Defense of Computer Network Viruses Based on Data Mining Technology

Cen Zuo
(Corresponding author: Cen Zuo)
Chongqing College of Electronic Engineering
Shapingba, Chongqing 401331, China
(Email: cenzuocq@126.com)
(Received Dec. 12, 2017; revised and accepted Apr. 6, 2018)

Abstract

Computer network has great influence on people's work and life. Enterprise information, personal information and even national information are stored in computers. Therefore, computer network security has become a problem that cannot be ignored. The biggest threat to computer network security is computer network virus. To cope with the invasion of different viruses, this study analyzed the characteristics of network viruses and designed computer data mining module by combining data mining technology with dynamic behavioral intercept technique to mine hidden information and determine whether there was virus. The method was applied in the detection of Trojan horse virus on network. The efficacy of defense of Trojan horse viruses was tested through indexes including false alarm rate, accuracy rate, omission rate and information gain value extracted based on API characteristics. The detection suggests that data mining technology is useful in the defense of computer network viruses and has a favorable development prospect.

Keywords: Computer Network; Data Mining Technology; Virus

1 Introduction

Computer network viruses spread with the constant development of computer network technology. Computer virus can be created through compilation on advanced program, and other viruses can be derived through modification. Therefore there are diverse network viruses with certain uncertainty [1, 2]. El-Sayed *et al.* [3] established mathematical model for the transmission process of computer viruses, which is beneficial to the understanding of computer virus behaviors and prevention of viruses. They put forward fractional order SIR (Susceptible, infective and removal) model to study computer virus.

To effectively resist computer viruses, Shahrear *et al.* [4] proposed a compartmental model, made changes for each compartment, and analyzed the local stability of virus-free and endemic disease equilibrium models based on basic reproduction number. Trojan horse virus was selected as the research subject in this study. Differing from other viruses which are capable of self-reproduction, Trojan horse viruses will not intentionally infect other documents but confuse users to download and then invade host to steal document information [5]. Feature code scanning technology, active defense technology and network monitoring [6] are the main technologies for the detection of Trojan horse viruses currently. Data mining technology was used in this study. Application of data mining technology in network virus defense system is a new idea for enhancing computer network security [7]. The technology takes data in a certain range as the research subjects and collects, analyzes and classifies them; the processing result is regarded as the determination basis for a potential relationship and data regularity [8]. The preparation of data and searching and presentation of data regularity are the important components of data mining technology [9]. It is found that data mining has a high accuracy, low omission ratio and low false alarm rate in the detection of Trojan horse viruses, suggesting data mining has a bright development prospect in the detection of computer viruses.

2 Data Mining Technology

Three important components of data mining technology can be subdivided into data preprocessing module, decision-making module, data collection module, data mining module and rule base module [10, 11]. The main process steps of data mining technology are data collection, preprocessing, data cleaning, data mining, modeling and model evaluation.

2.1 Classification Algorithm

The common data mining classification algorithms mainly includes Bayesian algorithm, support vector machine and

decision-making tree. Bayesian algorithm is to calculate the prior probability of a known object firstly based on Bayesian classifier and then the probability of a category using theorem formula. Bayesian classifier can optimize classification model based on minimum error rate. The algorithm has advantages of high preciseness and simple operation. Support vector machine can map linearly inseparable samples in a low-dimensional space to linearly separable samples in a high-dimensional space. Moreover it can establish optimal segmentation plane in data space and learn globally optimal solution. Decision-making tree is a tree-structure model composing of branches in data structure. It represents a decision making system and is a frequently used multi-layer classifier in data mining [12].

2.2 Application Programming Interface Function

Data mining technology is extensively applied. Detection of Trojan horse viruses, defense of viruses and network monitoring all need data mining technology. Especially in the detection of Trojan horse viruses, it can distinguish Trojan horse viruses with normal documents through learning relevant rules based on the features of Trojan horse viruses. Application programming interface (API) function and relevant parameters were regarded as the data set. The behavioral characteristics of Trojan horse viruses were tracked, and signal gain was assigned with characteristic value. Finally features were extracted through effective classification algorithm, with high accuracy and small error.

3 Data Mining Technology - Taking Trojan Horse Viruses as an Example

Many API functions need to be called if there are a larger amount of malicious codes. Therefore it is critical to mine combinations which are beneficial to the detection of malicious codes in API function. The structural chart of the detection system is shown in Figure 1.

As shown in Figure 1, the system was mainly composed of tracking module, preprocessing module and data mining and analysis module. The tracking module was responsible for analyzing dynamic link library (DLL) formed in sample operation, comparing it with illegal behaviors captured in training stage, and monitoring specified API function. When the defined API function was detected, the system interrupted, and then the function given by the current parameters was analyzed. Preprocessing module was responsible for loading program to sample database in batches. Mining and analysis module is the decision-making module of the system. API functions which were called by Trojan horse viruses could also be called in normal documents; the combination of those functions threatened the safety of computer network. Then the data mining module was designed in details.

3.1 Tracking and Monitoring of API Function

API function is usually acquired through Windows Debug AP and APIHOOK. API function was the core in the tracking module. Windows Debug API could call API, load a program, and bind itself on running programs; moreover it could collect image bases addresses and call addresses in the process of debugging. If an incident was correlated to debugging, debugger would be immediately triggered, and monitoring process would be activated, including creating process, thread, loading and releasing. Debugger could respond to and handle incidents. Submodule was responsible for the interactive behaviors between the process of debugger and debugged process and help debuggers realize different debugging functions. In short, API function could coordinate with debugging submodules.

3.1.1 Procedures of Tracking and Monitoring

The first step was to input samples, start monitoring program, and suspend all threads. The second step was to analyze library files which were called by debugging programs to acquire derived functions and the entry addresses of different functions. The third step was to set breakpoints on the entry addresses of functions. The fourth step was to debug incidents and collect the information of debugging process if there was interruption at the breakpoints. The fifth step was to execute debugging process repeatedly till the end and acquire behavioral reports.

3.1.2 Implementation of Monitoring

The first step was to obtain the names of process and process modules in EnumProcess function, including enumerating process, enumerating process module, and acquiring the names of process modules.

1) Enumerating process:

```
typedef BOOL(_stdcall*ENUMPROCESSES)(
    DWORD*pProcessIds, \\ ID array chains of
                          \\ pointing process
    DWORD*cb, \\ Size of ID array
    DWORD*pBytesReturened \\ Bytes returned
    );
```

2) Enumerating process module:

```
typedef BOOL(_stdcall*ENUMPROCESSESMODULES)(
    HANDLEh_Process, \\ Process handle
    HMODULE*lphModule, \\ Array chain
    DWORD cb, \\ Size of array (bytes)
    LPDWORD lpcbNeeded \\ Total number of
                       \\ bytes needed by memory handle
```

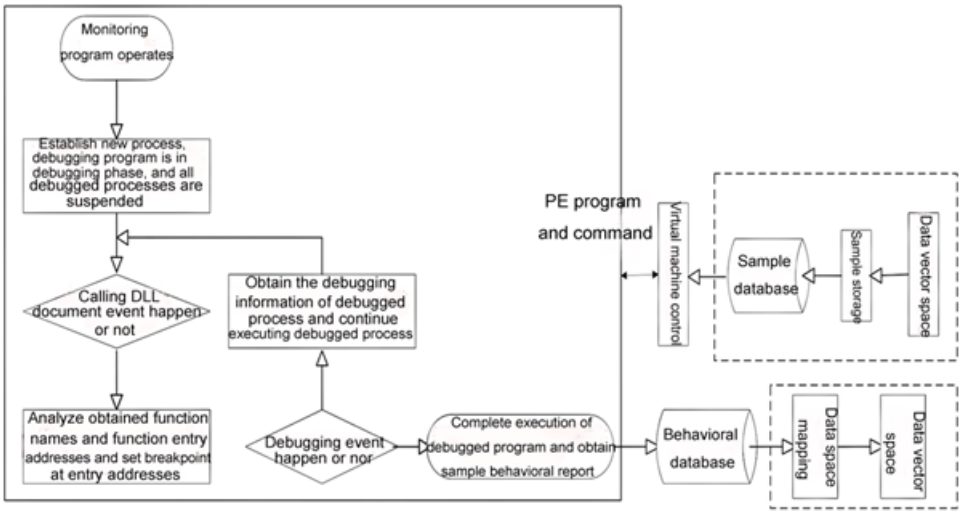

Figure 1: The flow chart of the detection system

);

3) Acquiring the names of process modules;

typedef DWORD(_stdcall*GETMODULEFILENAMEX)(
HANDLE hProcess, \\ Process handle
HMODULE hModule, \\ Process handle
LPTSTR lpFileName, \\ Full path of module storage
DWORD nSize \\ lpFileName Size of buffer (character)
);

3.2 Data Mining and Analysis

WEKA platform is the most commonly used analysis tool in data mining. It is an intelligent analysis platform based on Java environment which is feasible to all operation environments including data preprocessing, cluster analysis and classification prediction.

3.2.1 Vector Space

In the capture of API function by the tracking module, function was regarded as a feature, and sample vector space was established for normal samples and Trojan horse virus samples separately. If the API function was in Trojan horse virus, then the characteristic value was set as 1; if it was not, then the characteristic value was set as 0. In this way, vector space could be established for legal programs, all samples could be represented as the vectors of the vector space, and relevant models could be established using data mining technology. In the construction of data set, ID stands for the number of program file sample, Function stands for whether there was presence of corresponding function in the sample, and Type stands for the type of corresponding sample, normal file or Trojan horse virus.

The model building of characteristic value is as follows. The entropy of random variable E was:

$$H(E) = -\sum_{i=1}^{n} p_i(E = c_i) \log_2 p_i(E = c_i), \tag{1}$$

where there were n values for E and pi stands for the probability when E was equal to ci. Entropy could reflect the distribution condition and distinction degree of E. High value of entropy indicated uniform distribution and low distinction degree, while small value of entropy shows large distinction degree. When random variable G was known, the conditional entropy of E was:

$$H(E|G) = -\sum_{s=1}^{m} p_s(Y = c_s) \times H(E|G = c_s). \tag{2}$$

$H(E|G)$ means E was still uncertain even if G was known. Random variable E could be either malicious code or non-malicious code. Hence

$$H(E) = -\frac{|c|}{|x|} \log_2 \frac{|c|}{|x|} - \frac{|a|}{|x|} \log_2 \frac{|a|}{|x|} \tag{3}$$

where c stands for program containing malicious code, a stands for normal program, $|x| = |c| + |b|$ stands for the total number of programs, and G stands for specified API function. Then

$$H(E|G) = -\sum_{n=0}^{1} \frac{|x_n|}{|x|} (-\frac{|c_n|}{|x_n|} \log_2 \frac{|c_n|}{|x_n|} - \frac{|a_n|}{|x_n|} \log_2 \frac{|a_n|}{|x_n|}) \tag{4}$$

Information gain $IG(E|G)$ was:

$$IG(E|G) = H(E|G) - H(X). \tag{5}$$

The above calculation was based on the difference of probability when specific API function program contained malicious code and probability when program which did not call the API function contained malicious code. A large difference indicated a high probability of the presence of API function in viruses. Through calculation, characteristics which had large effects in the detection were reserved.

3.2.2 Validation Method

Cross validation was involved in this study. K-fold cross validation could randomly divide data set into k subsets. One subset was taken as the test set each time, and the remaining subsets were regarded as the training sets. After k times of repetition, the average testing result was taken as the final result. K-fold complete cross validation was adopted as the division of data had significant fluctuation. That validation method could take all possible division methods that could divide the data set into k subsets into account. K-fold cross validation repeated for n times, and the average value was taken as the final result. K-fold complete cross validation is complete though its accuracy is high. Hence K-fold hierarchical cross validation was proposed. Then the information gain values of Kernel32. dll function and its parameters in the detection of Trojan horse viruses were statistically analyzed.

3.3 Interception Method for Function Call

Interception methods for function call mainly included modifying System Service Descriptor Table (SSDT), monitoring CreateremoteThread, monitoring NtCreateProcessEx function and intercepting NtCreateProcessEx function. The operation of process initiation could be realized by intercepting function through modifying SSDT or shadow SSDT. In the monitoring of CreateremoteThread, CreateremoteThread was the tool for remote injection of Trojan horse virus. Monitoring CreateremoteThread was helpful to the checking of system abnormality. The main aim of monitoring NtCreateProcessEx function and intercepting NtCreateProcessEx function was to intercept through API function, achieving the detection of Trojan horse viruses through process monitoring, and introducing new safety monitoring system. In the process of interception, the path of function call and document names were acquired, as shown in Algorithm 1.

Algorithm 1 The process of interception

```
1: GetModuleFileName(NULL, cur_mod, sizeof(cur_mod));
2: while (pMyAPIInfo[count].module_name != NULL)
   do
3:    Strcpy(pszModuleName, pMyAPIInfo[count].module_name);
4:    Strcpy(pszModuleName, pMyAPIInfo[count].function_name);
5:    Strcpy(pszMyFuncName, pMyAPIInfo[count].myfunc);
6: end while
7: if (pszAPIName!=NULL) then
8:    pszParameterList = strchr(pszAPIName,'('));
9: end if
10: if (pszParameterList != NULL) then
11:    pszParameterList[0] = '\0';
12:    pszParameterList++;
13: end if
14: if ((myFunc=(tagMyFunc)GetProcAddress(hMyDLL,
    pszMyFuncName)) = NULL) then
15:    Return false;
16: end if
17: pAPIInfo=HookAPIfunction(pszModuleName,
    pszAPName, myFunc);
18: ...
```

4 Evaluation Based on Tests

The false alarm rate, accuracy and omission rate were taken as the evaluation indexes for the detection system. False alarm rate referred to the percentage of the normal files which were wrongly classified as Trojan horse viruses; accuracy rate referred to the percentage of program files which were evaluated accurately; omission rate referred to the percentage of Trojan horse viruses which were wrongly evaluated as normal files. In the test, there were 900 legal files and 1300 Trojan horse viruses.

4.1 Native API Related Information After Interception

Table 1 exhibits Native API related information after program interception. Through comparison and analysis, the intercepted API function was regarded as a characteristic for establishing data format of the whole WEKA (Waikato Environment for Knowledge Analysis) analysis. The data were analyzed using Naive Bayesian algorithm, support vector machine and J48 algorithm.

4.2 Analysis of Trojan Horse Virus Identification with Three Algorithms

Table 2 exhibits that the three algorithms performed differently in identifying Trojan horse viruses. In conclusion, defense based on data mining had favorable effect in identifying Trojan horse viruses; all the three algorithms could identify more than 1200 Trojan horse viruses (90%).

4.3 K-fold Hierarchical Cross-validation

The average results of the false alarm rate, accuracy and omission rate when K-fold hierarchical cross-validation was sued are shown in Table 3.

Table 3 exhibits that the detection method based on data mining performed best in the detection of Trojan horse viruses because of its high detection rate and low false alarm rate. Among the three algorithms, the omission rate of Naive Bayesian algorithm was significantly higher than that of the other two algorithms; the accuracy of support vector machine was the highest; the false alarm rate of J48 algorithm was the highest, and next was Naive Bayesian algorithm. Overall the accuracy of them was all high, larger than 95%. They were able to make beneficial determination based on the vector space constructed by intercepted API function, and moreover the technology was verified as quite effective in detecting Trojan horse viruses.

Table 1: Native API related information after interception

Timestamp	API ID	Native API
17	4	NtAllocate VirtualMemory (allocate virtual memory)
18	23	NtQueryVirtualMemory (query virtual memory)
19	15	NtFreeVirtualMemory (free virtual memory)
20	7	NtSetEvent (setting event object)
21	16	NtCreateEvent (creating event object)
22	18	NtCancelTimer (cancel current time setting)
23	34	NtSetTimer (resetting time)
24	69	NtDelayExecution (delay execution)
25	44	NtClearEvent (clearing event object)
26	9	NtOpenThreadToken (opening thread mark)
27	8	NtAllocateVirtualMemory (allocate virtual memory)

Table 2: Relevant information for identification of Trojan horse viruses

	Number of Trojan horse viruses identified	Number of Trojan horse viruses identified as legal	Number of non-classified
Naive Bayesian algorithm	1223	71	6
Support vector machine	1270	28	2
J48 algorithm	1265	32	3

4.4 Analysis of Kernel32.dll Parameters in the Detection

Table 4 suggests the information gain values of Kernel32.dll function in the detection of Trojan horse viruses. Features were extracted through feature extraction algorithm, and the file features which existed or did not exist were examined. The values of information gain were small, indicating that the reserved function had high distinction degree and the classification efficiency was high.

5 Discussion and Conclusion

Computer plays an increasingly larger role in the life and work of people in the process of its continuous development and improvement, which imposes a huge threat to the security of computer network. There is no effective way to prevent viruses from intruding into computer network, and only defense is feasible [13]. Defense of network viruses is challenging as computer network viruses have multiple transmission modes and strong pertinence. Trojan horse virus as a kind of computer network viruses is occult; hence monitoring Trojan horse viruses simply and blindly will consume a large number of system resources, leading to the occurrence of false alarm [14, 15]. Data mining can excavate processes concealed, analyze API function call, summarize a new detection scheme, and establish new detection system to reduce false alarm rate. In this study, API tracking module was used, and the false alarm rate, accuracy and omission rate of three algorithms were tested to investigate the role of data mining technology in the detection of Trojan horse viruses. Moreover it was found from the extraction of information gain based on the characteristic values of API function that the classification efficiency was improved. Data mining could make beneficial judgment through the vector space established based on the intercepted API function, which enhanced identification efficiency. Hence data mining is quite effective in detecting Trojan horse viruses. In conclusion, data mining technology is useful in the detection of computer network viruses, which is worth promotion.

References

[1] C. H. Zhai, "Discussion on the utilization of prevention Technology of Computer Network Security," *Applied Mechanics and Materials*, vol. 556-562, pp. 5523–5525, 2014.

[2] R. K. Upadhyay, S. Kumari, A. K. Misra, "Modeling the virus dynamics in computer network with SVEIR model and nonlinear incident rate," *Journal of Applied Mathematics and Computing*, vol. 54, no. 1-2, pp. 485–509, 2017.

[3] A. M. A. El-Sayed, A. A. M. Arafa, M. Khali, A. Hassan, "A mathematical model with memory for propagation of computer virus under human intervention," *Progress in Fractional Differentiation and Applications*, vol. 2, pp. 105–113, 2016.

[4] P. Shahrear, A. K. Chakraborty, M. A. Islam, U. Habiba, "Analysis of computer virus propagation based on compartmental model," *Applied and Computational Mathematics*, vol. 7, no. 1-2, pp. 12–21, 2018.

[5] M. M. Saudi, A. M. Abuzaid, B. M. Taib, Z. H. Abdullah, "Designing a new model for Trojan horse de-

Table 3: The average test results

	False alarm rate	Accuracy	Omission rate
Naive Bayesian algorithm	2.4%	95.61%	5.5%
Support vector machine	1.6%	98.08%	2.2%
J48 algorithm	3.1%	97.26%	2.5%

Table 4: The information gain values of Kernel32.dll parameters in the detection of Trojan horse viruses

Kernel32.dll	Parameters	IG	Description of functions
Open process	dwDesiredAccess	12.12%	Needed process rights
	bInheritHandle	6.02%	Whether can inherit or not
Allocation space	hProcess	14.23%	Process handle
	IpAddress	4.56%	Address pointer
	flProtect	4.56%	Page protection attribute
	dwSize	4.56%	Size

tection using sequential minimal optimization," *Lecture Notes in Electrical Engineering*, vol. 315, pp. 739–746, 2015.

[6] J. A. Ortega, D. Fuentes, J. A. Alvarez, L. Gonzalez-abril, F. Velasco, "A novel approach to Trojan horse detection in mobile phones messaging and Bluetooth services," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 8, pp. 1457–1471, 2011.

[7] L. P. Feng, H. B. Wang, S. Q. Feng, "Computer network virus propagation model based on biology principle," *Computer Engineering*, vol. 37, pp. 155–157, 2011.

[8] H. T. Tavani, "Genomic research and data-mining technology: Implications for personal privacy and informed consent," *Ethics and Information Technology*, vol. 6, no. 1, pp. 15–28, 2004.

[9] J. J. Xie, "Integration of GIS and data mining technology to enhance the pavement management decision making," *Journal of Transportation Engineering*, vol. 136, no. 4, pp. 332–341, 2010.

[10] Q. Jiang, H. Lin, J. Li, J. Liu, "The research on spatial data mining module based on multi-objective optimization model for decision support system," in *IEEE Second WRI Global Congress on Intelligent Systems (GCIS'10)*, pp. 299–302, 2010.

[11] C. Chen, A. Chen, "Using data mining technology to provide a recommendation service in the digital library," *The Electronic Library*, vol. 25, no. 6, pp. 711–724, 2007.

[12] P. Sangitab, S. R. Deshmukh, "Use of support vector machine, decision tree and naive Bayesian techniques for wind speed classification," in *International Conference on Power and Energy Systems (ICPS'11)*, pp. 1–8, 2011.

[13] P. Kata, *The in Vitro Effects of Herpes Simplex Virus and Rubella Virus on Autophagy*, Ph.D. Thesis, University of Szeged, 2014.

[14] B. Liu, C. Lin, Q. Jian, J. He, P. Ungsunan, "A NetFlow based flow analysis and monitoring system in enterprise networks," *Computer Networks*, vol. 52, no. 5, pp. 1074–1092, 2008.

[15] S. Zhong, X. Cheng, T. Chen, "Data hiding in a kind of PDF texts for secret communication," *International Journal of Network Security*, vol. 4, pp. 17–26, 2007.

Biography

Cen Zuo is a teacher from the school of computing of Chongqing College Of Electronic Engineering, China. His interests of research are computer application and software engineering. He gained the bachelor's degree from the software college of Chongqing University of Posts and Telecommunications in 2009 and now is studying for a master's degree in the Computer College of Chongqing University. He have participated in many provincial-level subjects such as Research and Practice of Fusion of Innovative and Entrepreneurial Education of Higher Vocational College and Status Analysis and Countermeasure Study of Modern Apprenticeship Practice in Higher Vocational College and published more than 10 papers including one EI indexed paper titled Research on Scheduling Algorithm for Cloud Computing and one CSCD core paper titled The Route Optimization of the Shortest Path with Fruit Fly Optimization Algorithm, and co-edited a computer science textbook titled Management Information System Design Practice Guide. Moreover he taught courses such as Fundamentals of Computer Culture, Program Design Fundamentals and Database Fundamentals. He also gained honorary titles such as school-level excellent education workers.