Research Journal of Applied Sciences, Engineering and Technology 4(1): 82-85, 2012
ISSN: 2040-7467

Submitted: September 23, 2011 Accepted: October 24, 2011 Published: January 01, 2012

Unknown Malicious Executables Detection Based on Immune Principles

[1,2]Jinquan Zeng, [3]Caiming Liu, [4]Jianbin Hu and [5]Yu Zhang
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]Sichuan Communication Research Planning and Designing Co., ltd, Chengdu 610041, China
[3]Laboratory of Intelligent Information Processing and Application, Leshan Normal University, Leshan 614004, China
[4]School of Electronics and Information, Nantong University, Nantong, 226019, China
[5]College of Information Science and Technology, Hainan Normal University, 571158 Haikou, China

Abstract: Detecting unknown malicious executables is a challenging task. Traditional anti-virus systems use signatures to detect malicious executables. However, the method cannot detect unseen instances or variants. Inspired by biological immune systems, an immune-based approach for detection of unknown malicious executables is proposed in this paper, which is referred to MEDMI. The approach can use the benign executables to be the training set for building the profile of the system and then generates detectors to detect malicious executables. The experiments comparing with different detection methods show that the approach provides an effective novel solution to detect malicious executables.

Key words: Anomaly detection, artificial immune system, malicious executables

INTRODUCTION

The computer malicious executable code has been with us for a quite long time. With the fast development of Internet, security threats of malicious executable code are getting more serious. Staniford introduces a worm that can spend the whole Internet within 30 seconds (Staniford *et al*., 2002). So how to detect malicious executables, specially unknown malicious executables, has become one of the prime research interests in the field of information security (Staniford *et al*., 2002; Hassan *et al*., 2011). Current anti-virus systems with a large number of virus signatures can only detect known viruses and cannot detect unknown viruses and the variants of known viruses (White, 1998). In order to detect new or unknown malicious executables, some researchers begin to investigate learning methods. Early, Lo *et al* (1995) proposed the filter for the viruses that can escape from signature-based methods; however, no experiment was conducted to validate the method. Tesauro *et al*. (1996) investigated the neural network for detecting boot-sector viruses and incorporated it into IBM Anti-virus software. This method can efficiently detect boot-sector viruses, however, not other viruses.

The problems found in computer security systems are quite similar to the ones encountered in Biological Immune Systems (BIS). BIS has successfully solved the problem of unknown virus detection (Li, 2004). Therefore, Artificial Immune System (AIS) (De Castro *et al*., 2003) is considered as a new way to defeat fast-proliferating malicious executables. In order to detect new or unknown malicious executables, this paper presents an immune-based model to detect unknown malicious executables, which is referred to MEDMI. Experimental results with these executables show that MEDMI has better detecting ability than that of the previous techniques.

MODEL THEORIES

Problem definitions: Given problem domain Ω, where $\Omega = \bigcup_{i=1}^{\infty} \{0,1\}^i$, i is a natural number. The executables are denoted as $E \subset \Omega$, and are divided into two set: B and M, such that: $B \cap M = \Phi, B \cup M = E$, where B is the benign executable set, and M is the malicious executable set which is infected by computer viruses, respectively. The task of malicious executable detection system is to classify an input as either benign or malicious. The state of an executable can be represented by a vector:

$$x^i = (x^i_{l_1}, x^i_{l_2}, \ldots, x^i_{l_n})$$

Corresponding Author: Jinquan Zeng, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

where, $0 \le x_{ij}^{i} \le 1$, j = 1, ..., n, and so the state space of the executables, denoted by U = [0,1]n, is a n-dimensional space. The state space of the executables can be divided into two set: Vb and Vm, such that:

$$V_b \cap V_m = \Phi, V_b \cup V_m = U$$

where, Vb is the benign executable vector set, and Vm is the malicious executable vector set, respectively. Malicious executable detection is defined as: given the set of benign executables B , where $B' \subset B$, builds a benign executable space characteristic function f:[0,1]n→{0,1}. Given the state of an executable, the function f is able to distinguish between benign executable and malicious executable.

Model definitions: Define the binary strings extracted from benign executables as antibody gene, and let Agdl devote the antibody gene set given by:

$$Agd_l = \left\{ad \middle| ad \in D_l, |ad| = l, l \in N\right\}$$

where l is the length of antibody gene (the number of bytes), N is the natural number and Dl is the binary strings extracted from the benign executables. Dl is described by:

$$D_l = \bigcup_{b \in Bl \in N, i=0} \bigcup^{|b|} \left\{ f_e(b,i,l) \right\}$$

where the function fe(b,i,l) extracts the binary string from the benign executable b (b∈B), i is the extracted position and l is the number of extracted bytes, respectively. Let Agd denote the antibody gene library given by:

$$Agd = Agd_{l_1} \cup Agd_{l_2} \cup, ..., \cup Agd_{l_n}$$

where, $l_i \in N, i = 1...n$ is the length of antibody gene and N is the natural number. The antibody gene library is made up of variable-length antibody genes, and the antibody gene library is used to extract the characteristics of the executables.

Antigens are defined as the executables, simulating the antigen presenting cells in BIS, and the characteristics of the executables are extracted by the antibody gene library Agd. Let C (C⊂U) devote the set of executable characteristics given by:

$$C = \left\{ \begin{array}{l} C = \left\langle x_{l_1}, x_{l_2}, ... x_{l_n} \right\rangle \middle| 0 \le x_{l_i} \le 1, e \in E, x_{l_i} \\ = f_c\left(e, Agd_{l_i}\right), i = 1, ..., n \end{array} \right\} \quad (1)$$

where, x_{li}, i = 1, ..., n is the characteristic of the executable e(e∈E) extracted by the antibody gene set Agd_{li}, and n is the dimension; the function $f_c(e, Agd_{li})$ counts the characteristic of the executable e (e∈E) in the antibody gene set Agd_{li}, described by the equation:

$$f_c\left(e, Agd_{l_i}\right) = \frac{\left| Agd_{l_i} \cap \left\{ \bigcup_{l_i \in N, j=0}^{|e|} \left\{ f_e(e, j, l_i) \right\} \right\} \right|}{\left| \bigcup_{l_i \in N, j=0}^{|e|} \left\{ f_e(e, j, l_i) \right\} \right|} \quad (2)$$

Let S denote the self set given by:

$$S = \left\{ s = \left\langle ch, rd \right\rangle \middle| ch \in C_b, rd \in R \right\}$$

where ch is the characteristics of the benign executables, Cb is the set of the characteristics of the benign executables, rd is the self radius of self elements, and R is the real number, respectively. The self radius of self element specifies the capability of its generalization (the elements within the self radius of the self element are considered as self elements). The bigger the self radius of the self element is, the more generalization the self element is. The big self radius of self element can decrease the number of self elements used to train detectors and false positive rate. Furthermore, the characteristics extracted from the benign executables are used to build the profile of the benign executables, and then the detectors can be generated to cover the space of the malicious executables. Let D denote the detector set given by:

$$D = \left\{ d = \left\langle ch, rd \right\rangle \middle| ch \in U, rd \in R \right\}$$

where, ch is the characteristics of the detectors, rd is the detection radius of the detectors, and R is the real number, respectively. D is subdivided into immature and mature detectors. Immature detectors are newly generated ones given by:

$$I = \left\{ \left\langle x_{l_1}, x_{l_2}, ..., x_{l_n} \right\rangle \middle| 0 \le x_{l_i} \le 1, i = 1, ..., n \right\}$$

Mature detectors are the ones that are tolerant to S given by:

$$M = \left\{ \begin{array}{l} x \middle| \forall s \in S, f_d(s,x) \rangle s.rd, \exists s' \in S, \forall s'' \in S, f_d(s',x) \\ < f_d(s'',x), x.rd = fd(s',x) - s'.rd \end{array} \right\} \quad (3)$$

Table 1: The comparison for detection performance of Schultz (Schultz *et al*., 2001) and MEDMI

	TP	TN	FP	FN	Detection	False positive rate	Overall accuracy
Signature Method[12]							
Bytes	1102	1000	0	2163	33.75%	0%	49.28%
RIPPER [12]							
DLLs used	22	187	19	16	57.89%	9.22%	83.2%
DLL function calls	27	190	16	11	71 .05%	7.77%	89.36%
DLLs with counted function calls	20	195	11	18	52.63%	5.34%	89.07%
Naïve Bayes[12]							
Strings	3176	960	41	89	97.43%	3.80%	96.96%
Multi-Naïve Bayes[120]	3191	940	61	74	97.76%	6.01%	96.88%
Bytes							
MEDMI	4135	968	24	57	98.64%	2.42%	98.44%

Furthermore, the detection radius of the detectors is decided by the nearest self element in S, and so the detector does not detect self elements and decrease the false-positive rate. Fd (x,y) is the Euclidean distance between x and y given by:

$$f_d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (4)$$

Immune surveillance: After the characteristics c (c∈C) of an executable e (e∈E) is extracted, its characteristics are presented to the detectors for detecting and the detecting process is given by:

$$fdetect(c) = \begin{cases} 0, \ iff \ \forall m \in M \wedge f_d(c,m) \rangle m.ra \\ 1, \ iff \ \exists m \in M \wedge f_d(c,m) \le m.ra \end{cases} \quad (5)$$

If the executable lies within the detection radius of a detector, the function fdetect(c) returns 1 and then the executable is malicious. Otherwise, the function fdetect(c) returns 0 and then the executable is benign.

SIMULATIONS AND EXPERIMENTAL RESULTS

Similar to the method gathering data in (Schultz *et al*., 2001), we gathered benign executables from a freshly installed Windows XP running MSOffice 2000 and Visual C++. The number of the benign executables was 992. The number of the malicious executables was 4192. There were no duplicate programs in the data set and every example in the set was labeled either malicious or benign by the commercial virus scanner. All labels were assumed to be correct.

To evaluate our method, we compared MEDMI with the methods used by Schultz (Schultz *et al*., 2001), including signature method, RIPPER, Naive Bayes and Multi-Naive Bayes. We compared these quantities: TP (True Positive) TN (True Negative) FP (False Positive) FN (False Negative) the Detection Rate (TP/(TP+FN)) False Positive Rate (FP/(FP+TN)) and Overall Accuracy ((TP+TN)/(TP+TN+FP+FN)). Table 1 shows the comparison for detection performance, where self radius is 0.001 and the number of detectors is 1050 in MEDMI. Table 1 shows the signature-based method has the worst true-positive rate, the learning-based methods have better detection performance than that of signature-based methods, and Na e Bayes and Multi-Na e Bayes have almost same overall accuracy; our proposed approach, MEDMI, has higher detection performance than that of Na e Bayes and Multi-Na e Bayes, for example, the true-positive and false-positive rates of Multi-Na e Bayes are 97.76 and 6.01%, however, the true-positive and false-positive rates of MEDMI are 98.64% and 2.42%. The reason is that the self radius increases the generalization capability of benign executables and the evaluative antibody gene library can avoid the new benign executables being detected by detectors.

CONCLUSION

Traditional anti-virus systems can not detect unseen malicious executables or variants and the previous learning-based models or methods, such as Na e Bayes, lack the ability of self-adaptation, have a high false-positive and false-negative rate, therefore, and have limited applications. In this paper, an immune-based dynamic model for unknown malicious executables is built. Experiments with a large number of malicious executables demonstrate that MEDMI is an efficient solution to unknown malicious executables detection and offers the characteristics of high-true positive rate, low false-positive rate.

ACKNOWLEDGMENT

This study is supported by special technology development fund for research institutes of the Ministry of Science and Technology of China (2009EG126226, 2010EG126236, and 2011EG126038), China Postdoctoral Science Foundation (20100480074) and NSFC (61003142).

REFERENCES

De Castro, L.N. and J.I. Timmis, 2003. Artificial immune systems as a novel soft computing paradigm. Soft Comp. J., 7(8): 526-544.

Hassan, K., M. Fauzan and A.K. Syed, 2011. Determining malicious executable distinguishing attributes and low-complexity detection. J. Comp. Virol., 7(2): 95-105.

Li, T., 2004. Computer Immunology. Publishing House of Electronics Industry, Beijing.

Lo, R.W., K.N. Levitt and R.A. Olsson, 1995. A malicious code filter. Comp. Security, 14(6): 541-566.

Schultz, M.G., E. Eskin, F. Zadok and S.J. Stolfo, 2001. Data mining methods for detection of new malicious executables. Proceedings of IEEE Symposium on Security and Privacy, Oakland, CA, pp: 1207-1217.

Staniford, S., V. Paxson and N. Weaver, 2002. How to own the internet in your spare time [C]. In: Proceedings of the 11th USENIX Security Symposium San Francisco Marriott, August, pp: 149-167.