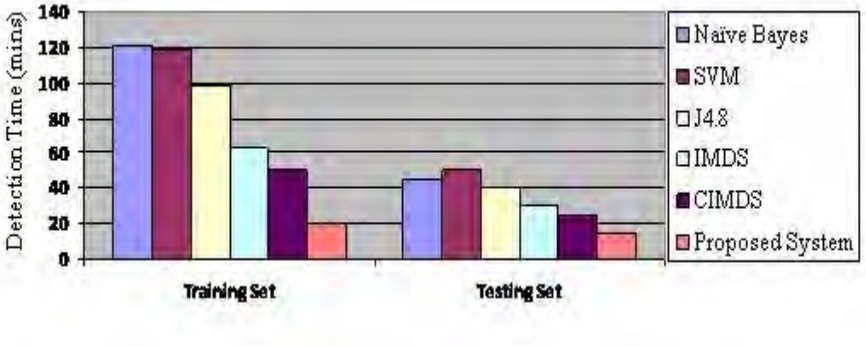

Fig 5. DT of different malware detection systems

VII. CONCLUSION

A malevolent discovery system is projected which used system calls for discovery of malevolent applications. A 5th order Markov chain models the system calls. This characteristics set is given as an input to the malevolent discovery system. Positive Association Rule Mining is used to identify frequent patterns and it gives higher detection rate and less detection time compared to existing mining based discovery system. Only important system calls are monitored yet Detection Rate is high and Detection Time is less. It is inferred that the projected malevolent discovery system performs better than previous systems.

REFERENCES

[1] Rizwan Rehman, G.C. Hazarika and Gunadeep Chetia, "Malware Threats And Mitigation Strategies: A Survey",Journal of Theoretical and Applied Information Technology, Vol.29, No.2, pp.69-73, July 2011.
[2] Journal of Theoretical and Applied Information Technology, Vol. 29, No. 2, pp. 69-73, July 2011.
[3] OECD Ministerial Meeting Report, "Malicious Software (Malware): A Security Threat to the Internet Economy", Korean Communication Commision, Final draft, May 2007.
[4] Vinod P, V.Laxmi and M.S.Gaur, "Survey on Malware Detection Methods", in Proceedings of the Hacker 2009, pp. 74-79, 2009.
[5] Nwokedi Idika andAditya P. Mathur, "A Survey of Malware Detection Techniques", SERC Library, February 2007.
[6] Faraz Ahmed, Haider Hameed, Zubair Shafiq M and Muddassar Farooq, "Using Spatio-Temporal Information in API Calls with Machine Learning Algorithms for Malware Detection," in Proceedings of the 2nd ACM Workshop on Artificial Intelligence and Security (AISec 2009), pp. 55-62, Nov. 2009.
[7] Yi-Dong Shen, Zhong Zhang and Qiang Yang, "Objective-oriented utility-based association mining," in Proceedings of the IEEE International Conference on Data Mining (ICDM 2003), pp. 426-433, Dec. 2002.
[8] Yanfang Ye, Dingding Wang, Tao Li and Dongyi Ye, "IMDS: Intelligent malware detection system," in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07), pp. 1043–1047, Aug. 2007.
[9] Yanfang Ye, Dingding Wang, Tao Li, Dongyi Ye and Qingshan Jiang, "An intelligent pe-malware detection system based on association mining," Journal in Computer Virology, vol. 4, pp. 323–334, Feb. 2008.
[10] Jiawei Han and Micheline Kamber, Data mining: Concepts and Techniques, Morgan Kaufmann publishers: San Francisco, 2nd edition, 2006.
[11] Yanfang Ye, Tao Li, Qingshan Jiang and Youyu Wang, "CIMDS: Adapting Postprocessing Techniques of Associative Classification for Malware Detection," IEEE Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews, vol. 40, No. 3, pp. 298-307, May 2010.
[12] "Overview of the Windows API", Available at: http://msdn.microsoft.com/en-us/library/aa383723(VS.85).aspx.
[13] "VX Heavens Virus Collection", Available at: http://vx.netlux.org/.
[14] "IAT-Hooking-Revisited", Available at: http://www.autosectools.com/IAT-Hooking-Revisited.pdf.
[15] "Understanding the Import Address Table", Available at: http://sandsprite.com/CodeStuff/Understanding_imports.html.
[16] "IAT Function Hooking", Available at: http://sandsprite.com/CodeStuff/IAT_Hooking.html.