Volume 2, No. 6, June 2011

Journal of Global Research in Computer Science

ISSN-2229-371X

RESEARCH PAPER

Available Online at www.jgrcs.info

FEATURE DETECTION APPROACH FROM VIRUSES THROUGH MINING

Raviraj Choudhary[#1], Ravi Saharan[*2]

[#]Department of Computer Science & Engineering, Techno India NJR Institute of Technology, Udaipur
[1]ravisuncity@gmail.com
[*]Department of Computer Science & Engineering, Central University of Rajasthan, Kishangarh, Ajmer
[2]ravijee82@gmail.com

***Abstract*—**Anti-virus systems traditionally use signatures to detect malicious executables, but signatures are over fitted features that are of little use in machine learning. Other methods seek to utilize more general features, with some degree of success. Through this paper we present a new approach that conducts an exhaustive feature search on a set of computer viruses. This method detects mnemonics patterns in large amounts of data, and uses these patterns to detect future instances in similar data. We use apriori algorithm for select features to detect malicious executables. Through those features we make a rule set or detection model for trained over a given set of training data.

Keywords—Antivirus, mnemonics, apriori algorithm, malicious executables

INTRODUCTION

There are many approaches used for detecting malicious program. But every year thousands of new viruses are found for that traditional approaches are not sufficient to detect those files. To address this problem, we explore solutions based on machine learning and not strictly dependent on certain viruses. The term virus is commonly used for malicious code, but for clarity reasons, we will use the term malicious code in further discussion, since it is relevant for all kinds of malicious code, such as viruses, worms, and Trojan horses.

Malicious software is becoming a major threat to the computer world. The general availability of the malicious software programming skill and malicious code authoring tools makes it easier to build new malicious codes. Recent statistics from Windows Malicious Software Removal Tool (MSRT) by Microsoft shows that about 0.46% of computers are infected by one or more malicious codes and this number is keep increasing [1]. Moreover, the advent of more sophisticated virus writing techniques such as polymorphism [2] and metamorphism [3] makes it even harder to detect a virus. The data-mining framework automatically found patterns in our data set and used these patterns to detect a set of new malicious binaries [4].

Our aim is to develop a more systematic and efficient approach in building virus detection model. In first section Method we present whole model for select top L feature from malicious data set. We generate a data set of malicious programs and disassemble all files.

FEATURE DETECTION APPROACH FROM VIRUSES THROUGH MINING

Method

In this paper we present a virus detection approach through data mining. For that we used some virus files from corpus data set and some viruses generate from vcl32 virus kit.

```
loc_40209: xor cmp jnz mov cmp jz

loc_40227: dec jmp
```

Abstract Assembly

Example of abstract assembly

Major Steps

a. Make virus data sets.
b. Disassemble virus files using any disassembler.
c. Generate abstract assembly opcode.
d. Feature selection algorithm.

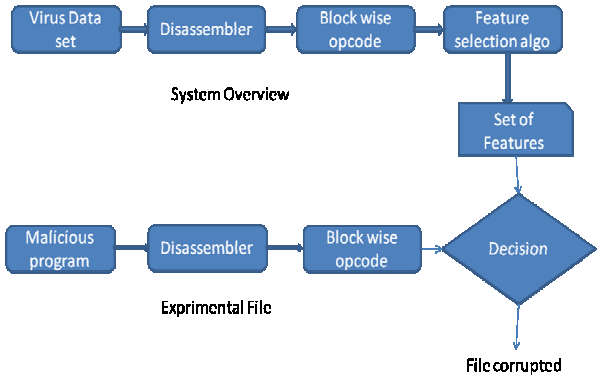

Figure. 1 Flow Diagram

FEATURE SELECTION

The features for our classifier are instruction associations. To select appropriate instruction associations, we use the following two criteria:

A) The instruction associations should be frequent in the training data set. If it occurs very rarely, we would rather consider this instruction association is a noise and not use it as our features.
B) The instruction associations should be an indicator of malicious code.

To satisfy the criteria, we only extract frequent instruction associations from training dataset. Only frequent instruction associations can be considered as our features. We use a variation of Apriori algorithm to generate all three types of frequent instruction associations from abstract assembly. One parameter of Apriori algorithm is" minimum support"[5]. It is the minimal frequency of frequent associations among all data. More specifically, it is the minimum percentage of basic blocks that contains the instruction sequences in our case. Normalized count is the frequency of that instruction sequence divided by the total number of basic blocks in abstract assembly. We can also use N gram approach to find feature set from that data [6].

Then select top L features as our feature set. For one executable in training dataset, we count the number of basic blocks containing the feature, normalized by the number of basic blocks of that executable. We process every executable in our training dataset, and eventually we generate the input for our classifier as like Naive Bayes, Ripper [8].

Algorithm

Find the frequent itemsets: the sets of items that have minimum support
–A subset of a frequent itemset must also be a frequent itemset
•i.e., if {*AB*} is a frequent item set, both {A} and {B} should be a frequent itemset.
•Use the frequent itemsets to generate association rules.
C_k: Candidate itemset of size k
L_k : frequent itemset of size k
L_1 = {frequent items};

for(k= 1; L_k !=∅; k++) do begin

C_{k+1}= candidates generated from L_k
for each transaction t in database do
increment the count of all candidates in C_{k+1} that are contained in t
L_{k+1}= candidates in C_{k+1} with min_support
end
end
return ∪ L_k;

EXPERIMENTAL SETUP & RESULT

Virus data set:
(i) 1500 files from corpus data set [7]
(ii) 500 files from vcl32 generator
IDA Pro: Disassembler to generate ASM file from malicious files
Virus Code: ASM file of any virus file
Opcode selector: select opcode from asm files and make logic assembly and abstract assembly.
Abstract assembly: Opcode of all virus file as per basic blocks.

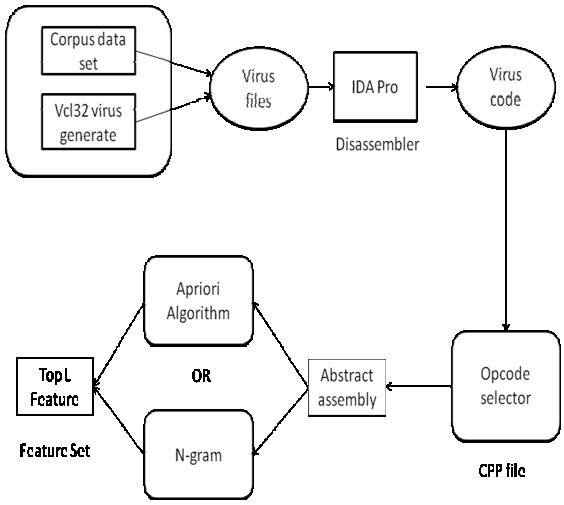

Figure.2 Flow Diagram

In above fig virus files are generated from VCL32 and corpus data set. Through Idpro disassembler we generate instruction code of those files. We present whole model for select top L feature from malicious data set. We generate a data set of malicious programs and disassemble all files. Then we use opcode selector for refine virus code and generate abstract assembly.

CONCLUSION

We have implemented apriori algorithm. Using apriori algorithm we can select top L features. These L features can be used to recognize weather a file is virus or not. We can use any classification model or neural network approach to get decision.

REFERENCES

[1] Microsoft Antimalware Team, "Microsoft Security Intelligence Report (January - June 2007)," 2007.
[2] C.Nachenberg,"Computer virus-antivirus coevolution," Communications of the ACM, vol. 40, no. 1.
[3] P. Szor and P. Ferrie, "Hunting for metamorphic," in 11th International Virus Bulletin Conference, 2001.
[4] Data Mining Methods for Detection of New Malicious Proceedings of the 2001 IEEE Symposium on Security and Privacy Page: 38 Year of Publication: 2001 ISSN:1081-6011
[5] Efficient Virus Detection Using Dynamic Instruction Sequences Jianyong Dai, Ratan Guha, Joohan Lee JOURNAL OF COMPUTERS, VOL. 4, NO. 5, MAY 2009. University of Central Florida.
[6] A Feature Selection and Evaluation Scheme for Computer Virus. This paper appears in: Data Mining, 2006. ICDM '06. Sixth International Conference on Publication Date: 18-22 Dec. 2006 on page(s): 891-895, ISSN: 1550-4786, ISBN: 0-7695-2701-7 INSPEC Accession Number: 10222296 Digital ObjectI dentifier: 10.1109/ ICDM. 2006.4 Current Version Published: 2007-01-08.
[7] Vx heavens. http://vx.netlux.org/lib.
[8] A Data Mining Framework for Building Intrusion Detection Models. Wenke Lee; Stolfo, S.J.; Mok, K.W.; Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on 9-12 May 1999 Page(s):120-132 Digital Object Identifier 10.1109/SECPRI.1999.766909