A Spyware Detection System with a Comparative Study of Spywares using Classification Rule Mining

Satya Narayan Tripathy[1], Sisira Kumar Kapat[2], Susanta Kumar Das[3] and Binayak Panda[4]
1 Department of Computer Science, Berhampur University, snt.cs@buodisha.edu.in
2 Department of Computer Science, Berhampur University, skk.rs.cs@buodisha.edu.in
3 Department of Computer Science, Berhampur University, skd.cs@buodisha.edu.in
4 Department of IT, GIET, Gunupur, Odisha, binayak.panda@gmail.com

ABSTRACT

Malicious software befall a major threat to the security of computer system. The quantity and multiplicity of its variants, render classic security defense in futile and millions of host in the internet are being affected with malware. In this paper, we propose a framework for auto-identification of spyware using data mining techniques. Our framework allows automatic identify classic spyware with similar behavior and assigning unknown spyware to these of discovered classes (classification). We collected, analyzed and processed some thousands of malicious and trustworthy programs for in our experiment to find out the best framework that can classify a given program into a spyware or a trustworthy class. Our research is closed related to information retrieval and classification techniques. We designed a web crawler and generated dataset for our experiment by inputting a spyware URL in order to generate an array of spyware URLs and repeated the process to avail sufficient dataset. We also classified some of the spywares using classification techniques and compared the result. By using this technology, a user can easily find whether a process or a program in the system is involved in spying activity or not. With our experiments, we were able to achieve as high as 97.8% and as low as 95.7% accuracy with a kappa statistics around 0.7 which shows that the statistical significance is much stronger.

Key Words: Spyware, data mining, classification, web crawler.

1. INTRODUCTION

In this current era, web has become part and partial of our life which supports a wide range of platform for the criminal enterprise [9,10]. It allows propagating a vector of malware into the trustworthy systems [11]. Malware is one of the major threats on the internet. In the increasing quantity of multiplicity of malware renders classic security techniques and effects millions of hosts in the internet. Starting from mobile phone to laptop or tablet, software is a major requirement everywhere [12]. Software is written using some computer language. Some of the software coders take advantage of it and misuse their knowledge in creating malicious codes or malwares. ‘Spyware’ is a kind of malware. It can be any software or hardware [14], which gather confidential information about the user and sends back to the controller of the spyware.

1.1. Types of Spyware

Spyware may be categorized based on user prospective such as Domestic Spyware and Commercial Spyware [13]. In other hand this can also be categorized on business prospective as Surveillance Spyware and Advertising Spyware [14]. Some kinds of the examples of spyware are adware, Browser Hijackers, Spybots, and Cookies etc.

Domestic spyware is something which is installed by the user, employer or third parties to monitor the network activity or to collect some personal information (often confidential) of the user. The spyware collects the information of the user and send it to the administrator or the controller of the spyware such as dialers, key logger, spybot etc. Commercial spyware is installed by the companies to monitor the browsing habits of the user. After installation, they use the information for business or marketing purpose. Some of them are Browser Hijack, Profiling Cookies, and Drone Ware etc.

Surveillance Spyware is used for our daily habits or purpose. Often this type of spyware is used by the user, corporations, detectives, intelligence agencies etc. Such Surveillance spyware includes key logger, screen recorder etc. Advertising spyware is used by most of the companies for advertisement. These are otherwise called as adware.

1.2. Symptoms

Once the system get effected by the spyware, it exhibits the symptoms like making the system slow down, slow connection, system may seize to work, targeted pop-ups, targeted email (spam), program customization [4], un-authorized access, unwanted toolbar or search box.

1.3. Sources

Generally the source of spyware is limitless, but the basic major sources of the spyware are drive by download, as a result of clicking some options in a pop-up window, free games, some antivirus also acts as innocent but actually contain a spyware, etc.

1.4. Working Process of Spyware

The working process of spyware is as like it come from any of the sources like free games downloaded, some links provided in some sites, by mail or voluntarily installed by the spyware controller. It resides in the computer memory and then the real game starts. The spyware affects the system resources for which it is created.

The key logger spyware targets the key board as it can capture the information, which key is being pressed. The screen recorder spyware targets the monitor screen as it captures every screen change in the monitor. It captures a small gray-scale image every time the screen is changed and sends to the third party or the controller of the spyware.

2. RELATED WORKS

The user is unaware of the spyware, when it enters into the computer system. So this is a head ache for many of the computer users in a network.

2.1. Signature Based Malware Detection

Signature based malware detection technique uses some specific features or unique strings extracted from binaries [7] of the portable executable (PE) file to analyze and detect the malwares. Obfuscation technique can bypass the signature of a file [5,6]. Malware writers use obfuscation technology such as packing, encryption or polymorphisms to avoid being detected by antivirus tools/engines [13,15]. Hence signature based malware detection technique fails to detect some of the malicious files.

In previous studies Naïve Bayes, Support Vector Machine, Decision tree classifiers were used to detect new malicious executables. Using API calls was quite good but large set of rules generated to be analyzed.

Cumhur [4] stated that, the byte codes for infection is a distinguished feature of virus and these has to be used to detect spyware, but unfortunately this cannot be used as spyware does not use these techniques.

Hence again this was a need to be further analyze the spywares as well as the existing techniques and to develop a prototypical model which can detect all the spyware easily and efficiently.

2.2. Heuristic Search method

Heuristic Technology means “the ability of self-discovery” or “the knowledge and skills that use some methods to determine”, and intelligently analyze codes to detect the unknown virus by some rules while scanning. According to Parisa&et. Al. [7], heuristic search method uses heuristic analysis. In the same way that a human malware analyst would try to determine the process of a given program and its actions, heuristic analysis perform the same intelligent decision making process, effectively acting as a virtual malware researcher. As the human malware analyst learns more from and about emerging threats he/she can apply that knowledge to the heuristic analyzer through programming and improve future detection rates. Parisa& et.al.again said that, the heuristic detection often increases false positive rate.

False positives are when the antivirus software determines a file is malicious (and quarantines or deletes it) but actually that file is just looking like malicious but really they are not. Parisa&et. Al. [7] again said that in heuristic search method we can use Generic Signature. This can locate variations of viruses. Several viruses are renamed or inherited. By using generic signature, these can be classified to same family. E.g., twins can have slightly different fingerprint, but they have identical DNA pattern.

Heuristic based malware detection performs well opposite to known spyware but is not yet proven very successful for detection new spyware [8].

2.2. BF-tree algorithm

Parisa & et.al. [7], used Waikato Environment for Knowledge Analysis (WEKA) to perform the experiment. They first extracted the common features from the finery files. Then they used feature reduction method to reduce data set complexity. Then BF-tree algorithm is used and they got the overall accuracy of 90.5%.

3. METHODOLOGY

This paper uses the simple mechanism to find the spywares. Our framework composed of two threads; one for detection and another thread for classification and analysis. Initially we had designed a web crawler using PHP, whose algorithm is described below. XAMPP server was used to run the web crawler. We had made a list of 2074 spywares and some of their key parameters. The data is collected from http://spywaresignatures.com site (online) by using web crawler. We had analyzed the spyware list and stored them in 'known spyware' database in a structured manner. We also created one another database of 'system processes' collected from operating system. Whenever one new process created or a new program gets executed in the system, our prototype model compares the new process with the two databases. Then the process has to be detected and compared to both of the database. If it matches with at least one of the database then accordingly the result will generate. The result may be, whether the process is spyware, secure or suspicious. If the process does not match with any of the above mentioned two databases, then the process is a suspect and it will be sent for further analysis. The complete process (phases) is depicted in Figure 1 (Proposed System Architecture).

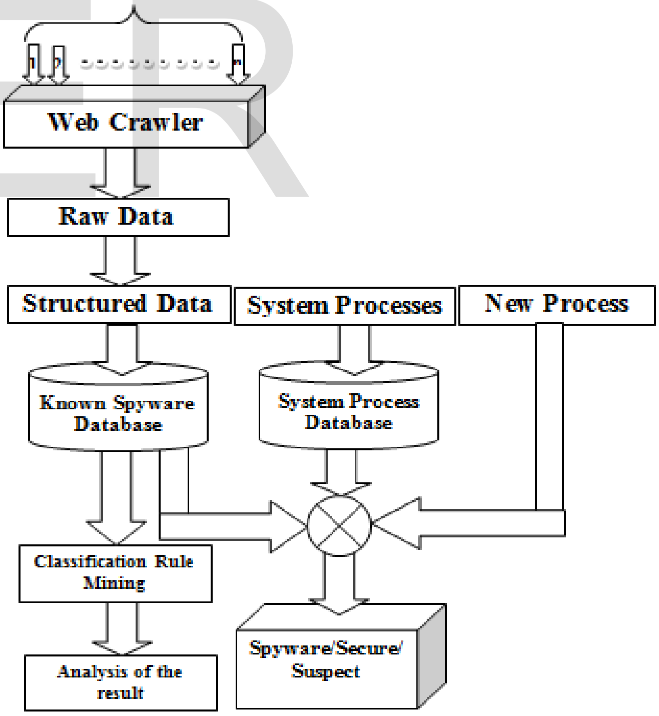


Figure 1 (Proposed System Architecture)

Our framework also classifies existing spyware data using some of the classification rules to analyze the result. We had used five classification rules such as ZeroR, Decision Tree, JRip, J48 and Naïve Bayes algorithms for analysis.

ALGORITHM : Web_Crawler

Input: **URL which inhabit Spyware data**

Output: A set of records including spyware name and its attributes

```
1. Input a URL which inhabit Spyware data
2. Find the child nodes of the current URL and construct an array
3. For i=1 to array_size
4.     Collect spyware name and attributes and store in a file
5. End For
6. Refine all the data in the file to a structured format
7. Store the result in the database
```

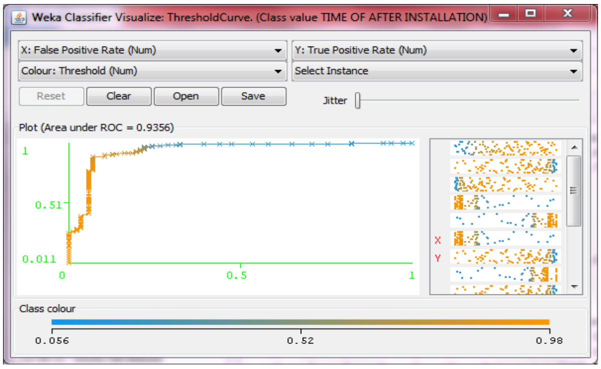
4. ANALYSIS OF THE RESULT

The given below table (Table 1) shows that, among all the algorithms, i.e, ZeroR, Decision Tree, JRip, J48 and Naïve Bayes, the J48 algorithm classified our data set with higher accuracy and with a stronger statistical significant.

Among the ROC area we got for all the algorithms, we found that, the ZeroR rule shows a baseline for all the rules. Again other classifier rules got the similar values (approximately equal). But in comparison, JRip performs well.

From the 2074 data, we have taken dataset of 2032 spyware and analyzed the result by using classification Algorithm with ZeroR, JRip, Decision Table, Naïve Bayes, and J48 and compared the results.

1. *ZeroR:* By using this rule we get '95.7677%' accuracy with a kappa statistics of '0'. Zero kappa statistics shows that there is the lack of any statistical dependence.
2. *JRip:* By using this rule we get '97.687%' accuracy with a kappa statistics of '0.7074'. As compared to ZeroR, the accuracy of the model increased as well as the statistical significance also became stronger.
3. *Decision Table:* By using this rule we get 97.7854% accuracy with a kappa statistics of '0.723'. This means that, as compared to ZeroR and JRip rule, Decision Table has higher accuracy as well as stronger statistical significance. The ROC curve is shown in Figure 2 (ROC curve for Decision tree classification).
4. *Naïve Bayes:* By using this rule we get 96.3583% accuracy with a kappa statistics of '0.379'. This implies that, Naïve Bayes classification is not having stronger significance as compared to JRip and Decision Table rule, but higher than ZeroR.
5. *J48:* By using J48 tree, we get 97.8346% accuracy with a kappa statistics of 0.7307. The ROC curve is shown in Figure 3 (ROC curve for J48 rule)



Weka Classifier Visualize: ThresholdCurve. (Class value TIME OF AFTER INSTALLATION)

X: False Positive Rate (Num)

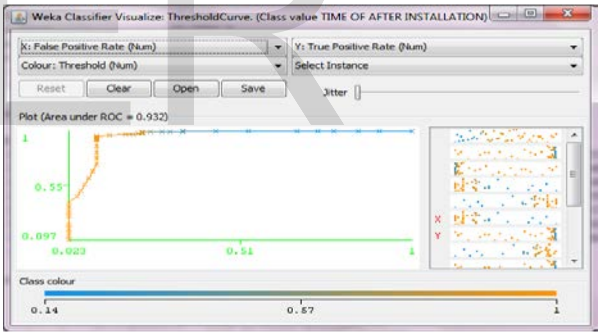
Y: True Positive Rate (Num)

Colour: Threshold (Num)

Select Instance

Plot (Area under ROC = 0.9356)

Figure 2 (ROC curve for Decision tree classification)



Weka Classifier Visualize: ThresholdCurve. (Class value TIME OF AFTER INSTALLATION)

X: False Positive Rate (Num)

Y: True Positive Rate (Num)

Colour: Threshold (Num)

Select Instance

Plot (Area under ROC = 0.932)

Figure 3 (ROC curve for J48 rule)

Overall comparison of the above mentioned all the algorithms are summarized in the following Table 1: (Overall comparison of test results)

Algorithms	Test Accuracy	Kappa Statistic	ROC area
ZeroR	95.7677	0	0.485
Decision Tree	97.7854	0.723	0.9356
JRip	97.687	0.7074	0.971
J48	97.8346	0.7307	0.932
Naïve Bayes	96.3583	0.379	0.9647

Table 1: (Overall comparison of test results)

5. COMPLEXITY ANALYSIS OF THE ARCHITECTURE

5.1. Complexity analysis for initialization

The time required to initialize the architecture completely for a single instance is,

$$T = T_{ES} + T_{SYS} + T_{NEW} + T_{COMP} \qquad (1)$$

Where,
'T' is the total time to run the proposed system architecture for a single instance.
' T_{ES} ' denotes the time taken to build existing/known spyware database

$$T = \sum_{n=1}^{x} T(n) + T_{PRO} \qquad (2)$$

' T(n) ' denotes the time taken to crawl the required data from n'th page.
' T_{PRO} ' is the time taken to process the raw data into structured data.
' T_{SYS} ' denotes the time taken to build system process database and

$$T_{SYS} = \sum_{s=1}^{y} T(s) \qquad (3)$$

' T(s) ' denotes the time taken to find all the system processes
' T_{NEW} ' denotes the time taken to detect a new process, when created in the system.
' T_{COMP} ' is the time taken to compare the new process with both the databases

$$T_{COMP} = T_{KS} + T_{SP} \qquad (4)$$

' T_{KS} ' is the time taken to compare the new process with the known spyware database
' T_{SP} ' is the time taken to compare the new process with the system process database.

Where, $x,y \in N$ and $N \neq \infty$. There must be a limited number of pages in the internet. So value of x is limited. There are limited numbers of processes in the system. So value of y is limited.

Now, combining all the above chunks of the equation, we get the complete equation as,

$$T = \sum_{n=1}^{x} T(n) + T_{PRO} + \sum_{s=1}^{y} T(s) + T_{NEW} + T_{KS} + T_{SP} \qquad (5)$$

5.2. Complexity analysis for regular running

For a single run of the engine to compare one process with both the database is the efficiency of the engine. The performance of the complete architecture is a function of time and space.

Mathematically, this can be represented as,

$$Perf(arch) = f(t,s) \qquad (6)$$

Where,
' Perf(arch) ' denotes the performance of the architecture.
't' denotes the time taken to complete comparison process with time taken to update the database.
's' is the space complexity of the program and the architecture for a single run. Here,

$$t = T_{SYS} + T_{ES} + T_{UP} \qquad (7)$$

Where,
' T_{SYS} ' is the time taken to compare the process with system process database

And ' T_{ES} ' is the time taken to compare the process with existing/known spyware database
' T_{UP} ' is the updating time of both the databases and this is given by,

$$T_{UP} = \sum_{x=1}^{m} T(x) + \sum_{y=1}^{n} T(y) \qquad (8)$$

Where,

T(*x*) denotes the time taken to update the database for *x*'th record of known spyware database and T(*y*) denotes the time taken to update the system process database for *y*'th record.

Whenever a new spyware is found, then it must be added to the database. And also, whenever a new program is loaded into the system or any new service is created in the system, then it is added to the database. Both the values are limited. Hence the value of m and n are limited.

4. CONCLUSION AND FUTURE WORK

Web crawler is the most important part of our research. This research work can be used to find known spywares efficiently, whereas it can be extended to detect new spywares. This work can be extended to develop an efficient rule based framework, which can integrate the signature based detection technique with classification mining.

REFERENCES

[1] “Data Mining Concepts and Techniques”, Jiawei Han and MichelineKamber, , second edition

[2] “Potentially Unwanted Programs Spyware and Adware”, white paper, McAfee® Proven SecurityTM, September 2005, www.mcafee.com

[3] “An Integrated Malware Detection and Classification System”, Ph.D. Thesis, Ronghua Tian, Deakin University, August-2011

[4] “Application of Data Mining based Malicious Code Detection Techniques for Detecting new Spyware”, Cumhur Doruk Bozagac, Bilkent University, Computer Science and Engineering Department, 06532 Ankara, Turkey

[5] “Intelligent Malware Detection System”, Sandeep B. Damodhare, Prof. V.S. Gulhane, International Journal of Advanced Research in IT and Engineering, ISSN:2278-6244

[6] “Prediction and Detection of Malware Using Association Rules”, Mr. B. Dwarakanath, Mr. A. Suthakar, International Journal of Power Control Signal and Computation(IJPCSC), Vol3. No1. Jan-Mar 2012 ISSN: 0976-268X

[7] “Utilization Data Mining to Detect Spyware”, Parisa Bahraminikoo, Mehdi Samieiyeganeh, G.PraveenBabu, IOSR Journal of Computer Engineering (IOSRJCE), ISSN: 2278-0661 Volume 4, Issue 3(Sep.-Oct. 2012), PP 01-04

[8] “Malicious Code Detection through Data Mining Techniques”, Ms. Milan Jain, Ms. Punam Bajaj, International Journal of Computer Science and Engineering Technology (IJCSET)

[9] “ A study on malware taxonomy and malware detection techniques”, Satya Narayan Tripathy, S.K.Das, Brojo Kishore Mishra, Om Prakash Samantray, International Journal of Engineering Research and Technology (ISSN (print): 2278-0181), may 2013, pp 266-273

[10] “A Review on Vulnerability Management in Wireless Communication System”, S. N. Tripathy, T. Mohan Rao; International Journal of Advanced Computer Research (ISSN (print): 2249-7277), March 2013, pp. 71-75

[11] “A Survey on Vulnerability Management for Web Security”, Debendra Dhinda, S. N. Tripathy, Nikunja Kishore Sabat; International Journal of Advanced Computer Research (ISSN (print): 2249-7277), March 2013, pp. 92-94

[12] “Designing A Trust Model for an Ad Hoc Social Network using Communication Data Value”, S. N. Tripathy, S. K. Tripathy, S. K. Das; International Journal of Computer Science & Management systems, 1(2), December 2009, pp. 85-89

[13] http://anti-spyware-review.toptenreviews.com/types-of-spyware.html

[14] Murad M. Ali, Supervised by Dr. Lo’aiTawalbeh, powerpoint presentation on “Spyware”, New York Institute of Technology (NYIT)- Jordan’s Campus, 2006

[15] “Survey on Malware Detection Methods”, Vinod P., V.Laxmi, M.S.Gaur,