Detection of Malicious Data using hybrid of Classification and Clustering Algorithms under Data Mining

Milan Jain
Research Scholar
Department of CSE
Chandigarh Engineering College.
Mohali, Punjab, India

Bikram Pal, PhD
Professor
Department Of CSE
Chandigarh Engineering College.
Mohali, Punjab, India

ABSTRACT

In today era modern infrastructures and technologies are more prone to various types of accesses. A method that is commonly used for launching these types of attack is popularly known as malware i.e. viruses, Trojan horses and worms, which, when propagate can cause a great damage to commercial companies, private users and governments. The another reason that enhance malware to infect and spread very rapidly is high-speed Internet connections as it has become more popular now a days, therefore it is very important to eradicate and detect new (benign) malware in a prompt manner. Hence in this work, proposing three data mining algorithms to produce new classifiers with separate features: RIPPER, Naïve Bayes and a Multi Classifier system along with hybrid of clustering techniques and the comparison between these methods to predict which provides better results.

Keywords

Malicious Code Detection; Data Mining; Computer Security; Prediction

1. INTRODUCTION

Rootkits is type of software that hinders the presence and activity of malware (such as viruses, worms and trojans) and allow attacker to capture a computer system. The first basic thing that an attacker does is to install rootkit after access is gained to a system, as this will conclude that the attack is going to remain undetected. However the attacker can then further began to store personal data, such as details of bank account, passwords, and credit card numbers. In general, Rootkits use several kind of hooking techniques such that they remain hidden and there are also various tools available, such as McAfee’s Rootkit Detective, that are used in detection of hooks which are designed by a rootkit on a computer system. Every time when this tool is run, there is generation of a log file which contains a list of the hooks that have been detected. The amount of information in these log files is as intense as they are having full knowledge about every type of hook that has been detected on the computer system [3]. We can say on an average, each and every of these log files holds several hundred lines of data. In order to assure the integrity of computer structure, both in terms of privacy and security is a very important concern in general. To know the mechanisms to protect the nation’s networks and computers that we have to apply, the first basic step is understand the types of threats. Hence the threats have been classified into two categories real-time threats as well as non real-time threats. A real-time threat is kind of threat which are helpful in preventing several catastrophic situation but within limited period. Non real time threat is another form of threat in which time boundation is not properly stated in order to prevent risk. The thing which we have to notify is that non real-time threats can easily transform into real-time threats as new information is revealed [1].

2. DATA MINING

Data mining (the analysis step of the "Knowledge Discovery in Databases" process), an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The basic aim of this process is that information is extracted from a data set and then it gets changes in a structure that is easily understood for its use. Aside from the step of raw analysis, it includes database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, visualization, complexity considerations, online updating and post-processing of discovered structures.

The real task of data mining is the instinctive or semi-instinctive analysis of big quantities of data to draw out previously not known absorbing patterns i.e. unusual records (anomaly detection), group of records of data (analysis of cluster) and dependencies (association rule mining). This actually involves through techniques of techniques such as spatial indices. Then these patterns can be viewed as a type of upshot of the input data, and may also be used in additional synthesis or can say, for example, in machine learning and predictive analytics.

3. MALICIOUS CODE DETECTION

Detection of malware executables is nothing new in security. Earlier signature method is used for detection of malicious programs [7]. These signatures were composed of many different properties: text strings, filename or byte code. Research also focused on preventing it from the security holes which are created by malicious programs. Experts were typically employed to find doubtful programs by hand. Using their expertisation, signatures were identified that made a malware data example distinct from other malicious data or unseen programs. However, some malware is assumed as genuine software, and it often comes from an official company website of company in the form of attractive or useful program which contains the harmful malware along with the addition of tracking software that collects marketing statistics. Software such as anti-malware, anti-virus and firewalls are integrated upon by users even at home, large and small enterprises around the globe to protect against virus attacks which helps in preventing and finding the further extend of malicious data in the network [1].

Thus it can be said that a malicious data is also known to be a program that are performing a malicious function, such as damaging a system, compromising a system security or obtaining sensitive information without the permission of user [8]. Using data mining methods, our target is to automatically build and design a scanner that correctly detects malware executables before they are given a chance to execute. Data mining methods find patterns in large quantity of data, such as byte code, and use them in detecting future instances in case of similar data. Our framework uses classifiers for the detection of new virus executables. A classifier is a detection model, or rule set figured by the data mining algorithm which was trained over a particular set of training data. One of the major problems faced by the malware community is to devise methods for detection of new virus programs which have not yet been detected. Eight to ten virus programs are created daily and most of them cannot be correctly detected until signatures are generated by them. During this phase, systems defended by signature-based algorithms are exposed to attacks.

4. LITERATURE SURVEY

The main criteria of research is to show the utilization of data mining methods that are exploring the utility of drawn out features so that the type of malware can be easily predicted. Rootkit prediction is not the new thing, so it gives chance to various activities of malicious data to generate viruses and rather some malware programs can even break the security and privacy thereby, infecting the system. Various techniques have been proposed by different researchers in recent years. Some of them have been mentioned here.

In the paper**"Rootkit (Malicious Code) Prediction through Data Mining Methods and Techniques"** Ramani et.al [3] described that rootkit is known as software that is used to hide the presence aynd activity of malwarve and allow an attacker to take control of a system thereby, affecting it. This study reveals the application of data mining method to identify rootkits depending on attributes drawn from the data present in log files. The recordes of the rootkit data are divided into types of categories one is Inline and the other is dependent on the attribute values. Here investigation of nine algorithms under classification is done to predict the better and proper classifier for the rootkit. Out of 9 Correlation Bayes algorithm achieves the best accuracy level by using 10-fold cross validation on the rootkit dataset and this algorithm gains the highest MCC. To confirm the algorithm performance on the basis of improper data, the Cofficient of Mathews Correlation was calculated.

Tangle et. al [4] in the paper, **"*Evolution, Detection and Analysis of Malware for Smart Devices*"** It is showing malware in current smart devices that equipped with networking capabilities, computing and powerful sensing have proliferated lately, range from famous smart android phones and tablets to Internet devices, smart TVs, and others that will soon occur. One main feature of devices is that they are capable to incorporate with third-party applications from markets. This has very strong security features and secrecy problems to user and infrastructure operator, specifically via software of malicious nature that got access to the service given by the devices and gathers the useful and personal data. Malware in latest smart devices – Smart phones and tablets– has got fame in the previous few years, in some cases supported by best techniques designed to provide better security architecture presently in use by these devices. As important advances have been made on malware detection in computers in the last decades it is still a big problem.

In the paper " **Detecting Malicious Code by Model Checking**" Kinder et. al [5] explained a model checking method for finding malicious code. The author represents a soft method to find malicious code that sets in running files by using a method known as model checking. While this method was introduced to find the accuracy of system against specifications, author say that it permit equally well to the finding of malicious code patterns. In last, they introduced the partircular language Computation Trees Predicate Logics which is enlarging the well-known logics CTL and allow describing about an efficient model checking method for their practical experiments explaining that they are able to find a large number of worm variants with specific characteristics.

Schultz et. al [7] in the paper" **Data Mining Methods for Detection of New Malicious Executables"** is explaining different methods of finding a harmful executables. These harmful executables are formed at the very high rate every year and make a serious security threat. The anti-virus systems use to detect these malware programs with heuristics make by hand. This method is very expensive and sometime less-effective. This paper is presenting a data-mining platform that finds new harmful files effectively and automatically. This platform automatically detects patterns in their data set and used these detected patterns to find a set of new harmful data. Comparing this method with a classical signature based method; the new method gives double the current detection rates for benign harmful file. Here harmful means malicious

In the paper**" *Applying Machine Learning Techniques for Detection of Malicious Code in Network Traffic"*** Elovici et. al [10] addressed that the Alert Early Detection, and Response (eDare) system is designed at purifying Web traffic circulating through the real estate of Network Service Providers (NSP) due to malware. To attain this goal, the system hire powerful network traffic scanners that are capable of removing traffic from known malware code. The leftout traffic is supervised and Machine Learning (ML) algorithms are executed in an attempt to pinpoint benign malicious code showing doubtful morphological patterns. Bayesian Networks, Neural Networks and Decision trees are useful for analysis of static code in order to determine whether a doubtful executable file actually inhabits malware.

5. PROPOSED METHODOLOGY

Thus the data mining methodology formulated for rootkit prediction is diagrammatically presented in fig 1. It consists of rootkit data collection, data pre-processing, and classification and performance evaluation phases. The objectives of research are:-

1. To implement the malicious code detection techniques in data mining.

2. To study and compare three classification approaches that are RIPPER, Naïve Bayes and Multi- naive bayes along with clubbing of clustering techniques that are KNN (Nearest Neighbor), SVM (Support Vector Machine) and Decision Tree.

3. Testing this method over a set of malicious executables.

4. To find out parameters ACC (Accuracy), precision, recall etc.

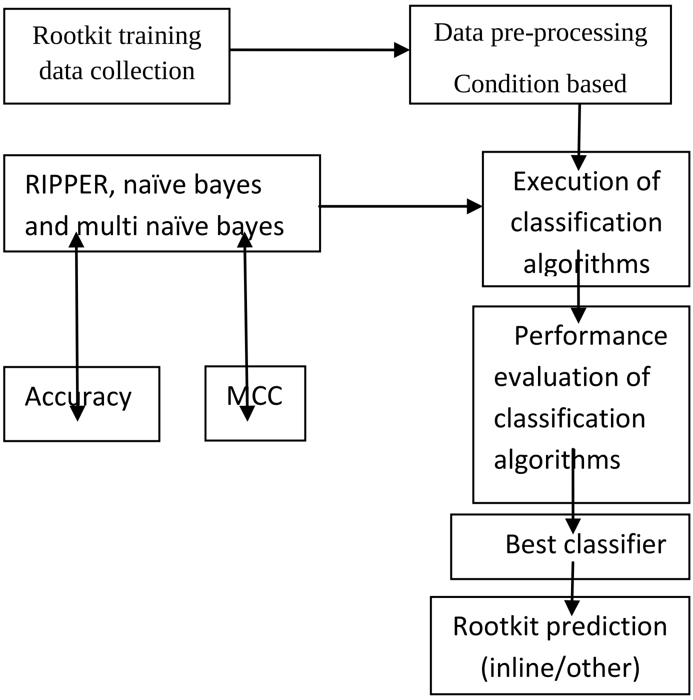
Rootkit training data collection
Data pre-processing
Condition based
RIPPER, naïve bayes and multi naïve bayes
Execution of classification algorithms
Performance evaluation of classification algorithms
Accuracy
MCC
Best classifier
Rootkit prediction (inline/other)

Fig 1:- Data mining methodology for rootkit prediction

Outline of the flowork

Step1 Load the root kit data.

Step2 Preprocess the data by applying clustering methods which are:-

- KNN(Nearest Neighbour) Algorithm

$$D (x,y) = \sqrt{\sum_{j=i}^{d} w_{j^2}(x_i - y_i)^2}$$

- SVM (Support Vector Machine)Algorithm

$$L = \frac{1}{2} |\omega|^2 - \sum_i . \alpha_i \ (y_i \left(\underset{\omega}{\rightarrow} . \underset{x_i}{\rightarrow} + b \right) - 1)$$

- Random tree

$$E (T, X) = \sum_{c \in X} P(c) \, E(c)$$

Step 3 Now applying classifiers which are as follows:-

Ripper

Assign learning rules for clustering of data to detect malicious executables.

It will built set of rules which reduce ambiguity and cluster them into the respective classes.

It will simplify unclassified rules by doing training.

W(R) = (p-n)/ (p+n)

p: number of positive examples covered by the rule in the validation set

n: number of negative examples covered by the rule in the validation set

Naïve Bayes

The classifier we are using calculates likelihood that also cluster is having malevolent codes given the features that are present in given cluster. The string or byte sequences in this method contain same feature as signature and instruction to the machines.

$$P(c/x) = \frac{P(x/c) \, P(c)}{P(x)}$$

$$P(c/X) = P(x_1/c) \times P(x_2/c) \times \ldots\ldots. \times P(x_n/c) \times P(c)$$

- $P(c/x)$ is the posterior probability of class (target) given predictor (atribute).
- P(c) is the prior probabiltiy of class.
- $P(x/c)$ is the likelihood which is the probabilty of predictor given class.
- P(x) is the prior probabilty of predictor.

Multi naïve bayes

To calculate a collection of cluster for ambiguity or malicious code

$$p\,(F/C) = \frac{(\sum_i F_i)!}{\prod_i F_i\,!} \prod_i {p_i}^{F_i}$$

It will generate set of rules and multiplication value for prediction of classifiers.

Step 4 The best classifier that is used detects the ambiguity in the data.

Step 5 Generate Root kit value for better prediction.

Step 6 Compare all the classifier prediction valued analyze

Generally the classification phase predicts that which algorithm will show the better performance i.e. which yields the best accuracy after all the preceding algorithms are executed.

Thus the goal is to implement the malicious code detection techniques in data mining that would produce the best predictive performance for rootkit prediction i.e. to show the following measures were utilized for the rootkit prediction using classification algorithms along with the clubbing of clustering algorithms in data mining.

6. CONCLUSION

As important contributions have been already made on malicious executables detection in personal computers in the previous decades which are shown in previous works. Thus these malwares are very harmful as they have lot of disadvantages on the disordered machines like disabling AV scanners or malware detectors which are installed for security reasons. However there is more need to adopt some better techniques which can ensure the malware code detection efficiently by testing method over a large set of malicious executables. In this work, implementation of three algorithms named as RIPPER, Naives Bayes approach, and Multi-Naïve Bayes using data mining techniques and the comparison of these algorithms is done to evaluate the parameters. In addition likely to make it more capable in terms of time and space and are also planning to enhance the system performance in both accuracy and time with a given data set.

7. REFERENCE

[1] Bhavani Thuraisingham, Latifur Khan, Mohammad M. Masud, Kevin W. Hamlen,"*Data Mining for Security Applications* ",2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, pp. 585-589.

[2] Boldt, M. ; Dept. of Syst. & Software Eng., Blekinge Inst. of Technol., Ronneby ; Jacobsson, A. ; Lavesson, N. ; Davidsson, P., "*Automated Spyware Detection Using End User License Agreements"* Information Security and

Assurance, 2008. ISA 2008. International Conference on 24-26 April 2008; 978-0-7695-3126-7.

[3] Dr.R.Geetha Ramani, Suresh Kumar.S , Shomona Gracia Jacob"Rootkit (Malicious Code) Prediction through Data Mining Methods and Techniques" , 978-1-4799-1597-2/13/$31.00 ©2013 IEEE.

[4] Guillermo Suarez-Tangle, "*Evolution, Detection and Analysis of Malware for Smart Devices"* IEEE communications surveys & tutorials, accepted for publication, 2013, pp.1-27.

[5] Johannes Kinder, "*Detecting Malicious Code by Model Checking*" In Second Int. Conf. Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA 2005), Springer, 2005, pp. 174–187.

[6] Kirti Mathur ,Saroj Hiranwal, "*A Survey on Techniques in Detection and Analyzing Malware Executables* "International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 4, April 2013, pp. 422-428.

[7] Matthew G. Schultz "*Data Mining Methods for Detection of New Malicious Executables*", IEEE Symposium on Security and Privacy: S amp; P 2001: proceedings: 14-16 May, 2001, pp. 38-49.

[8] Parisa Bahraminikoo "*Utilization Data Mining to Detect Spyware*", IOSR Journal of Computer Engineering (IOSRJCE), Volume 4, Issue 3, 2012, pp. 01-04.

[9] Robert Moskovitch "*Detecting unknown malicious code by applying classification techniques on OpCode patterns*" Springer-Verlag "http://link.springer.com/article/10.1186%2F2190-8532-1-1", 2012, pp. 1-22.

[10] Yuval Elovici, Asaf Shabtai, Robert Moskovitch, Gil Tahan, and Chanan Glezer" *Applying Machine Learning Techniques for Detection of Malicious Code in Network Traffic".* Proceedings of the 30th annual German conference on Advances in Artificial Intelligence, KI 2007, pp. 10-13.