A Survey on CIMDS: Adapting Post Processing Techniques of Associative Classification for Malware Detection

Atul Kamble, Prasad Kadam, Hardik Bhangale

Abstract—The Malware is program/software that damages or affects the computer system. Nowadays all the fields are computerized. So the valuable data is stored in computer. If the malware attacks on system then there may be chances of loss of data. Therefore it is very essential to provide security to system from Malware. A file that needs to be analyzed is called as Gray list. Along with Malware writing technique the number of gray list is increasing in large scale. In previous work IMDS (Intelligent Malware Detection System) had develop for malware detection. This system is based on analysis of API (Application Programming Interface) calls. But IMDS faces the two problems 1] Handling large set of generated rules to build classifier. 2] Finding the effective rules for classifying new file samples. In this paper we describe post processing techniques that are 1] Rule Pruning 2] Rule Ranking 3] Rule Selection. Then number of classification rule evaluation measures is considered. Here number of selection technique is used to order classification rule contained in classifier. This system is known as CIDCPF for malware detection. According to our knowledge this is first effort that uses post processing technique. It includes chi square, insignificant rule pruning. Then database coverage based upon chi square measure Rule Ranking mechanism is applied. Finally Performance Prediction is done by using Best First Rule.

From the experiment it is observed the promising result is obtained on gray list. As compared to other Anti viruses like McAfee, Virus scan, Norton this system gives best result. This indicates that the CIMDS system is more efficient and accurate for Malware detection. This system is data mining base detection system. In particular CIMDS system can greatly reduce the number of generated rules. This makes it easy for virus analyst to identify the useful ones.

Index Terms—Malware, Association Classification, Antivirus, Rule Pruning, Rule Ranking, Rule Selection.

I. INTRODUCTION

Malware is software designed to infiltrate or damage computer system without the owner's informed consent (e.g., viruses, backdoors, spyware, Trojans, and worms). Numerous attacks made by the malware pose a major security threat to computer users. Hence, malware detection is one of the computer security topics that are of great interest.

Currently, the most important line of defense against malware is antivirus programs, such as Norton, MacAfee, and King soft's Antivirus.

Manuscript Received on December, 2012.

Prasad Kadam, Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Nigdi, Pune-26, India.

Atul Kamble, Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Nigdi, Pune-26, India.

Hardik Bhangale, Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Nigdi, Pune-26, India.

These widely used malware detection software tools use signature-based method to recognize threats. Signature is a short string of bytes, which is unique for each known malware so that future examples of it can be correctly classified with a small error rate. However, this classic signature-based method always fails to detect variants of known malware or previously unknown malware, because the malware writers always adopt techniques like obfuscation to bypass these signatures.

In order to remain effective, it is of paramount importance for the antivirus companies to be able to quickly analyze variants of known malware and previously unknown malware samples. Unfortunately, the number of file samples that need to be analyzed on a daily basis is constantly increasing. Clearly, there is a need for an automatic, efficient, and robust tool to classify the "gray list".

II. RELATED WORK

There are many researches and work that has been done in the field of Malware Detection System.

A. Generation of Malware

First Generation

1) Malware that shares the properties of a virus.
2) Requires human action to trigger replication and Spreading.
3) Propagates via email and file sharing.
4) Examples: Melissa, Love Letter, VBScript worm.

Second Generation

1) Malware that shares the properties of a worm.
2) Does not require human intervention for Replication and spreading.
3) Automatic scanning of victims for vulnerabilities.
4) Hybrid in nature, blended with viruses and Trojans Propagates via Internet.
5) Examples: Slapper worm, SQL Slammer worm and Blaster worm.

Third Generation

1) Pre-compiled vulnerable targets.
2) Exploits known and unknown vulnerabilities.
3) Employs multiple attack vectors.
4) Geographical region- or organization-specific Malware.
5) Attacks security technologies and products.

B. Obfuscation

Obfuscation is to obscure information such that others cannot construe the true meaning. This is certainly true for code obfuscation where the objective is to hide the underlying logic of a program. As the opposite of code optimization where the goal is to minimize the execution time or memory size, code Obfuscation seeks to maximize the 'obscurity' of the code such that others are prevented from gaining knowledge about it.

C. Analysis technique

Static Analysis Technique

It is the process of analysing programmer's code without actually executing it. In this the binary code is translated to the assembler instruction and then data flow and control flow action is taken. It is faster than dynamic analysis but problem with this technique is that many questions arise with program property and program itself.

D. Dynamic Analysis Technique

It analyse the code during run time. In this only those instructions are analysed those are executed. This analysis can be done on virtual machine such as VMWARE

III. PROPOSED SYSTEM

We propose Malware Detection System as shown in Fig. The architecture consists of component like Gray List, Feature Extractor, Signature Database, Classification Associative Rule Generator, Rule Database, Postprocessor and Associative Classifier.

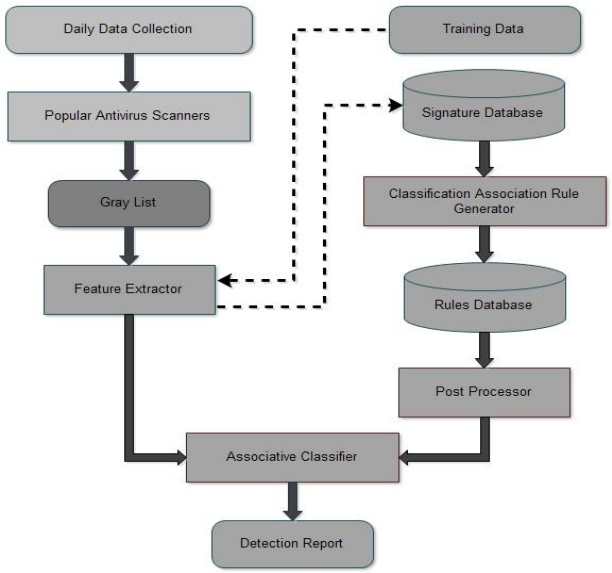

Fig.1. Architecture of Malware Detection.

A. Gray List

Gray list is collection of files on which the analysis is done. It contains both benign files as well as malicious files.

B. Feature Extractor

The system first uses the feature extractor to extract the API calls from the collected portable executable (PE) files, converts them to a group of 32-bit global IDs as the features of the training data, and stores these features in the signature database

C. Classification Associative Rule Generator

Associative classification, as a new classification approach Integrating association rule mining and classification, becomes one of the significant tools for knowledge discovery and data mining. It can be effectively used in malware detection since frequent item sets are typically of statistical significance and classifiers based on frequent pattern analysis are generally effective to test datasets.

After data transformation, it then generates the classification association rules from the training signature database. Rules are generated by using OOA_FP_ FAST_GROWTH algorithm. The rules generated by this algorithm is then store into Rule Database

For malware detection in this paper, the first goal is to find out how a set of API calls supports the specific class objectives: class1 = malicious, and class2 = benign.

1) Support and confidence: Given a dataset DB, let I = {I1, . . , Im} be an item set and I → class (os, oc) be an association rule whose consequent is a class objective. The support and confidence of the rule are defined as follows:
   os = supp(I, class) = count(I ∪ {class}) /DB× 100%
   oc = conf(I, class) = count(I ∪ {class}) /count(I,DB) × 100% where the function count(I ∪ {class}) returns the number of records in the dataset DB where I ∪ {class} holds.
2) Frequent item set: Given mos as a user-specified minimum support. I is a frequent item set/pattern in DB if os ≥ mos.
3) Classification association rule: Given moc as a user specified confidence. Let I = {I1, . . . , Im} be a frequent item set. I → class (os, oc) is a classification association Rule if oc ≥ moc.

D. Post Processor

Here the post processing techniques are applied. The post processing techniques are Rule Pruning, Rule Ranking, Rule Selection. Finally, it builds the classifier using the rules filtered by the postprocessor to detect malware from the "gray list".

IV. POST PROCESSING TECHNIQUES OF ASSOCIATIVE CLASSIFICATION FOR MALWARE DETECTION SYSTEM

A. Rule Pruning Approach

Signature database contains the 32 bit global ID's which is obtained from API calls of files. With the help of signature database Rules are created and stored it in a Rule Database. But problem with Rule Database is that it contains large number of rules which produces the redundancy. In order to remove the redundancy it is necessary to prune the rules. There are five Rule Pruning Approaches:

1) $\chi 2$ (chi-square) testing to measure the significance of the rule itself; 2) redundant rule pruning to discard the specific rules with fewer confidence values; 3) database coverage to just keep the rules covering at least one training data object not considered by a higher ranked rule; 4) pessimistic error estimation to test the estimated error of a new rule; and 5) lazy pruning to discard the rules incorrectly classifying the training objects, we here propose another rule pruning method before building the classifier, named "Insignificant Rules Pruning".

We use $\chi 2$ measure which is based on the comparison of observed frequencies with the corresponding expected frequencies, to test whether the rule is significant with respect to its ancestors. Given two rules generated from the training set T consisting of n data objects R1: A→ r-class (supp = s1, conf = c1) R2: AB →r-class (supp = s2, conf = c2) where A, B are the frequent item sets (A∩ B = φ) of the generated rules and r-class is the class label of T. If these two rules have the same class label, then we call R1 the ancestor

of R2 (or R2 the descendant of R1) If c1 ≥ c2 , namely the confidence of R1 is not greater than its ancestor R2, then R2 is insignificant and can be pruned. If c1 < c2 , we set up the hypothesis H0 that the two patterns A and B are independent. We then compute the observed and expected frequencies of R2. We later use $\chi 2$ measure to test the significance of the deviation from the expected values.

B. Rule Ranking Mechanism

In this mechanism rule ranking is done by using support value and confidence value. Many associative classification algorithms utilize rule ranking procedures as the basis for selecting the classifier during pruning and later for predicting new data objects.

Following are methods for rule ranking.

1) CSA: Based on the well-established "support-confidence" Framework, CSA first sorts the original rule list based on their Confidence values in a descending order. For those rules that share a common confidence value, CSA sorts them in a descending order based on the support values. CSA sorts the rules sharing common values for both confidence and support in an ascending order based on the size of the rule antecedent.
2) ACS: Ensuring that "specific rules have a higher precedence than more general rules" ACS considers the size of the rule antecedent as the most significant factor (using a descending order) followed by the rule confidence and support values, respectively.
3) $\chi 2$ measure: In associative classification algorithms, if the $\chi 2$ measure between two variables (the antecedent and consequent-class of the generated rule) is higher than a certain Threshold value, we can conclude that there might be a relation between the rule antecedent and consequent-class, otherwise, it implies that the two variables may be statistically independent. We can order the list of the generated rules in a descending order based on their $\chi 2$ values.

C. Rule Selection

After building the classifier by the techniques of rule pruning and rule ranking, we can select the subset of the rules from the classifier to predict the new file samples. There are three common rule selection approaches best first rule, all rules, and best k rules. For our malware detection system, we will also try all of these methods to predict the new file samples and find the best way for malware detection.

There are three Methods for Rule Selection approach which are Best First Rule, Best K Rules and All Rules.

1) Best First Rule: Select the first best rule that satisfies the given case according to some ordering imposed on the CAR list. The ordering can be defined according to many different ordering mechanisms, including: (1) CSA – combinations of confidence, support and size of antecedent, with confidence being the most significant factor (used in CBA, TFPC and the early stages of processing of CMAR); (2) ACS – an alternative to CSA that considers the size of the rule antecedent as the most significant factor; (3) WRA – which reflects a number of rule "interestingness" .(4)Laplace Accuracy – as used in PRM and CPAR; Testing – values as used, in part, in CMAR.
2) Best K Rules: Select the first best K rules that satisfy the given case and then select a rule according to some averaging process as used for example, in CPAR. The term "best" in this case is defined according to an imposed ordering of the form described in Best First Rule.
3) All Rules: Collect all rules in the classifier that satisfy the given case and then evaluate this collection to identify a class. One well-known evaluation method in this category is WCS testing as used in CMAR.

V. CONCLUSION

In this paper, we systematically evaluate the effects of the Post Processing Techniques (e.g., rule pruning, rule ranking, and rule selection) of associative classification in malware detection and propose an effective way, i.e., CIDCPF, to detect the malware from the "gray list". To the best of our knowledge, this is the first paper on using post processing techniques of associative classification in malware detection. CIDCPF method achieves better performance on detection ability and efficiency because of its concise, but effective classifier. In addition, our CIMDS system, which adopts CIDCPF method for building classifiers can greatly reduce the number of generated rules and make it easy for our virus analysts to identify the useful ones. Promising experimental results demonstrate that the efficiency and ability of detecting the malware form the "gray list" of our CIMDS system outperform popular antivirus software.

ACKNOWLEDGMENT

We express our great pleasure in submitting this Paper entitled "A Survey on CIMDS: Adapting Post processing Techniques of Associative Classification for Malware Detection". We are indebted to our honorary Principal of our institute Dr. A. M. Fulambarkar who has been a constant source of motivation and co-operating in bringing this project in very short time. We are also thankful to Prof. Dr. J. S. Umale (Head of the Computer Engineering Department) for their involvement and continuous interest in the project. We express our deep sense of gratitude towards Prof. S. R. Kokate for his valuable guidance and his interest; we are able to complete this paper in scheduled time. Lastly we are thankful to all other staff members (Teaching and Non-Teaching) of Computer Engineering Department of Pimpri Chinchwad College of Engineering, Pune and our colleagues who have directly or indirectly helped us while completing this Paper.

REFERENCES

[1] Yanfang Ye, Tao Li, Qingshan Jiang, and Youyu Wang, "Adapting Post Processing Techniques of Associative Classification For Malware Detection", J. Comput. Virol., vol. 4, pp. 323–334, Jan. 2008.
[2] Y. J. Wang, Q. Xin, and F. Coenen, "A novel rule ordering approach in Classification association rule mining," in Proc. 7th IEEE Int. Conf. Data Mining Workshops 2007, pp. 339–348.
[3] U. Bayer, A. Moser, C. Kruegel, and E. Kirda, "Dynamic analysis of Malicious code," J. Comput. Virol., vol. 2, pp. 67–77, May 2006.
[4] A. Sung, J. Xu, P. Chavez, and S. Mukkamala, "Static analyzer of vicious executables (save)," in Proc. 20th Annu. Comput. Security Appl. Conf., 2004, pp. 326–334.