RESEARCH ARTICLE OPEN ACCESS

Security Applications for Malicious Code Detection Using Data Mining

Suvendu Jena, Eshwari Kulkarni, Namrata Annaldas,
Pravin Yalameli, Karishma Shingade
Department of Computer Science and Engineering
Vidya Vikas Pratishthan Institute of Engineering & Technology
V.V.P.I.E.T- Solapur
Maharashtra – India

ABSTRACT

Data mining is the process of posing queries and extracting patterns, often previously unknown from large quantities of data using pattern matching or other reasoning techniques. Data mining has many applications in security including for national security as well as for cyber security. A serious security threat today is malicious executables, especially new, unseen malicious executables often arriving as email attachments. These new malicious executables are created at the rate of thousands every year and pose a serious security threat. Malicious code continues to evolve and create new challenges for organizations seeking to protect themselves. But these challenges are not insurmountable (overcome) and there are a number of practical and effective strategies to reduce the risk

***Keywords:* -** Data Mining, Malicious Codes, RIPPER Algorithms, Decision Tree, Cyber Security.

I. INTRODUCTION

a) *What is Malicious Code?*

Malicious code is a term used to describe any code in any part of a software system or script that is intended to cause undesired effects, security breaches or damage to a system. Malicious code describes a broad category of system security terms that includes attack scripts, viruses, worms, and Trojan horses, backdoor and malicious active content.

b) *Harmful effects of Malicious code on Computer:*

Viruses and worms are some types of malicious code. There are some attack tools like Trojans, spyware and other types of attack tools and some mobile code also have the power to harm the confidentiality, integrity or availability of your computer data or personal information. it can potentially cause more harm in terms of stealing your personal information.

Like other forms of computer network threats, malicious code continues to evolve and create new challenges for organizations seeking to protect themselves. But these challenges are not insurmountable (overcome) and there are a number of practical and effective strategies to reduce the risk.

c) *How Data Mining is useful in detecting malicious code?*

In this system we are using data mining methods, we were planning a system that will automatically design and build a scanner that accurately detects malicious executables before they get a chance to run on the system.

Data mining methods detect patterns in large amounts of data, such as byte code, and use these patterns to detect future objects in similar data in the database. Our framework uses classifiers to detect new malicious executables. A classifier is a set of rules, or detection model which is generated by the data mining algorithm that was trained on a given set of training data.

One of the primary problems faced in today's world by the virus community is to find methods for detecting new malicious programs that have not yet been analyzed before . Many malicious programs are created every day in the world by hackers and most cannot be accurately detected until proper signatures have been generated for them. During this time period, systems protected by signature-based algorithms are vulnerable to attacks on such malicious codes.

We designed a framework that will use data mining algorithms to train multiple classifiers on a set of malicious code and benign (kind) executables to detect new examples for reference to form more frameworks . The binaries in the data are first statically analyzed to extract properties of the binary, and then the classifiers trained over a subset of the data. Our goal in the evaluation of this method is to simulate the task of detecting new malicious executables. To do this we separated our data in the database into two sets which are a training set and a test set with standard cross-validation methodology.

II. PROBLEM STATEMENT

The detection of malicious code is too difficult, because of continuously changing of malware nature and shapes through confusing design techniques. Some characteristics are effective to detect malicious code from huge historical data using classifiers, security and learning algorithm such as RIPPER technology .

III. OBJECTIVE AND SCOPE

The following are the objectives of the project:

1. Ripper algorithm will create proper format to input to Machine Learning classifiers to perform data pre-processing.

2. To develop two program monitored machine learning models; Support artificial neural network.

3. To calculate the functioning of Support vector machine and artificial neural network to classify for new malicious codes.

In this project, machine learning algorithms will be used for classification of dataset as benign or malicious executables. So, the scope of this project will be the following:

1. Consider malicious program that are presents in Microsoft Windows as experiment platform and VMware as virtual machine. Specially, we consider portable executable files because Windows has such a large share of the personal computing market, in addition to malicious writers target the Microsoft Window.

2. The project based on statistical comparisons particular data to determine true value of trained classifiers for that they use machine learning techniques.

3. Four common performance metrics will be used, so evaluate the performance of ensemble machine learning technique are True Positive (TP), False Positive (FP), True Negative (TN), and finally False Negative (FN).

IV. METHODOLOGY

Some of the popular data mining methods are as follows:

1. Decision Trees and Rules
2. Nonlinear Regression and Classification Methods
3. Example-based Methods
4. Probabilistic Graphical Dependency Models
5. Relational Learning Models

These are some famous data mining methods :

1. On-Line Analytical Processing, (OLAP)
2. Classification.
3. Clustering.
4. Association Rule Mining.
5. Temporal Data Mining.
6. Time Series Analysis.
7. Spatial Mining.
8. Web Mining etc.

These methods use different types of algorithms and data. The data source can be data warehouse, database, flat file or text file. The algorithms may be

Statistical Algorithms, Decision Tree based, Nearest Neighbour, Neural Network based, Genetic Algorithms based, Ruled based, Support Vector Machine etc. Generally the data mining algorithms are fully dependent of the two factors these are:

i. Which kind of data sets are using?

ii. What sort of requirements of the user?

Based on the above two factors the data mining algorithms are used and implemented. A knowledge discovery (KD) process includes preliminary processing data, choosing a data-mining algorithm for example RIPPER, and post processing the mining outcome. The Intelligent Discovery Assistants (IDA) assists users to try for valid knowledge discovery processes. The IDA can present users with three benefits:

✓ A methodical enumeration of well-founded knowledge discovery processes.

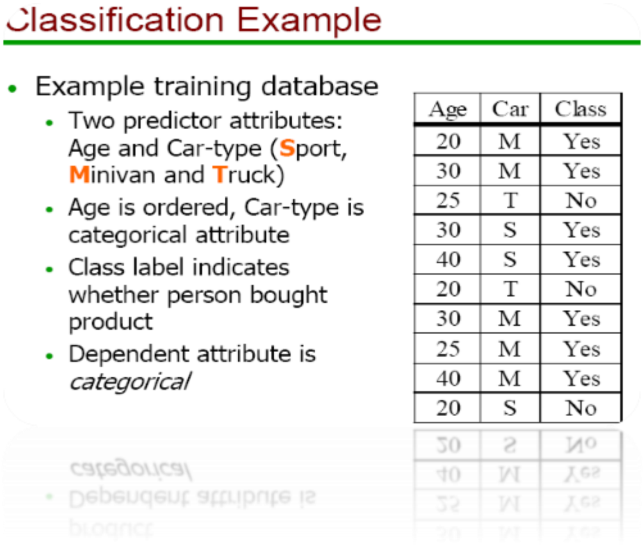
✓ Successful rankings of valid activity by disparate criteria, which guide to pick between the options.

✓ An infrastructure for distribute knowledge, which direct to network externalities.

1.1. *Decision Tree:*

Set of order methodical hierarchically in such a way that the last decision can be decided succeeding the orders that are satisfied from the root of the tree to one of its leaves.

- They build a model made up by rules easy to understand for the user.
- They only work over a single table and only one attribute at a time.
- They are most common used data mining technique.



Classification Example

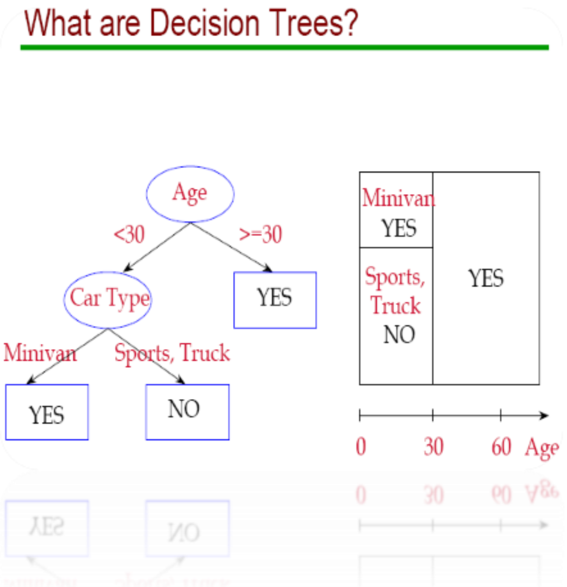
- Example training database
  - Two predictor attributes: Age and Car-type (Sport, Minivan and Truck)
  - Age is ordered, Car-type is categorical attribute
  - Class label indicates whether person bought product
  - Dependent attribute is *categorical*

Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

- The decision trees are built on the strategy "divide and conquer". What discriminate the dissimilar algorithms from each others are the partitions they permit, and what criteria they use to pick the partitions.
- There are two possible category of divisions or partitions:

a. Nominal partitions: a nominal attribute may head to a split with as numerous branches as values there are for the attribute.

b. Numerical partitions: typically, they choose partitions like "X>" and "X <a". Partitions concern with two distinct attributes are not permitted.



What are Decision Trees?

The objective is to generate a model that predicts the value of a target variable based on various input variables. All interior node matches to one of the input variables; there are boundaries to children for each of the probable values of that input variable. Each leaf indicates a value of the target variable given the values of the input variables constitute by the track from the root to the leaf.

A tree can be "learned and understood" by splitting the source set into subsets placed on an attribute value experiment. This procedure is repeated on each obtained subset in a recursive manner called *recursive partitioning*. The recursion is fulfilled when the subset at a node has completely the matching value of the target variable, or when splitting no extended joins value to the predictions.

V. SYSTEM ARCHITECTURE & ALGORITHM

Architecture:

Figure (1) shows the architecture of system detecting malware. The system comprise of three main modules: (1) PE-Header & DLL function , (2) feature selection and transformation, (3) leaning RIPPER algorithm.

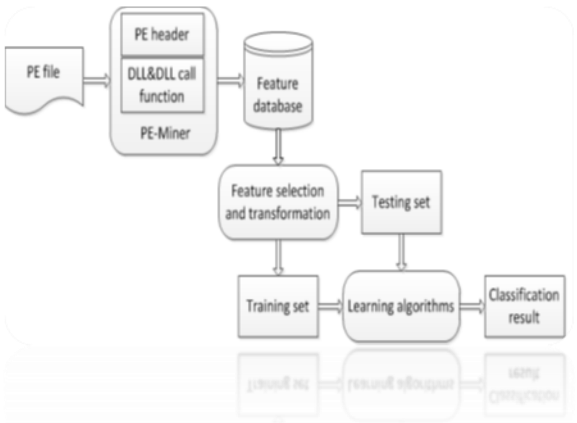


Figure 1 . Architecture of the Malware Detection System

Module 1: PE-Header & DLL function parses the program Executable information in tables of all executables files to find out all Program Executable header, DLL call function , and Data Linked Library functions includes Application Program Interface as raw features.

Module 2: Information Obtained values of each Program Executable header and Application Program Interface function and calling frequency of each Data Link Library are computed with calling frequencies and Information obtained values greater than a portal are selected and main Component Analysis is applied to further alter and also to minimise the number of features.

Module 3: According to the features after PCA, transform every program in the Huge database to its corresponding feature vector and then use a learning RIPPER algorithm to obtain a classified result from these labelled feature vectors.

Algorithm:

We suggest RIPPER algorithm of data mining that is that create new classifiers with distinct features. The data mining technique express for root kit prediction is presented diagrammatically in Figure (2). It includes root kit training data collection, data pre-processing, classification algorithm and performance evaluation phase of classification algorithm .

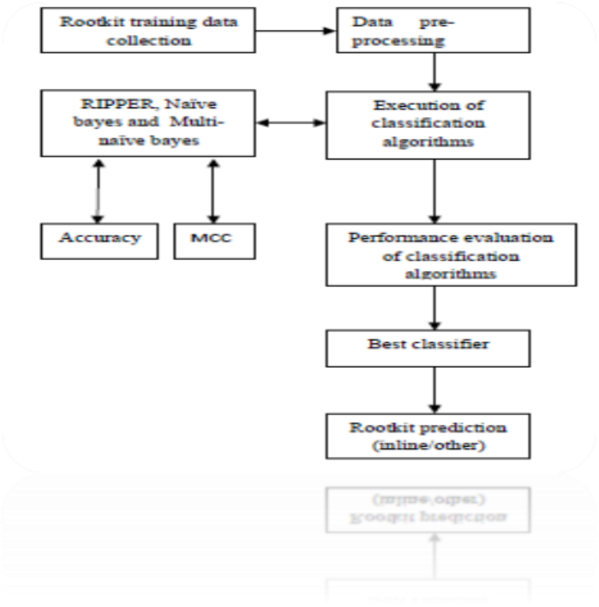

Rootkit training data collection
Data pre-processing
RIPPER, Naïve bayes and Multi-naïve bayes
Execution of classification algorithms
Accuracy
MCC
Performance evaluation of classification algorithms
Best classifier
Rootkit prediction (inline/other)

Figure 2: Data mining methodology for root kit prediction

RIPPER:

RIPPER was developed by William Cohen in the year 1995 based on application of Incremental Reduced Error Pruning (IREP) algorithm. The Repeated Incremental Pruning to Produce Error Reduction that is RIPPER algorithm represents a significant performance improvement over the previous derived algorithms.

The RIPPER algorithm developed a detection model consists of rules that used to detect the examples of malicious executable files. This algorithm is use a LibBFD (Binary Featured Data) as prediction. RIPPER builds a set of rules so it is called a rule-based learning algorithm that determine the classes while minimise the ambiguities.

In RIPPER, a decision rule is defined as a sequence of Boolean clauses linked by logical AND operators that together imply membership in a particular class. The clauses are of the form $A = x$ or $A \neq x$ for nominal attributes and $A \leq y$ or $A \geq y$ for continuous attributes and y is the value for A that occurs in the training set.

To explain the operation of RIPPER, the restricted case is considered in which the examples fall into one of two classes: positive or negative. A high level view of the algorithm is presented in the subsequent paragraphs.

To the basic algorithm, RIPPER adds several rules of optimizing steps as well as the option to improve the rule set by repeating the entire process for a number of times.

Now an overview of the RIPPER algorithm is presented as below:

RIPPER Algorithm

```
LOOP n TIMES
Start with the empty rule (TRUE => positive)
LOOP UNTIL the stopping condition is reached
Partition the training set into a growing set and a
pruning set
Grow a rule by greedily adding a clause to the left
hand side
guided by the grow heuristic
Prune a rule by greedily deleting sequences of final
clauses
Guided by the prune heuristic
Remove examples covered by the rule from the
training set
END LOOP
Perform rule optimization on the entire rule set
END LOOP
```

The algorithm forms rules through a process of repeated building and trimming. During the building phase the rules are made more limiting in order to adjust the training data as close as possible. During the trimming phase, the rules are made less limiting in order to avoid over fitting which can cause poor performance on unseen examples.

RIPPER splits the training examples into a building set and a trimming set. Rules to foretell the positive class are grown one at a time by starting with an empty rule and then adding clauses to the left hand side in a greedy fashion under the guidance of a grow heuristic. Building up a single rule stops when it covers no negative examples from the growing set. Each rule is trimmed immediately after it is grown by deleting clauses that cover too many negative clauses in the trimming set under the guidance of a trim heuristic. After a new rule is build up and trimmed, the covered examples are removed from both the building and trimming set. Then the remaining data is distributed and another rule is buildup. The rules in building process continues until all the examples in the training set are covered or some stopping condition is arrived.

The bold-faced line in the RIPPER algorithm is new, while the rest of the algorithm is an implementation of IREP. During the rule build up phase, the goal is to add clauses greedily into an initially empty rule in such a way that the set of examples covered by the rule contains maximum positive examples and minimum negative examples.

VI. CONCLUSION

Data mining-dependent malicious code detectors have been very successful in detecting malicious code such as viruses and worms. There is a need for a technique in which detection of malicious patterns in executable code sequences can be done more efficiently. Moreover with a larger data set, we can calculate data description method on many types of malicious executables root kit. There is a need to implement the algorithm on the interconnected computers for calculating the performance in terms of time, space and accuracy in real world environments so that we can detect the attacks in larger data sets correctly. It is expected that this procedure will lead to the development of better algorithms for identifying the malicious code that has infected a system and the vulnerable data.

REFERENCES

[1] F. Cohen. "Computer Viruses“. Ph.D thesis, University of California, 1985.

[2] William Stallings. “Cryptography and Network Security Principles and Practices, 4ed, 2005.

[3] Bhavani Thuraisingham, Data Mining for Security Applications, IEEE/IFIP International Conference on Embedded and Ubiquitous Computing,2008 .

[4] 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.

[5] E. Chin, A. P. Felt, V. Sekar, and D. Wagner, “Measuring user confidence in smartphone security and privacy,” in Symp. on Usable Privacy and Security. Washington: Advancing Science, Serving Society, March 2012

[6] M. G. Schultz, E. Eskin, E. Zadok and S. J. Stolfo, “Data Mining Methods for Detection of New Malicious Executables”, Proceedings of the 2001 IEEE Symposium on Security and Privacy, IEEE Computer Society.

[7] MIT Lincoln Labs. 1999 DARPA intrusion detection evaluation.

[8] Tom Mitchell. Machine Learning. McGraw Hill,1997.

AUTHORS

Suvendu jena

Pravin Yalameli

Namarata Annaldas

Karishma Shingade

Eshwari Kulkarni