Automatic Malware Detection Through Sequential Pattern Mining

Lakshmi Priya A[1], Ms. Vidhya P.M[2]

[1]MTech Cyber Security, Dept. of CSE, SNGCE, Kadayiruppu, Kerala, India.
[2]Asst. Prof. MTech Cyber Security, Dept. of CSE, SNGCE, Kadayiruppu, Kerala, India.

---***---

Abstract - *Nowadays many organizations are a major victim of malware. Malicious software is software that is intentionally included or inserted in a system for a harmful purpose. We all are heard about different types of malwares such as virus worms Trojan etc. Some malwares like virus, worms can be detected using antivirus software. But they can't detect many types of malware such as polymorphic, metamorphic and other unknown malwares. Many detection systems are commonly used signature based detection and the signature byte patterns are derived from known malware, these byte patterns are also commonly known. Hence they can be easily evaded by hackers. These all techniques are time consuming and not effective against the Zero-day attack. So, In this paper we propose an effective method to detect unknown malware It contains an instruction sequence extractor to extract the portable executable (PE) files . Then used a mining algorithm to mined the extracted feature to discover the malicious natured files. Then use a classifier to detect the file is malware or benign and also it reduce the execution time than ANN (All Nearest Neighbor) classifier.*

Keywords: Malware, Instruction Sequence Extractor, Disassembling, Sequential Pattern Mining, RF Classifier,

1. INTRODUCTION

Malware is a short term of malicious software and it is used to damage or destruct our system very easily. Nowadays many of the systems which used internet might be a victim of malware. Many early infectious programs, including the first internet Worm, were written as experiments or pranks. Today, malware is used primarily to steal sensitive personal, financial, or business information for the benefit of others. Malware is sometimes used broadly against government or corporate websites to gather guarded information, or to disrupt their operation in general however, malware is often used against individuals to gain personal information such as social security numbers, bank or credit card numbers, and so on. . So identification of the malware is a serious concern and it is not as much as easy as we think.

Malware is a program that must be triggered or somehow executed before it can infect your computer system and spread to others. Here are some examples on how malware is distributed: a) Social network b) Pirated software c) Removable media d) Emails.

One of the most recognizable signs of malware is a sudden change in how the user's computer is running. The symptoms can include are Poor system performance, Longer start-up times for user's computer, Unexpected closing of browser, or it stops responding, Unresponsive links , or they take you to unrelated pages, Pop-up advertising windows appear when the browser is not open, Additional toolbars are added to the browser.

We can protect our system by these methods like install protection software, practice caution when working with files from unknown or questionable sources, do not open e-mail if you do not recognize the sender, download files only from reputable Internet sites, Install firewall and scan your hard drive for viruses monthly.

Malware detection methods are classified in to different classes based on the type of malware they are targeting and the specific techniques used for the analysis and detection of malware. [2]. Damages that occur in the infected systems are:

- Data Loss: Many viruses and Trojans will attempt to delete files or wipe hard drives when activated, but even if you catch the infection early, you may have to delete infected files.
- Account Theft: Many types of malware include keylogger functions, designed to steal accounts and passwords from their targets. This can give the malware author access to any of the user's online accounts, including email servers from which the hacker can launch new attacks.
- Botnets: Many types of malware also subvert control over the user's computer, turning it into a 'bot' or 'zombie'. Hackers build networks of these commandeered computers, using their combined processing power for tasks like cracking password files or sending out bulk emails.
- Financial Losses: If a hacker gains access to a credit card or bank account via a keylogger, he can then use that information to run up charges or drain the account. Given the popularity of online banking and bill payment services, a hacker who manages to secrete a keylogger on a user's system for a full month may gain access to the user's entire financial portfolio, allowing him to do as much damage as possible in a single attack.

In recent years, developers of anti-malware solutions need to develop counter mechanisms for detecting and deactivating them, playing a cat-and-mouse game. The huge number of malware families, and malware variants inside the families, causes a major problem for antimalware products. For example, McAfee Lab's antimalware solutions reported more than 350M total unique malware sample in 2014, that represents a growth of 17 percentages with respect to the analogous data in Symantec [5] reported more than 44.5 million new pieces of malware created in May 2015 [6].

In the proposed system we can find unknown malwares with the help of mining algorithm and reduce the execution time with the use of Random Forest (RF) classifier

The remainder of this paper is organized as follows: Section 2 introduces the Background. In Section 3, describes about the proposed system. Section 4 shows the analysis result and in section 5 gives the conclusion.

2. BACKGROUND

In past three decades almost everything has changed in the field of malware and malware analysis. From malware created as proof of some security concept and malware created for financial gain to malware created to sabotage infrastructure. There are mainly three techniques for malware detection [4]: Signature based, Heuristic based and Specification based techniques. To overcome the limitations of signature-based detection some malware researchers apply Control Flow Graph, machine learning techniques and data mining techniques

2.1 Signature Based Detection

It maintains the database of signature and detects malware by comparing pattern against the database. It shall require some amounts of system resources to detect the malware also this technique [4] can detect the known malware accurately. The disadvantage of this technique is it not effective against the Zero-day attack so it cannot detect the new, unknown malware as no signature available for such type of malware.

2.2 Heuristic Based Detection

It is also called as anomaly based detection[4]. Here mainly the goal is to analyze the behavior of known or unknown malwares. Behavioral parameters include various factors such as source/ destination address of malwares, different types of attachments and other measurable statistical features.

A key advantage of anomaly based detection is its ability to detect zero-day attacks. Zero-day attacks are attacks that previously unknown to the malware detector.

2.3 Specification Based Detection

Specification-based detection [4] is a derivative of anomaly based detection that tries to defeat the typical high false alarm rate associated with the anomaly-based detection. Specification-based detection relies on program specifications that describe the intended behavior of security-critical programs. It monitors executions program involve and detecting deviation of their behavior from the specification, rather than detecting the occurrence of specific attack patterns. This technique is similar to anomaly detection where they detect the attacks as deviate from normal. The difference is that instead of relying on machine learning techniques, it will be based on manually developed specifications that capture legitimate system behavior. It can be used to monitor network components or network services that are relevant to security, Domain Name Service, Network File Sharing and routers.

3. PROPOSED SYSTEM

The proposed system presents an effective method to detect the malware in your system .Here , detect the malware very fast than the existing mechanism. There are three stages to detect the malware .In the first stage we used an instruction sequence extractor to extract the sequences with the help of disassembling and parsing. In the second stage a sequential pattern mining algorithm is used to predict that the mined sequences have a chance to being malware. in the last and third stage a random forest algorithm is used to detect that the taken .exe file is benign or not. Using RF method we can reduce the execution time than the earlier method such as ANN (All nearest neighbor)[1] classification algorithm.

3.1 System Architecture

The Fig - 1 shows the system architecture of the proposed method for detecting whether the given files are malware or not.

First we take some malicious samples and some benign samples as input. A dataset which contains malicious sequences are also entered as input. Then an instruction sequence extractor is used to extract the feature of the given samples using disassembling and parsing technique and then these data are taken as the input of the pattern miner and it outputs the malicious sequences and using the RF classifier classifies the taken samples are malicious or benign samples.

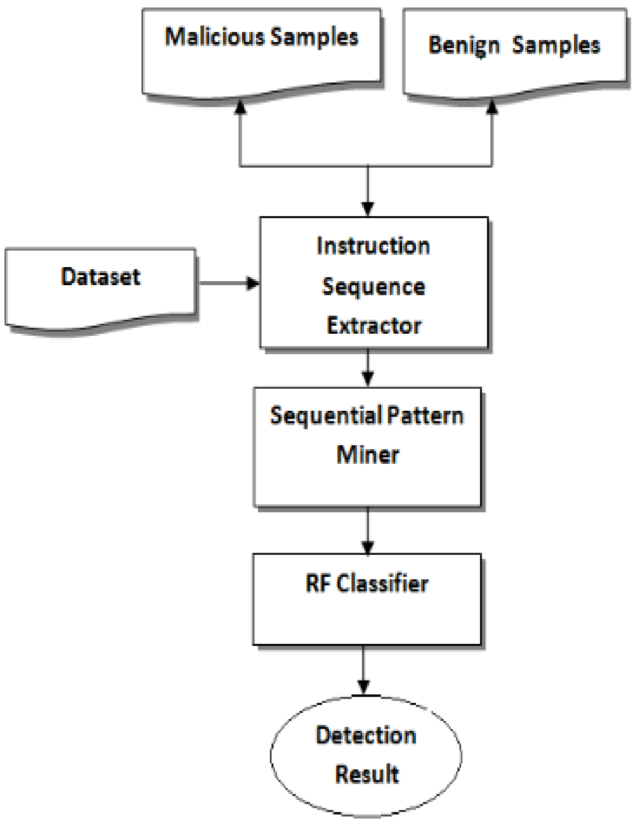
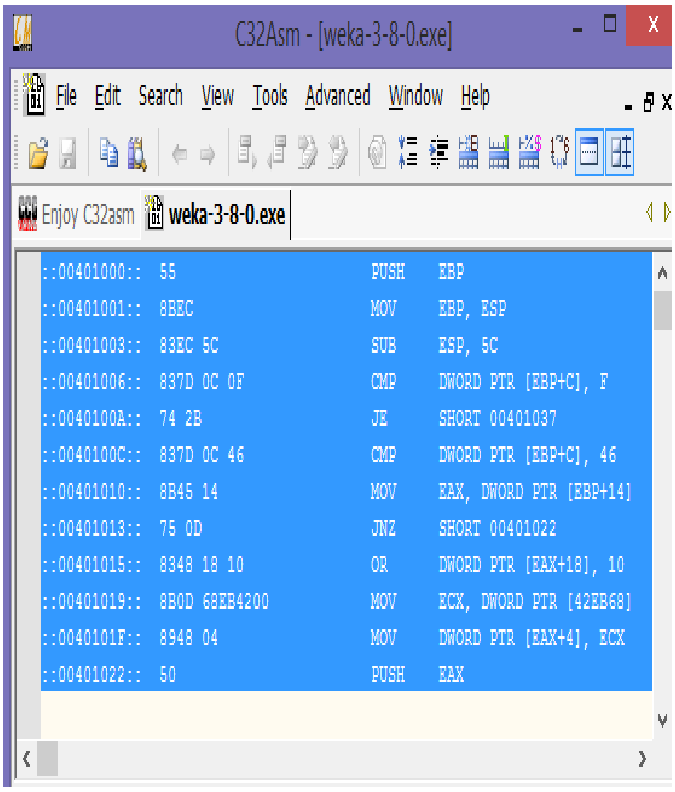


Fig-1: System Architecture

3.2 Instruction Sequence Extractor.

It is very important to extract the feature of the testing samples. Here the input file is PE file and it might be an encrypted format so to decrypt the .exe file, we use a disassembler called C32Asm [3]. For malware files , we use online disassembler to disassemble the file. The assembly code format for both disassembler are not same. There are different free disassemblers are available in the internet. Here we are used c32asm disassembler to disassemble the benign files.

Fig. 2: shows the sample .exe file 'weka-3-8-0.exe' is disassembled and it shows the assembly code of the sample file.



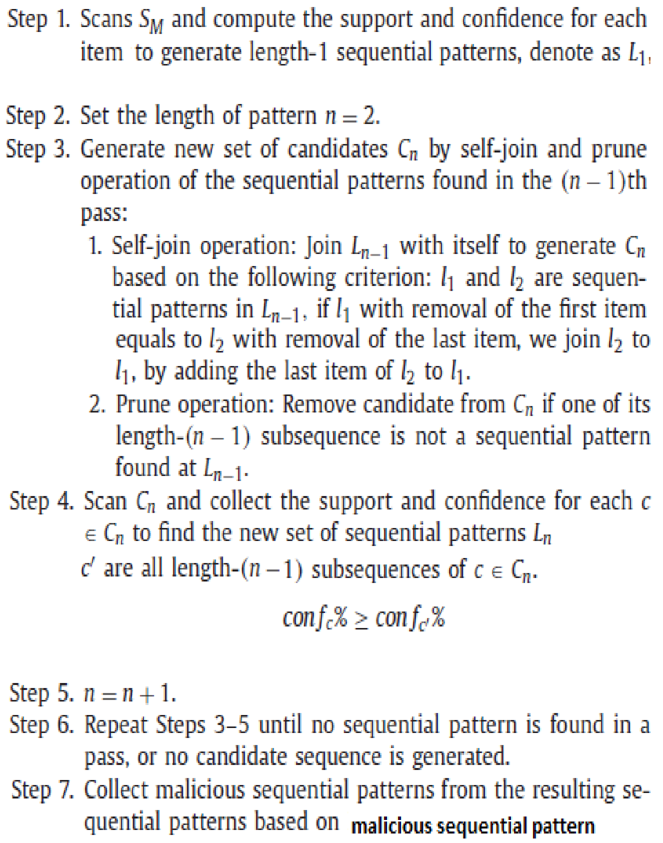
```
::00401000::  55                   PUSH   EBP
::00401001::  8BEC                 MOV    EBP, ESP
::00401003::  83EC 5C              SUB    ESP, 5C
::00401006::  837D 0C 0F           CMP    DWORD PTR [EBP+C], F
::0040100A::  74 2B                JE     SHORT 00401037
::0040100C::  837D 0C 46           CMP    DWORD PTR [EBP+C], 46
::00401010::  8B45 14              MOV    EAX, DWORD PTR [EBP+14]
::00401013::  75 0D                JNZ    SHORT 00401022
::00401015::  8348 18 10           OR     DWORD PTR [EAX+18], 10
::00401019::  8B0D 68EB4200        MOV    ECX, DWORD PTR [42EB68]
::0040101F::  8948 04              MOV    DWORD PTR [EAX+4], ECX
::00401022::  50                   PUSH   EAX
```

Fig 2: C32asm disassemble disassembling weka-3-8-0.exe file

After disassembling we take those assembly codes and extract the features by removing operands from the code and transform each operator into a unique code. Like MOV – 120, SUB – 108 etc.. Then calculate the tendency [1] value to measure the minimum chance of instruction to be malicious. An instruction i is selected only if tendency (i) > t, where t is a user specified threshold.

3.3 Sequential Pattern Mining

This is a method used to mine the extracted data to discover those files which contain malicious nature. . For mining we use an algorithm called malicious sequential pattern mining algorithm [1] .Fig – 3: [1] shows malicious sequential pattern mining algorithm.



Step 1. Scans S_M and compute the support and confidence for each item to generate length-1 sequential patterns, denote as L_1,

Step 2. Set the length of pattern $n = 2$.

Step 3. Generate new set of candidates C_n by self-join and prune operation of the sequential patterns found in the $(n-1)$th pass:

1. Self-join operation: Join L_{n-1} with itself to generate C_n based on the following criterion: l_1 and l_2 are sequential patterns in L_{n-1}, if l_1 with removal of the first item equals to l_2 with removal of the last item, we join l_2 to l_1, by adding the last item of l_2 to l_1.
2. Prune operation: Remove candidate from C_n if one of its length-$(n-1)$ subsequence is not a sequential pattern found at L_{n-1}.

Step 4. Scan C_n and collect the support and confidence for each $c \in C_n$ to find the new set of sequential patterns L_n c' are all length-$(n-1)$ subsequences of $c \in C_n$.

$$conf_c\% \geq conf_{c'}\%$$

Step 5. $n = n + 1$.

Step 6. Repeat Steps 3–5 until no sequential pattern is found in a pass, or no candidate sequence is generated.

Step 7. Collect malicious sequential patterns from the resulting sequential patterns based on **malicious sequential pattern**

Fig-3: Malicious Sequential Pattern Mining (MSPE) Algorithm.

3.4 Random Forest Classifier

Random forest [7] (or random forests) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees.

Decision trees are individual learners that are combined. They are one of the most popular learning methods commonly used for data exploration. One type of decision tree is called CART. Classification and regression tree. [8] The advantages of random forest are:

- It is one a highly accurate classifier.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data
- Generated forests can be saved for future use on other data.
- Prototypes are computed that give information about the relation between the variables and the classification.
- RF is fast to build. Even faster to predict
- Automatic predictor selection from large number of candidates

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them

3.4.1 RF Classifier Algorithm

Each tree is constructed using the following algorithm:[]

Step 1. Let the number of training cases be *N*, and the number of variables in the classifier be *M*.

Step 2. We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.

Step 3. Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.

Step 4. For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.

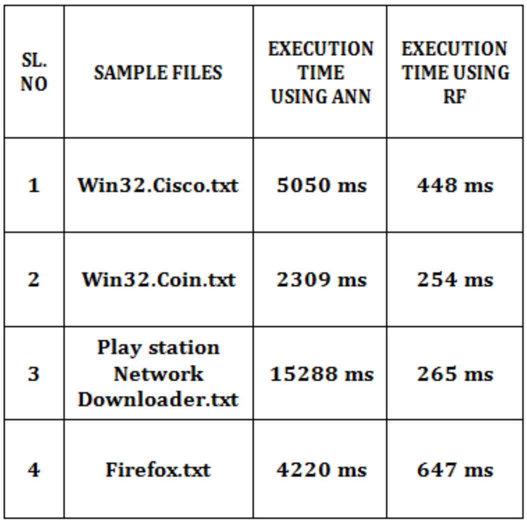
Step 5 Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction.

4. SECURITY ANALYSIS

The system ensures the faster way to detect a malware in your system. The good performance achieved by malicious sequential patterns owes to their strong ability to represent malicious executables. The malicious sequential patterns are generated by MSPE algorithm which integrates the concept of objective-oriented. In our case, the objective is to detect malware, thus the MSPE algorithm is tend to find patterns to support this specific objective. Different from other instruction features used in the experiment above, these discriminative patterns capture the notable difference between malware and benign executables and are essential for malware detection.

The execution time for the existing system (which used ANN classifier) is very larger than the proposed system (used RF classifier). Fig – 4: shows the Comparison of the execution time for different malware samples and benign samples using ANN and RF classifiers.



SL. NO	SAMPLE FILES	EXECUTION TIME USING ANN	EXECUTION TIME USING RF
1	Win32.Cisco.txt	5050 ms	448 ms
2	Win32.Coin.txt	2309 ms	254 ms
3	Play station Network Downloader.txt	15288 ms	265 ms
4	Firefox.txt	4220 ms	647 ms

Fig-4: Comparison of Execution time (using ANN and RF)

From the Fig 4 we can easily identify the difference between the execution time for the four (1 and 2 are malicious samples and 3 and 4 are benign samples) samples while using both classifier.

5. CONCLUSION AND FUTURE WORK

In this paper, we used an Instruction sequence extractor, a data mining based detection frame work called malicious sequential pattern mining and Random Forest classifier. With the help of Instruction sequence we extract the feature of the sample .exe file and with the feature extracted value we mined the samples using MSPE algorithm to discover the malicious natured files .Random Forest is a basic classifier which is used to identify the mined samples are malware or not . Using the RF classifier we can reduce the execution time much lesser than ANN classifier.

In this paper we only detect the malware in the faster way. In future we can use some other classification mechanism to identify the detected malware is in which type and log the details for future needs.

REFERENCES

1) Yujie Fan, Yan fang Ye, Lifei Chen ,"Malicious sequential pattern mining for automatic malware detection". Journal in Expert Systems with Applications, pp 16 – 25, 2016.

2) 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

3) C32Asm (2011). https:// tuts4you.com/ download.php?view.1130.Accessed22.06.14.

4) Ms. Shital Balkrishna Kuber." A Survey on Data Mining Methods for Malware Detection"
International Journal of Engineering Research and General Science Volume 2, Issue 6, October-November, 2014

5) Mcafee labs threats report, "http://www.mcafee.com/us/resources/reports/rp – quarterly – threat - q4-2014.pdf", Feb. 2015..

6) Symantec (2015).Symantec intelligent report "http://www.symantec.com/content/en/us/enterprise/pdf" October 2015

7) Jehad Ali , Rehanullah Khan , Nasir Ahmad , Imran Maqsood "Random Forests and Decision Trees" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012

8) Andy Liaw and Matthew Wiener " Classification and regression by random forest" Vol. 2/3, December 2002