Research on Active Defence Technology with Virus Based on K-Nearest Neighbor Algorithm of Kernel

Xuedou Yu
Department of Computer Science and Technology. Dezhou University
Dezhou Shandong, China
E-mail: yuxuedou@163.com

Abstract—**In this paper, the active defence technology against virus, the detection technology of known variants of virus and the active defence technology against virus based on K-NN are introduced. Their disadvantages are analyzed. A solution which is based on k-nearest neighbor algorithm of kernel and the active defence technology against virus is proposed. The solution can distinguish safe process from the unknown viruses process efficiently, and can detect unknown viruses more accurately.**

Key words-Virus; Active Defence; Kernel Function; K-Nearest Neighbor Algorithm

1. INTRODUCTION

In recent years the security of computer systems is under growing threat, because the new virus, the new Trojan virus, and the malicious code are increasing at an alarming rate. But now the majority of antivirus software is based on the virus signatures for the virus detected, and the phenomenon that the prevention from new virus has always been lagging behind. The computer viruses and Trojan virus arisen in recent years can take advantage of "automatic deformation technology" to escape from being killed [1][2]. So the antivirus software which based on virus signatures loses the purpose of prevention in time.

In order to deal with the new problem, "initiative anti-virus technology" is proposed and has been a mainstream technology in antivirus software [2]. The technology can judge whether a procedure is harmful, mainly through calculating the similarity of suspicious procedure and the known virus to achieve. We can enhance the security of system this way. However, it is determined by the user completely whether a new process can run, and a common user don't know whether a process is harmful, for it is too high for the users.

There are two ways to differentiate the normal process from virus process. One is pick-up the signature of virus through searching the sequences of API or function of the system or specific software development kit, and then judge whether a process is the deformation of known virus. Another way to judge whether a process is harmful is based on the characteristic of behavior. These ways can judge a harmful process more exactly, but the efficiency of this method is relatively low.

The disadvantage of the active defence system based on k-nearest neighbor algorithm is that the classified effect whose distribution of class is the non-Gaussian distribution and the non-ellipse distribution is bad, and because its time complexity is O(N2) (N is the total number of data point) [3] [4], the method cannot meet the demand of the project when the data point number is large.

In this paper, a new way was present to judge whether a process is harmful. This method combines the positive features of both active defence technology and the k-nearest neighbor algorithm of kernel.

2. OVERALL SYSTEM STRUCTURE

The framework of active defence system against virus is showed in Figure 1. The system consists of three sets, which are "security file set" S, " virus set" V and "suspicious file set" Q. The security file signatures are stored in S, the known virus signatures are stored in V, the suspicious file signatures are stored in Q, and U represents full set, and the relationship between the various sets is:

$$U = S \bigcup V \bigcup Q$$

$$S \bigcap V = \emptyset$$

$$V \bigcap Q = \emptyset$$

$$S \bigcap Q = \emptyset$$

The information of the three sets can be obtained through the following method. The flow chart of active defence anti-virus system is showed in Figure 2.

2.1 The establishment of "security file set"

After finishing installing computer system in the no-virus environment, we pick-up the signatures of all the system file by signature extraction software, and encrypt through MD5 then save it to "security file set".

2.2 The establishment of "virus set"

We pick-up the signatures of the known virus by signature extraction software, and encrypt through MD5 then save it to "virus set".

2.3 The establishment of "suspicious set"

The monitor process intercepts and captures the running process and suspends it for a while. Then pick-up the

978-0-7695-3682-8/09 $25.00 © 2009 IEEE
DOI 10.1109/ESIAT.2009.121

signatures of the process by signature extraction software and encrypt it through MD5, then compare it with the

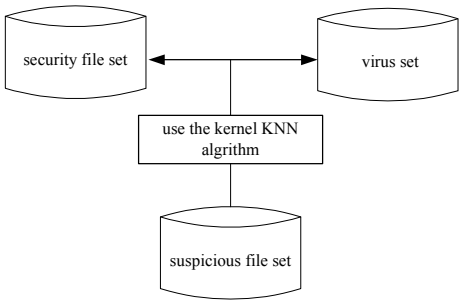

Figure 1 The framework of virus active defence system

background "security file database", if the running process is a legal process, then let it runs, if not so compare its signature with the signatures in "virus signatures database", if it is harmful, then kill it, otherwise suspend the process and save it to "suspicious file database".

2.4 How to deal with the suspicious process

Whether to allow the process whose signature was saved to "suspicious file set" to run depends on the detection by K-Nearest Neighbor Algorithm of Kernel.

2.4.1 The suspicious operation sequence

After the analysis of the virus and Trojans horse, we can sum up several common characteristics as follow.

(1) Abnormal file access. When the harmful process infects other executable files, it is generally to traverse all the executable files, which is not the characteristic of common files, and if finding a process with the kind of characteristic, we can recognize it is a suspicious process.

(2) Abnormal memory operation. When the virus infects the common files and runs, it will erase or move or replace the memory, and the operation is the unique to the virus generally.

(3) The destructive operation. The virus has destructive operation generally, such as erasing or modifying certain files or even formatting the disk, so if we detect a process performing destructive operations, we can determine it is a harmful.

(4) The leaking secrets operation. In order to steal information from users, Trojan horse will collect users' information automatically. This is an important feature of Trojan horse.

(5) The automatic access to mobile storage media. It is an important way that Trojan horse steals information by automatically accessing to mobile storage media, so the process which can access to mobile storage media automatically is commonly a Trojan horse.

(6) Self-replication operation. If a process has the self-replication code sequence, we should suspect it is virus.

(7) Abnormal stack operation. If there is a character array in local variable, and if we assign a long enough character string to the array, the address returned before will

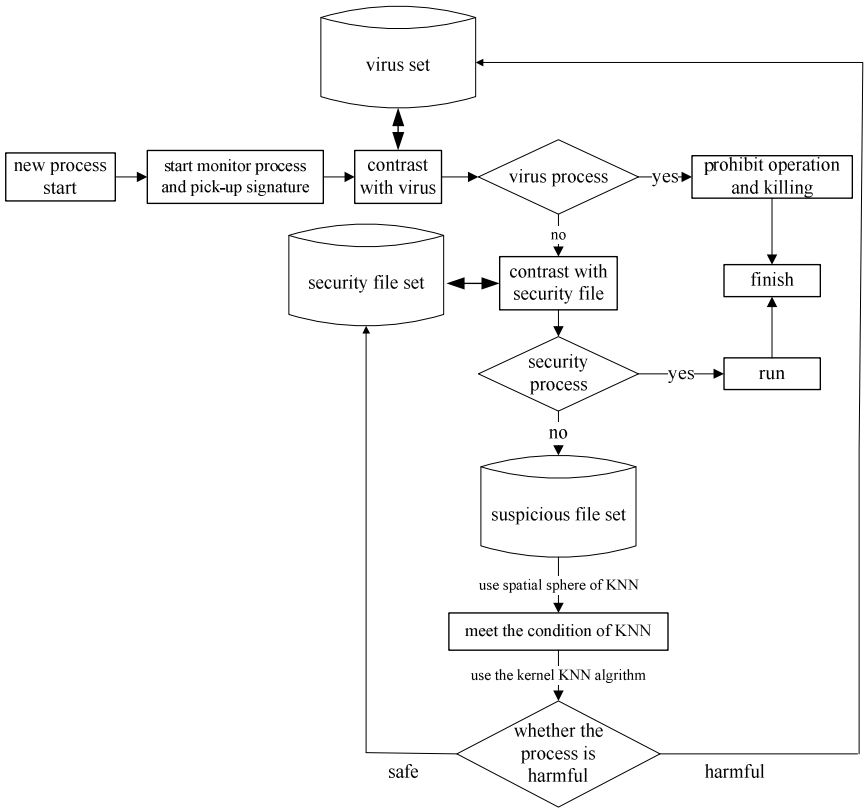

Figure 2 The flow chart of active defence system with virus

be covered. In this way, the buffer overflow happened. So if there is someone submits overlong data, we can list it as a suspected target.

(8) The suspicious jump structure. If the jump address is the address of core DLL, commonly we consider it as a virus or Trojan horse.

(9) The call of private API function. In order to reduce its volume, virus or Trojan horse often calls a large number of API functions, including private API function. The process with this feature can be classified as suspicious ones.

(10) Presence in memory. The virus must be presented in memory for a long-term in order to achieve the purpose of infection and spreading. So a process which automatically resides in memory for a long-term can be considered as a harmful process.

2.4.2 The k-nearest neighbor algorithm

The k-nearest neighbor algorithm is an analogy-based study method. The key technique of k-nearest neighbor algorithm is searching N-dimensional space to find out k training samples which are the nearest to the unknown sample, and the unknown sample is assigned to the public class of the k training samples which are the nearest neighbors and its neighbors are defined by Euclidean distance. The nearest neighbor algorithm is based on such assumption that the classification of an example is similar to the examples in the vicinity of the Euclidean space.

Based on the definition and analysis of suspicious operation sequence above, we can obtain the corresponding code feature and eigenvector of the process by scanning suspicious code in the corresponding instruction sequences operations, the definition of the eigenvector is as follow:

$$x = (t_1(x), t_2(x), \cdots, t_{10}(x)), t_i(x) \in \{0,1\}, 1 \le i \le 10$$

Then, the distance of two arbitrary examples is defined as follow:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{10} (t_k(x_i) - t_k(x_j))^2},$$

$$x = (t_1(x), t_2(x), \cdots, t_{10}(x)) \qquad (1)$$

Because of every attribute has different security level and different suspicious level, and in order to make $d(x_i, x_j)$ more realistic reflection of the distance between the two examples, we can add different weighted value w_j to different dimension $t_i(x)$ according to the characteristics that the virus use and may have. Then the formula (1) can be modified as follow:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{10} [w_k (t_k(x_i) - t_k(x_j))]^2},$$

$$x = (t_1(x), t_2(x), \cdots, t_{10}(x)) \qquad (2)$$

2.4.3 K-NN algorithm based on kernel methods

The K-NN algorithm based on kernel methods is simple, and has a broader applicability than the original K-NN algorithm, but the classification accuracy of this algorithm will be greatly reduced for some more complex issues, in particular the distribution of training samples is very irregular. The kernel methods can make the data map from low-dimensional space which is hard to divide to high-dimensional feature space which can be divided more easily through non-linear transformation, and does not need explicit non-linear transformation. The only need is that calculating the inner product only using the kernel function --the vector of the original features of the new features of the space to ensure that the conversion of space without increasing the complexity of computing, so kernel in the field of pattern recognition applications have been paid more and more attention to.

We can choose all kinds of kernel methods, but in theory, the polynomial approximation can meet any characteristics of the space transform in the appropriate parameters P. In order to enable the experimental result to have the generality and the widespread compatibility, the paper uses polynomial kernel. The kernel, such as the definition of formula (3) as shown.

$$K_p(x, y) = (1 + < x, y >)^p \qquad (3)$$

In order to solve the problems facing K-NN based on traditional Euclidean distance, the literature [8] gives a K-NN algorithm based on kernel methods. This method uses the kernel distance as the distance of samples to be classified and the samples of training set. Experiments show that the algorithm is more effective than the traditional classification of K-NN. However, a major drawback of the algorithm is that the kernel parameter P is difficult to determine, mainly depend on one's experience to determine. By optimizing the kernel parameters, we can access to the best of the space conversion, to obtain the best classification results.

The kernel distance defined as follows:

$$KD(x, y) = K_p(x, x) - 2K_p(x, y) + K_p(y, y) \qquad (4)$$

2.4.4 Experiment reseach

In order to verify the validity of K-NN algorithm based on kernel methods, in Windows2000 Server environment, we selected 210 PE files of system as a security database, taken another 60 PE file of system as the samples of classification. According to the virus signature provided by Kingsoft, we obtained 80 virus files through various channels, of which 60 were randomly selected from the virus database, and 20 samples as the samples of classification.

The guiding ideology is to find out k nearest neighbor of x respectively in security database S and virus database V, then solving the center vector ys and yv respectively in S and V, calculating the distance of x , ys is $KD(x, ys)$, and x , yv is $KD(x, yv)$, if $KD(x, ys) < KD(x, yv)$, then $x \in S$, else $x \in V$. Table 1 lists the results of the experiment.

TABLE 1 THE CLASSIFY RESULT OF SYSTEM FILES AND VIRUS FILES

arithmetic	correct classification rate	
	The system files	*The virus files*
K-NN	***75***	***55***

K-NN based on kernel	*88.3*	*75*

2.4.5 The disposal of "suspicious file set"

If a process is suspicious, we can pick-up the eigenvector of it according to the above method. We regard "suspicious file set" as test subset KT, "security file set" and "virus set" as reference subset KR. In reference subset KR, we can get k nearest neighbors relatively fast by spatial sphere algorithm. Using kernel K-NN algorithm, we can tell whether the suspicious process is harmful. If it is harmful, the monitor process adds its eigenvector to "virus set", and then kills it, otherwise if it is safe, the monitor process adds its eigenvector to "security file set", and let it run.

3. CONCLUSION

This paper proposed the active defence technology against virus based on k-nearest neighbor algorithm of kenel. The system can not only judge security process efficiently, but also can detect unknown virus more accurately. The system can judge whether a process is harmful, and can store the eigenvector of suspicious process to "security file set" or "virus set" automatically, so the users can use the system easily.

REFERENCES

[1] DENG Lujuan, LIU Tao, GAN Yong, XIONG Kun. Active Defence Technology with Virus Based on Differentiation and Hiding Process [J]. Computer Engineering, March 2007. Vol.33 No.5.117-119. (in Chinese)

[2] WANG Haifeng,XIA Honglei,SUN Bing.The Study of Anti -Virus Engine Base On the Analysis Of Virus Behaviors[J]. Computer Systems Applications, 2006.No. 5.29-31. (in Chinese)

[3] ZHANG Boyun,YIN Jianping,ZHANG Bingxing,HAO Jingbo.Unknown Computer Virus Detection Based on K-Nearest Neighbor Algorithm[J]. Computer Engineering and Applications, 2005. No.6. 7-10. (in Chinese)

[4] XIE Jinjing,ZHANG Yibin. Computer Virus Detection Based on Improved K-Nearest Neighbor Algorithm[J]. Modern Electronics Technique,2007.No.3.51-53. (in Chinese)

[5] WANG Yejun,NI Xizhen,WEN Weiping, JIANG Jianchun. Research on Principle and Defense of Buffer Overflow Attacks [J]. Application Research of Computers,2005.No.10.101-104. (in Chinese)

[6] M. Kwiatkowska, V. Sassone, "Science for Global Ubiquitous Computing," Grand Challenges in Computing (Research), edited by T. Hoare and R. Milner, 2004.

[7] S. W. Smith, Trusted Computing Platforms: Design and Applications,Springer, New York, 2005.

[8] WEI Wei , ZHAN G Li-yan , ZHOU Lai-shui, A Spatial Sphere Algorithm for Searching K-Nearest Neighbors of Massive Scattered Points[J]. ACTA AERONAU TICA ET ASTRONAU TICA SINICA, Sept . 2006. Vol127 No15.944-948. (in Chinese)

[9] RAO Xian , YAN G Shao-quan , WEI Qing , DON G Chun-xi. Kernel nearest neighbor rule and its appl ication in intrusion detection. Systems Engineering and Elect ronics. Mar . 2007. Vol . 29 No. 3.470-471. (in Chinese)

[10] YU Xue-dou, Research on Active Defence Technology with Virus Based on Improved K-Nearest Neighbor Algorithm, ITESS'2008.