A Research on the Heuristic Signature Virus Detection Based on the PE Structure

Di Gao
Harbin Engineering University
College of Computer Science and Technology
Harbin, China
e-mail: gaodi@hrbeu.edu.cn

Guisheng Yin
Harbin Engineering University
College of Computer Science and Technology
Harbin, China
e-mail: yinguisheng@hrbeu.edu.cn

Yuxin Dong
Harbin Engineering University
College of Computer Science and Technology
Harbin, China
e-mail: dongyuxin@hrbeu.edu.cn

Liang Kou
Harbin Engineering University
College of Computer Science and Technology
Harbin, China
e-mail: kouiang@hrbeu.edu.cn

***Abstract*—With the development of network technology, computer networks are becoming increasingly popular in people's daily life. Computer brings us not only convenience but also potential problems caused by computer viruses. Most viruses are Win32 PE viruses. This paper firstly analyzes the Win32 PE file structure, then analyzes the virus's Principles of infection in detail and finds the PE virus Heuristic feature vector and stores Heuristic feature vector into a database. It reduces the redundant feature items with the feature extraction method of minimizing discriminate entropy. Finally the improved KNN algorithm is used to classify. The experiment results show that the method has a high hit rate and lower missing rate.**

Keywords-PE virus; heuristic signatures; Win32 PE file structure

I. INTRODUCTION

With the popularity of the Internet and the infiltration of computer network technology in various fields, computer network security issues are becoming the focus of people's attention. Network security is facing unprecedented threats and challenges, and the culprit is the computer virus. On Windows systems Viruses, file viruses are skilled, in large quantities, destructive and hardest preventive. They generally infect executable files by modifying the structure to get control of the system, to run malicious code and endanger the operating system. Executable files are generally PE[1] (Portable Execute) file format in Windows NT platform therefore the study of detection of PE file becomes very meaningful to PE file virus detection[2].

Computer viruses are increasingly rampant which promotes the rapid development of virus detection technology. Now introduce the mainstream virus detection technology. Features scanning technology[3] is the most effective and accurate method to detect known viruses. Virtual machine technology, Active Defense Technology and Behavior analysis technology are relatively advanced technologies against variants of the virus[4-6]. Features scanning technology's detection capability for unknown Trojan is weak. Virtual technology and active defense technology relatively consume system resources and have poor stability. This paper proposes a heuristic signature extraction algorithm based on PE File Structure by comprehensive advantages and disadvantages of various common signature technologies. Specific ideas are as follows:

- Analyze how the virus infected Win32 PE file structure through deep study of virus infection strategic.
- Attempt to Set PE structure instructive signs through virus infection strategy and analysis of a large number of viruses.
- Seek its PE structure instructive signs one by one when scanning a virus sample. If the inspiring flag was found, the corresponding signature sequences of the flag ware extracted.
- Select the most representative characteristics of the virus signature sequences to carry out features sequence combinations and form signature.
- Verify the robustness of inspiring symbols by experiment, modify and continue to find new inspiration signs.

II. HEURISTIC SCANNING TECHNOLOGY

Heuristic refers to "the ability of self-discovery" or "the knowledge and skill that use some way or mothed to judge thing". In fact, heuristic scanning technology virus detection software is the PE file structure analysis and scanner that is achieved in a particular way. This technology discovers structure differences between virus samples and normal file PE file in order to find these easily modified PE structures by virus and set them as instructive signs. It also acquires some statistics, static inspired knowledge to form a static heuristic analysis technology.

Virus detection software used heuristic scanning technology is able to find suspicious code's instructive signs and sort according to these signs' harm to a computer system

and give these signs different weighted values based on characteristics used by virus. Virus detection program can claimed find virus if weighted sum of each inspired signs has been more than one pre-defined threshold value. Heuristic scanning technology has a certain false alarm rate and it is probabilistic method[7].

Heuristic scanning technology can compensate for the disadvantage that signature scanning lags behind virus update and discover new viruses or new variants of the virus rapidly.

III. STRATEGIC ANALYSIS OF VIRAL INFECTION FROM PE FILE STRUCTURE

A. PE file structure introduction

For the PE file, the most important point needed to know is that the structure of this executable code on disk is very similar to that when it is loaded into memory and ready to perform. It makes the work that system loader needs to do become easy. PE applications no longer need to perform modifications for library call. PE file system loader uses the special area, the Import Address Table to complete the function.

The code, data, resources, import and export tables of Win32 executable program is a contiguous linear memory space in memory. Executable files only need to know the position in memory that they are mapped by the loader program. It is easy to find the various parts of file based on stored pointer in the File impression.

Another concept that should be familiar is the relative virtual address. Many fields in PE files are specified by the RVA. RVA indicates offset between this domain and memory image file starting position.

Before telling about important details about PE file further, referring to Figure 1, it shows the overall structure of PE files.

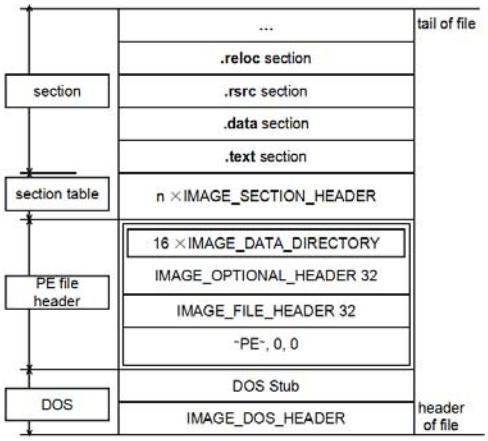

Figure 1. PE file structure diagram.

Brief introduction of PE file structure: PE file start with MS-DOS executable body. PE header is in the rear of the DOS header, it has main header information and some optional headers information. Immediately after the head of the PE is section table that contains all sections' name, position, length and attribute. Section table also contains the type information of the data sections or code sections. Code only has one type, but data has many types such as API import table and export table, resource and relocation information and the data that program reads and writes.

In order to deepen the understanding of the PE file, author uses the PEditor software to open a PE file, shown as Figure 2.

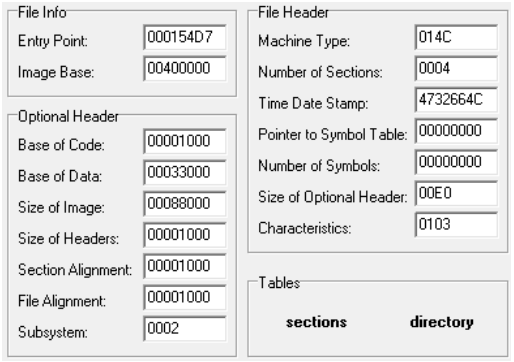

Figure 2. view server.exe information.

Section	Virtual Size	Virtual Offset	Raw Size	Raw Offset	Characteristics
.text	000313C2	00001000	00032000	00001000	60000020
.rdata	00008C8C	00033000	00009000	00033000	40000040
.data	00016930	0003C000	00002000	0003C000	C0000040
.rsrc	00034B68	00053000	00035000	0003E000	40000040

Figure 3. server.exe PE Section Information.

B. PE type virus Heuristic

After full understanding of the PE file structure, we analyze PE file virus heuristic flag as follow.

1) Code starts executing from the last section: most Win32 viruses apply some common infection techniques for file infection and modify the application's entry point to the last section of the program instead of Text section. If the entry point of PE image does not point to the code section, we can choose the content pointed to as a part of signature.

2) Suspicious properties of head section: code section has a "executable" sign, but does not need to have "writable" sign, because the code is separate from the data. Very often, the section that virus exits does not have "executable" sign but has "writable" sign, or it has "executable" sign and "writable" sign at the same time.

3) Suspicious code redirection: place a jump instruction at the entry point to lead the program to other section. If code execution flow is found to jump from the main code section the other section close to the location of the program entry point, then extract the code after jump as a signature.

4) Suspicious code section name: if one section does not normally contain code obtains control of the program,

corresponding signature based on similar information to match with the above behavior.

5) Possible infection of the head: if the entry point of PE does not point to any section but it points the area between PE header and the original data of first section then the PE file may be infected with a head virus. Position the code of head infection virus based on above information and extract some as a signature.

6) Suspicious import entries based on the serial number from KERNEL32.DLL: Some Win32 viruses modify import table of the infected program and add entries based on serial number. If there is importing entries based on serial number from KERNEL32.DLL, it can be suspicious. If GetProcAdress function or GetModuleHandle function uses the serial number to import, it should extract feature sequence here.

7) Import Address Table is modified: if application's import table of contains GetProcAdress and GetModuleHandle API import entries and uses the serial number to import them then the import table is certainly been modified. The corresponding signature sequences should be extracted at this point.

8) Multiple PE headers: A PE program that has more than one PE header must be regarded as viral properties. Using hexadecimal byte code to indicate abnormal structure above.

9) The virtual address of A section points to OxC000000 and attempt to obtain control of the underlying.

10) SizeOfCode value of Optional header field is incorrect: most viruses add a new section of code in the PE program will not modify the SizeOfCode value of optional header fields. If the calculation of the sizes of all the code sections doses not match with the SizeOfCode value, we believe that the PE file is a new section added by the virus. Extract corresponding sequence feature based on this.

C. Heuristic signature extraction

Divide all the files into virus sample set and normal sample set and count the number of various abnormalities in the two sample set which is called frequency. Calculate the posterior probability according to Bayes' formula:

$$p(w_i \mid X) = \frac{p(w_i \mid X)p(w_i)}{\sum_{j=1}^{c} p(w_j \mid X)p(w_j)}$$

Pattern vector X consist of file abnormal structure Heuristic. Such as $\mathbf{X_1}$=[1,1,0,…,1], 1 indicates that this heuristics feature exits and 0 indicates that this heuristics feature does not exit.

$p(X \mid w_i)$ is the conditional probability density that Pattern vector X is in state w_i. $p(w_i)$ is the priori probability of class w_i.

Then calculate the separability based on entropy function. Set up $\sum_{i=1}^{c} p(w_i \mid x) = 1$ .Then the entropy is $H_c(p) = -\sum_{i=1}^{c} p(w_i \mid x)\log p(w_i \mid x)$.

Take the expectations of entropy[8] : $J_H = E_x\left[-\sum_{i=1}^{c} p(w_i \mid x)\log p(w_i \mid x)\right]$ as severability determine function.

The test selects 297 virus samples of different types and 2600 normal sample files and selects features as Figure 4.

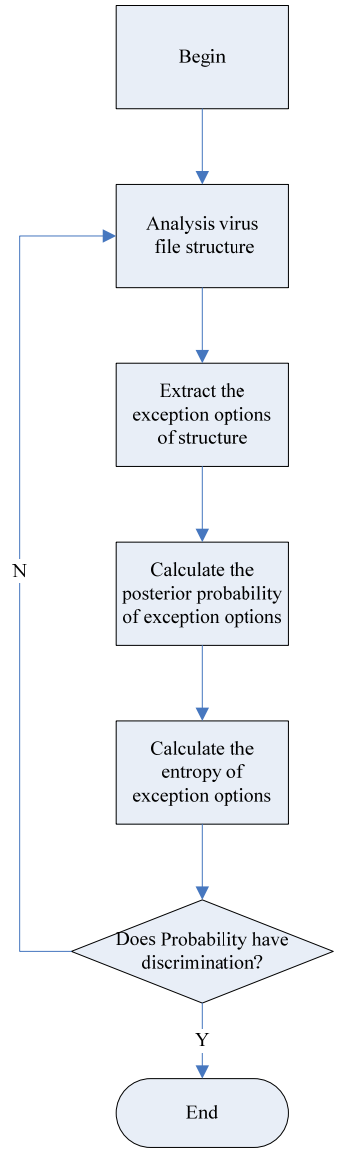

Figure 4. Feature Selection workflow.

The smaller entropy of heuristic features is the greater contribution these features make to distinguish between two

types of samples. As a result, select the 12 smallest entropy feature items.

1) e-lfanew of IMAGE-DOS-HEADER;

2) Number Of IMAGE-OPTIONALHEADER32 ' s Sections;

3) BaseOf Code of IMAGE-OPTIONAL-HEADER32;

4) BaseOf Data of IMAGE-OPTIONAL-HEADER32;

5) Image-Base of IMAGE-OPTIONAL-HEADER32;

6) FileAli-gnment of IMAGE-OPTIONAL-HEADER32;

7) Check-Sum of IMAGE-OPTIONAL-HEADER32;

8) IMAGE-OPTIONAL-HEADER32-> DataDirectory[1]. VirtualAddress;

9) IMAGE-OPTIONAL-HEADER32-> DataDirectory[1].Size;

10) IMAGE-OPTIONAL-HEADER32-> DataDirectory[5]. VirtualAddress;

11) IMAGE-OPTIONAL-HEADER32-> DataDirectory[5].Size;

12) Name of IMAGE-SECTION-HEADER.

IV. CLASSIFIER CONSTRUCTOR

Computer virus detection is undecidable so existing algorithms can not accurately detect. Data mining mines information and discover knowledge in the absence of explicit assumptions premise. Machine learning can be used for data analysis, pattern discovery and prediction. So it is a feasible way to detect unknown viruses by using Machine learning methods[9] and Data mining methods[10-12].

A. Classification based on K-means Method

This paper defines sample set as $U=\{U_1,U_2,U_3,\cdots,U_n\}$, each sample use feature vector expressed as $U_i=\{X_{i1},X_{i2},X_{i3},\cdots,X_{im}\}$, X_{im} represents the m-th feature of sample i.

Traditional K-means algorithm divides data set into different categories through an iterative process, makes criterion function which is used for evaluating performance of clustering achieve optimal, thereby making each generated cluster internal compact and relative independence between classes with better clustering results. The algorithm is as follows.

For the sample set U, select k samples arbitrarily as as the initial cluster centers:

a) divide samples in samples set into the nearest neighbor clustering by the principle of the minimum distance.

b) Use the sample mean of each cluster as the new cluster center, sample mean is: $\overline{X} = \frac{1}{c_i}\sum_{c_i} x$.

Repeat *a), b)* steps until no change then get k-clustering.

K-means algorithm can get better clustering effect for large data sets and have clear distinction between class and class. However the downside is that the result depends on the selection of the initial cluster centers. It is easy to lead to local optima and affect the results while the k values of clusters can't be calculated in determining calculation which is determined by subjective judgment. This article uses the improved KNN algorithm to cluster the training sample set, dynamically determines the value of k during the initialization process to obtain the optimum initial cluster center collection.

B. The detection for unknown viruses

By clustering the training set, k cluster centers are obtained: $S_{U_i,U_j} = 1 - \frac{\sum_{m=1}^{m} \frac{X_{im} - X_{jm}}{\mathrm{MAX}(X_{im}, X_{jm})}}{n}$.

The similarity of the two samples is higher, S value is closer to 1. Conversely S is closer to 0.

For the unknown sample X to be detected, the similarity with each cluster is calculated. If the similarity reaches a certain threshold, the sample X is determined to belong to one class to know whether the sample is virus file. Through several experiments analysis, when the similarity is up to 0.8, it can determine which class the sample belongs to and achieve a higher accuracy and lower false positive rate at the same time. When an unknown file does not match with all of the clusters (does not reach the similarity threshold value), this sample is used as a new cluster center and marked as the new class that feature library can not identify, then it enters the next round testing of samples. For all unidentified classes in system maintenance, they are determined whether the virus with other method such as behavior characteristic analysis thus deciding to remain in the feature library or being removed. In order to maintain the feature library timeliness of information, reduce the match detection time consuming that information redundancy causes, besides, it is impelled to delete the items that have few number of clusters and no new file feature matched with long time.

C. Improved KNN classification algorithm

Detecting unknown viruses accurately requires the use of accurate and efficient classification algorithm. KNN algorithm method is robust for noise training data and easy to use and very effective for large-scale data. KNN algorithm is an instance-based learning method. It finds K nearest neighbors of object X_d to be identified in N-known samples. Set class ω_i have N_i samples, $i \leqslant N$. If k_1, k_2, …, k_c are separately the sample number of ω_1, ω_2 ,..., ω_c class in K nearest neighbors and the discriminant function is defined as: $g_i(X_d) = k_i, i = 1,2,...,c$.

Decision rule is: $g_j(x_d) = \max_i k_i$, so decision $x_d \in \omega_j$.

The status of x_d object waiting for recognition of K-nearest neighbors samples are equal in KNN algorithm, but in the practical application, x_d object waiting for recognition has different distances from its K nearest neighbors and makes different contribution of decision making. What's more, what has been found in virus detection practice is that Feature vectors' different properties make different contributions to the file's dubiety. Some properties can play a decisive role while some other properties play a secondary

role or do not work. Therefore, the attribute selection and accuracy of the results are closely connected. Therefore, the KNN algorithm is improved from the following two aspects to adapt to the detection of unknown viruses.

1) The attributes of different dimensions in the feature vectors are granted different weights ψ_k. The weight ψ_k is set based on degree of harm that heuristic makes to the computer system. $\psi_k \in [1,2,3,...,10]$, weight value of 1 indicates least threatening while weight value of 10 indicates most threatening. File feature vector is $\boldsymbol{X}=(t_1(x),t_2(x),\ldots, t_{10}(x))$, $t_i(x)=\{0, 1\}$, $1 \leqslant i \leqslant 10$. The distance between object to be identified and any neighborhood is:

$$dist(x_d, x_i) = \sqrt{\sum_{k=1}^{10} \psi_k (t_k(x_d) - t_k(x_i))^2} .$$

2) Different distances of virus samples have different contributions. The distances of samples are closer and they are more similar. So weight the distances of K neighbors different μ_k to reflect the degree of contribution of neighborhood. It sets inverse of distance square from the identified object as its distance weight:

$$\mu_k = \frac{1}{dist(x_d, x_i)^2} .$$

Improve the decision making rule as two dimensional determination: not only compare the number of samples in different categories but also compare the weight sum of different vectors' distance in k vectors. Because the detection of the virus is a dichotomous classification problem, there are two classification for given samples: ω_1 = 0 represents that the sample is normal procedure and ω_2 =1 represents that the sample is virus. Then $g_j(x_d) = \max_i k_i$ is modified as:

$$g_1(x_d) = \begin{cases} k_1 > k_2 \\ \sum_{i \in \omega_1} \mu_i dist(x_d, x_i) > \mu_j dist(x_d, x_j) \end{cases}$$

The decision x_d belongs to ω_1, otherwise the decision x_d belongs to ω_2.

This paper gets improved KNN algorithm through the method above. This algorithm is more suitable for detection of unknown viruses in "Cloud security" system. Decision rules that it sets reduce the chance that the virus samples to be identified as of normal procedures and discover unknown virus as much as possible. File discovery module of "Cloud security" system focuses on the recognition rate of unknown virus and its goal is to detect all potential unknown viruses and upload to the server for analysis. Although initially a normal file is more likely to be misinterpreted as suspicious file, after anti-virus expert and a phase of self-learning with "black and white list" control method, normal procedures will be progressively added to the "white list", thus probability that normal procedure is misinterpreted as a suspicious file becomes smaller and smaller.

V. EXPERIMENTS AND RESULTS

The total number of samples space for the experiment is 300. The samples are divided into the normal program and virus File, shown as Table 1. Normal procedures are selected in newly installed Windows XP system C drive. Virus files are downloaded from the Information Security Forum including Trojans, backdoor programs, worms.

TABLE I. EXPERIMENTAL SAMPLE DATA

	Sample space	Training set	Test set
Normal file	110	80	30
Virus file	300	200	100
Total	410	280	130

The experiment focuses on the number of virus files that are identified as the normal procedure(False Negative, FN) and normal procedures that are identified as virus file(False Positive, FP) . If virus behavior characteristic feature vector match with the normal file characteristic vector, the system will be false positives. If there is no feature to match with new attack behavior, the system will be false negative.

Main purpose is to verify the above experimental model for unknown virus file classification and recognition. For the determination of the result is a "yes" and "no" dichotomous question, This problem can produce four kinds of forecasting results, namely: the normal procedure sentenced as normal, marked as TP;(1)The normal procedure is determined for the virus, marked as TN;(2)Contracting the virus files as normal, marked as FP: Contracting the virus file as a virus marked as FN. Accuracy rate of the test samples were correctly classified probability; False positive rate of 175 to be classified as normal file sentenced probability of virus. As shown in Table Ⅱ.

TABLE II. CLASSIFICATION TEST RESULTS

Category Name	Num	k_means Algorithm		Improved KNN Algorithm	
		accuracy	False positives	accuracy	False positives
Trojans	43	89.2	3.4	88.2	3.8
Backdoor	31	78.7	4.4	88.7	3.8
Worm	23	86.3	5.3	89.3	3.3
Normal	24	90.7	6.1	93.7	4.1

Experimental results: Using the improved KNN algorithm relative to the general k_means clustering has better classification and identification ability to unknown files, accuracy and false alarm rate the former than the latter. The result is satisfying.

Comparing improved KNN algorithm and other algorithms based on the unknown virus detection, this model determines the time required for an unknown file is about 1min or so, while vm-based detection model for each sample takes about 2min ~ 3min, you can clearly see the detection efficiency is greatly improved. Comparing with other shelling operation required the detection side, the detection rate and false alarm rate is very accurate. But in fact less detection process operation. For the ever increasing number of new classes, the method has expansibility, therefore in practical applications, there are more advantages.

VI. CONCLUSION

This paper presents a static PE file structure based on heuristic signature extraction and detection strategy. The strategy uses file abnormal structure as the characteristic vector, uses feature extraction method based on discriminant entropy minimization to streamline feature items adaptively and takes advantage of improved KNN algorithm to study and classify the sample. It makes up for deficiencies that unknown virus detection technology used in the industry that is based on behavioral analysis. It has a high detection rate and low false alarm rate. It does not have to unpack , be expanded and do other heavy computing, compared to the detection method based on API sequence. This technique can be used as a separate detection tool and can also be combined with antivirus software as part of heuristic scanning.

ACKNOWLEDGMENT

First of all, I would like to extend my sincere gratitude to my supervisor, Yin Guisheng, for her instructive advice and useful suggestions on my thesis. I am deeply grateful of his help in the completion of this thesis.

I am also deeply indebted to all the other tutors and teachers in Translation Studies for their direct and indirect help to me.

This work was partially supported by the Fundamental Research Funds for the Central Universities under Grant No.HEUCF100608

REFERENCES

[1] Kruegel C, Robertson W, Valeur F, et al. Static Disassembly of Obfuscated Binaries[D]. Santa Barbara, CA, USA: Reliable Software Group, Computer Science Department, University of California, 2004.

[2] Christodorescu M, Jha S. Static Analysis of Executables to Detect Malicious Patterns[C]//Proceedings of the 12th USENIX Security Symposium. BerKeley, CA, USA: [s. n.], 2003.

[3] S Berkovits, etal. Public-key Infrastructure Study: Final Report[R]. Produced by the ITRE Corporation for NIST, 1994. 72-82.

[4] D Boneh, M Franklin. Identity-based Encryption from the Weil Pairing[J]. Advances in Cryptology-CRYPTO'01, Lecture Notes in Computer Science, 2001, 2139: 213-229.

[5] D Krishna Sandeep Reddy, Arun K Pujari. N-gram analysis for computer virus detection, Journal in Computer Virology Vol 2,NO 3 , 2006 12, 231-239.

[6] Gregory P.. Computer Viruses For Dummies.[s.1.]:Wiley Publishing. 2004:45-61P.

[7] Muazzam Siddiqui, Morgan C. Wang, Joohan Lee. A survey of data mining techniques for malware detection using file features. Proceedings of the 46th Annual Southeast Regional Conference on XX.2008: 509- 510P.

[8] PeterSzor. The Art of Computer Virus Research and Defense [M]. USA: Addison-Wesley, 2005.

[9] Shabtai A, Moskovitch R.Detection of malicious code by applyingmachine learning classifiers on static features：A state of the art sur-vey[J].Journal Information Security Tech Report, 2009, 14(1): 16-29.

[10] Siddiqui M, Wang M C, Lee J.A survey of data mining tech-niques for malware detection using file features[C]//Proc of the46th Annual Southeast Regional Conference on XX, 2008：509-510.

[11] Ye Yanfang, Wang Dingding, Li Tao, et al.MDS：intelligent malware detection system[C]//Proceedings of the 13th ACM SIG-KDD International Conference on Knowledge Discovery and Da-ta Mining, KDD'07, 2007：1043-1047.

[12] Han Zhixue, Feng Shaorong, Ye Yanfang, et al.A parameter-freehybrid clustering algorithm used for malware categorization [C]//Proc of the 3rd International Conference on Anti Counterfeit-ing, Security, and Identification in Communication, 2009：480-483.