Biologically Inspired Computer Virus Detection System*

Hyungjoon Lee[1], Wonil Kim[2,**], and Manpyo Hong[1]

[1]Digital Vaccine Lab, Graduated School of Information and Communication
Ajou University, Suwon, Republic of Korea
{prime, mphong}@ajou.ac.kr
[2]College of Electronics and Information Engineering
Sejong University, Seoul, Republic of Korea
wikim@sejong.ac.kr

Abstract. There have been many researches in Computer Science that their fundamental ideas were based on Biology. Genetic algorithm and neural network are best-known paradigms in this category. Recently, many ideas from immune system have been used in detecting computer virus and worm. Since the first computer virus has been found, scanning detection has been used as a primarily method in virus detection systems. As computer viruses and worms become more complex and sophisticated, the scanning detection method is no longer able to detect various forms of viruses and worms effectively. Many anti-virus researchers proposed various detection methods including artificial immune system to cope with these characteristics of computer viruses and worms. This paper discusses the principle of artificial immune system and proposes artificial immune based virus detection system that can detect unknown viruses.

1 Introduction

Since the computer virus first appeared in 1981, it has been evolved continuously as computer environment such as operating system has advanced. The frequent evolution of computer virus has lead anti-virus researchers to continuous development of new detection method. But most anti-virus systems are still based on scanning detection using signature, since other mechanisms have high false positive rate or detection speed problem in real situations [11]. Recently, several anti-virus researches focuses on biologically inspired system such as negative selection method as novel virus detection system [6,11].

In this paper, we propose a new computer virus detection approach that employs the artificial immune system. Artificial immune system is based on human immune system. The human immune system recognizes, memorizes, and responses to virus intrusion. Anomaly detection and adaptability are important properties of artificial

* This work was supported by grant No. (R05-2003-000-11235-0) from the Basic Research Program of the Korea Science & Engineering Foundation
** Author for correspondence, +82-2-3408-3975

A.J. Ijspeert et al. (Eds.): BioADIT 2004, LNCS 3141, pp. 153–165, 2004.

immune system. The proposed artificial immune based system exploits these properties and detects unknown computer viruses.

In the next chapter, we survey on human immune system. Artificial Immune system will be discussed in detail in Chapter 3 and Chapter 4. In Chapter 5, the proposed artificial immune based virus detection system will be described. Simulation results and the direction of future research will be discussed in Chapter 6 and Chapter 7 respectively.

2 Human Immune System

Human immune system (HIS) protects body from pathogen like virus using distributed immune cells. HIS is a complex system that consists of cells, molecules and organs. Elements of HIS cooperate with each other to identify "self" (own cells) and "non-self" (foreign cells or pathogens). When pathogens enter the body, they are detected and eliminated by HIS. HIS also remembers each infection type and pattern. If the same pathogen enters the body again, HIS copes with it more effectively than the first time. These processes are performed in distributed environment. There are two inter-related systems in the HIS: the innate immune system and the adaptive immune system [23].

The body is born with the innate immune system. The innate immune system has the ability to recognize some patterns of particular pathogens. The innate immune system is based on a set of receptors known as Pattern Recognition Receptors (PRRs). One receptor matches with particular molecular pattern of pathogen, called Pathogen Associated Molecular Patterns (PAMPs). The PAMPs originate from only foreign pathogens, not the body cells. Therefore, recognition by the PRRs means that there are some pathogens. This mechanism is based on the boost of adaptive immune system, and it gives a capability of distinguishing between self and non-self.

When innate immune system recognizes certain pathogens, it bursts stimulatory signal that will lead to T cell activation. T cell has receptors called antibody, and activating T cells is the start of the adaptive immune system. Activated T cell stimulates B cell, and then B cell proliferates and differentiates into non-dividing terminal cell and antibodies. Finally, antibodies neutralize recognized pathogens. While activated B cells secrete antibodies, T cells do not secrete antibodies, rather T cells play a central role in the control of the B cell response in adaptive immune system.

Negative selection is another element of adaptive immune recognition that does not need the innate immune recognition. Negative selection is an activity selecting receptor that does not detect self. Receptors have distinct protein structure, and protein structures are formed by a pseudo-random genetic process called negative selection. According to the molecular structure of a receptor, the receptor detects a particular cell that was indicated by its molecular structure called antigens. If pathogen intrudes human body, immune cells that have receptor matched for them are able to detect and neutralize them. In this paper, we will propose artificial immune system based on negative selection algorithms. The system generates detectors that never matched with normal programs. Detectors are classified into virus detectors and other things. Classification of virus detector based on clonal selection of HIS.

Clonal selection supports the classification ability of HIS. Although receptors of immune cells are generated randomly, the adaptive immune system maintains useful – detectable current intrusion – immune cells among many of them. When receptors of immune cells and antigens of pathogens are matched, T cells produce clones. This process results expansion of the corresponding T cells, so that adaptive immune system responds rapidly to the same pathogens. The adaptive immune system generates immune cells randomly, and then it selects detectable one. Expanding the detectable one, human immune system response rapidly. This process is called clonal selection.

HIS is a very complex system for protecting human body against harmful pathogen. The main purpose of HIS is to classify all cells into self or non-self. To do this, HIS learns and memorize useful antibody through negative selection and clonal selection in distributed environment. These mechanisms give inspiration in several computer science areas such as pattern classification, distributed architecture solution, and anomaly detection. In next chapter, we will enumerate several biological immune based systems that are used in computer security. They are called artificial immune system.

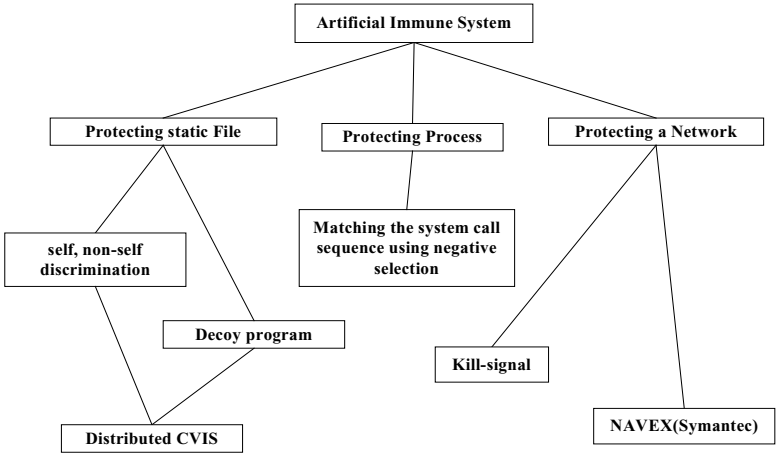

Fig. 1. Taxonomy of artificial immune system

3 Artificial Immune System in Computer Security

Artificial immune system is one of the biological inspired systems such as neural network and genetic algorithm. Artificial immune system is based on several capabilities of biological immune system. In computer security, artificial immune system should have anomaly detection capability to defend unknown intrusion. Adaptability is also a necessary property of artificial immune system that can learn unknown intrusion and respond to learned intrusions quickly. Other properties such as distributability, autonomy, diversity and disposability are also required to flexibility and stability for artificial immune system [14].

Artificial immune system can be divided into three categories according to which computer component will be identified as a protected cell as in Figure 1.

First, protecting static file category assumes a file including program code as a cell. Forrest proposed a method for change detection using negative selection algorithm [6]. Forrest's artificial immune system randomly generates check code called detector, and eliminates ones that can recognize self (benign program). This process is called negative selection algorithm. All programs or data matched with detectors are indicated as non-self (malicious code). Jeffrey O. Kephart proposed an artificial immune model that consists of monitoring, scanning and biological immunology [11]. In his model, known viruses are detected by scanner based on weak signature, and unknown viruses are collected and analyzed using decoy program. The decoy program is dummy program that is not changeable. If decoy program is modified, it is virus infection. The decoy program is a trap to collect unknown viruses. Distributed Computer Virus Immune System (DCVIS) combined negative selection and decoy program. It uses decoy program to detect the unknown virus and employs negative selection to identify detected unknown virus [16].

Second, protecting process category assumes a process as a cell. Matching the system call sequence is proposed by Forrest [10]. The mechanism of this is the same as self-nonself discrimination using negative selection. It is different that this mechanism matches detector with system call sequences, not with program codes.

Third, protecting a network category assumes a host as a cell. Kill-signal mechanism is proposed by Kephart, and NAVEX is immune system that is proposed by Symantec [11,17]. Both immune systems propagate signatures of detected viruses to protect hosts connected via network like biological immune system. In biological immune system, when an immune cell detects a virus, the cell spread chemical signal in order to duplicate the same immune cells. In network intrusion detection system, Williams et al. proposed Computer Defense Immune System (CDIS). It is proposed to protect local network using artificial immune system. CDIS is a distributed system with redundant links and decentralized control. These features provide several capabilities such as fault tolerance and no-single-point-failure that can be found in biological immune system [22].

The proposed artificial immune based virus detection system is included in the first category. It extracts suspicious code detector from all the suspicious codes based on negative selection mechanism of biological immune system. Then, it distinguishes virus detector from extracted detectors. Proposed detection system is described in detail in Chapter 5.

4 Computer Viruses Detection and Artificial Immune System

Immunologists have traditionally described the human immune system as the problem of distinguishing "self" from dangerous "non-self" and eliminating dangerous nonself. Similarly, the computer security can be viewed as the problem of distinguishing benign program from malicious program such as computer virus or worm. Eugene H. Spafford describes a virus from a biological point of view, and shows a several characteristic of virus as artificial life [1]. Since then, many anti-virus researchers have re-

searched on artificial immune system to protect a system from the intrusion including viruses and worms.

The characteristics of current viruses are polymorphic and fast spreading. These characteristics impose several problems on current anti-virus system. First, it is hard to extract virus signature from polymorphic viruses, since the virus changes its structure in every replication. Second, although we extract the virus signature, if viruses spread very quickly using network environment, they already damage many computer resources. Therefore, we need innovative detection mechanisms instead of scanning method based on virus signature. In this chapter, current virus detection methods and some of the artificial immune based virus detection system are briefly discussed.

Current commercial anti-virus systems are virus scanners. Virus scanners look for virus signature in memory or in program files. Once suspicious code is found, scanners alert to the user that the system is infected. Virus scanner is very fast, but managing huge size of signatures for virus scanning is getting more and more difficult. And also, virus scanner cannot detect unknown and polymorphic virus.

Many anti-virus system use heuristic algorithms in order to detect polymorphic viruses. Since most polymorphic virus technique is insertion and mutation, heuristic algorithm detects polymorphic virus by using frequency analysis of particular strings or codes. Although anti-virus system has heuristic algorithm that can detect some polymorphic viruses, it requires specific heuristic algorithm for each polymorphic virus. Moreover, recent complex polymorphic virus – encrypted polymorphic virus – cannot be detected with scanners equipped heuristic algorithm. Contrary to insertion and mutation, encryption technique generates new code that does not match particular code at all [3][4].

Change detector is a detection method using integrity check. Change detector scans the executable program files and records information about the files before the system initiates. The change detector periodically scans the program files to check the integrity of files. If there is difference between recorded and current check-codes, a virus could have caused the changes.

Recent change detector using negative selection algorithm proposed by Stephanie Forrest is different from the conventional ones such as MD5, CRC [6]. Forrest's model maintains the signature that does not match any of the protected files. This signature is called "detector". The system monitors the protected files by comparing them with the detectors. If a detector is matched any file, a change is known to have occurred. The algorithm for generating the detector is computationally expensive, but checking is cheap. It would be difficult to modify file and then alter the corresponding detector.

Related research using simple and efficient definition for abnormal behavior was proposed by Forrest [10]. Forrest collects normal behavior as short sequences of system call. Next, abnormal behaviors are generated using negative selection algorithm. The negative selection algorithm generates detectors that recognize abnormal behavior. In this case, the detector is a set of system call sequences of abnormal program. This approach could detect malicious program if the system call sequence of monitored program are the same as that of detector.

The most important characteristics of current computer viruses are polymorphic and spread quickly, whereas existing anti-virus systems are not flexible enough to cope with the property of recent computer virus. The negative selection algorithm that

supports the anomaly detection and distributed environment can be a good candidate for designing effective polymorphic virus detection system. It also can protect a network from fast-spreading virus.

5 Proposed Artificial Immune Based Virus Detection System

Negative selection algorithm and decoy programs are important methods in artificial immune system. Negative selection algorithm is inspired by genetic random-process. But this mechanism has weakness that come from the difference between human body and computer system. State of human body is stable, whereas that of computer system is not. Negative selection algorithm is suitable in stable system. Idea of decoy program comes from the distributability and diversity of immune system. In the case of decoy program, there is no guarantee that a virus will attack a decoy program. The proposed Artificial Immune based Virus Detection System (AIVDS) compromises this weakness, and is suitable for dynamically changing environment such as computer system.

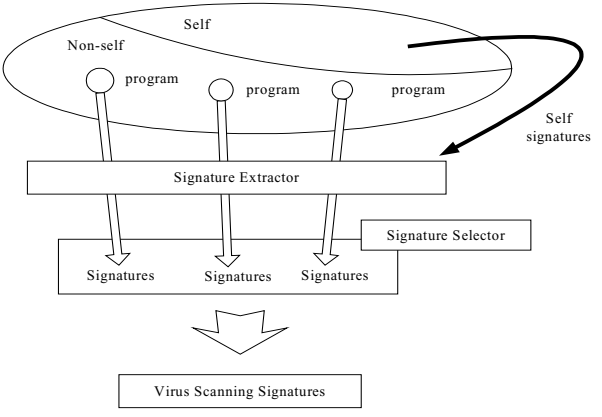

Fig. 2. Structure of proposed virus detection system

VDS is a signature learning system to detect unknown viruses. AIVDS classifies incoming or changed (patched) programs into legitimate programs and viruses. The process of AIVDS is consists of the following three steps. In the first step, AIVDS assumes that all existing programs are legitimate. In the next step, all incoming and changed programs are classified into suspicious programs. Finally, AIVDS selects virus programs from these suspicious programs using detection method based on virus behavior. Hereafter, we refer to legitimate program as self and suspicious program as non-self.

AIVDS consists of signature extractor and signature selector as in Figure 2. Signature extractor produces signatures of non-self. The main operation of signature extractor is selecting bit strings that are not matched with any self code at the same position. Therefore, signature extractor produces signature of non-self that never

matched with any self. Produced non-self signatures are collected and analyzed by signature selector.

Signature selector compares the similarity of non-self signature with each others. Since viruses tend to infect other programs, if the same non-self signatures are appeared frequently, signature selector classifies those into signatures of viruses, and these signatures are used for virus scanner. Notice that these non-self signatures never match with any self program. In the case of less frequent signatures, it is assumed as signature of self. We classify the program that contains any one of these self signatures as self. This process provides the adaptable capacity to the proposed AIVDS.

The detailed process of each steps are discussed in the following 3 sections.

5.1 Signature Representation

The AIVDS generates signatures to recognize self and non-self program. The size of signature extraction region, from which we analyze for signature extraction, is always constant. Since most infected file executes virus code at first, most entry pointer of infected program indicates virus code. Therefore, we decide that starting point of signature extraction region is entry point of program. Signature consists of several pairs of offset and bit string. Number of pairs, offset and bit string are selected by signature extractor.

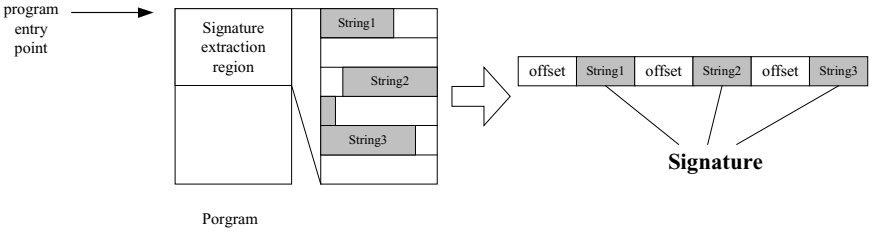

Fig. 3. Signature representation

5.2 Signature Extractor

Signature extractor produces non-self program signature that never matches with any self-program signature. This process consists of the following steps.

- Step 1: Divide a signature extraction region into several same sized comparison units.
- Step 2: Compare a signature extraction region of non-self program with each one of all self-programs.
- Step 3: If two comparison units on same position are same, discard the comparison unit in signature extraction region of non-self program. Continue this process on each signature extraction region of all self-programs.

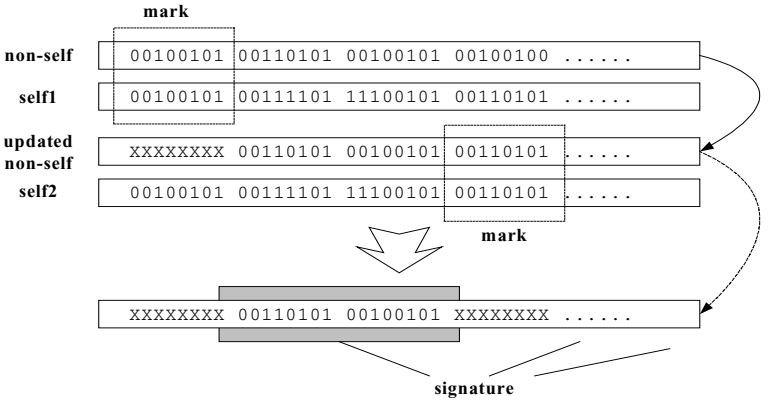

Fig. 4. Generating signature of non-self program

In Figure 4, comparison unit is 1 byte. Non-self is compared with self1 and self2. When two comparisons are finished, first unit and fourth unit are marked. After all the comparisons with other self-programs are done, the signature of non-self consists of remained unmatched units. Since all of the comparison units that are the same with self's are marked as useless, generated signature never matches with any self-programs.

5.3 Signature Selector

After we obtain signatures of non-self programs, we need to classify non-self program into virus or normal program. According to behavior of viruses, we can decide whether the signature indicates virus or not. Viruses tend to infect other programs. If virus infects other programs, the signature extractor generates signatures from same virus code because infected program is changed by one virus. Therefore, checking frequency of occurrence of the same signature is the same as checking the spread of the same virus code.

Signature selector calculates the similarity values of non-self signatures, as shown in Figure 5. Comparison factors are bit sequence and offset of comparison unit in signature extraction region. If two factors of comparison units are equal, similarity function adds one to similarity value. When consecutive comparison units are equal, similarity value is higher. For example, if two compare units are equal and adjacent, similarity value is 3 (11_2). In other words, when n consecutive comparison units are equal, similarity value is 2^n-1.

Note that signatures of the same programs are more similar than signatures of different programs. Therefore, similarity value between signatures of same program codes is higher than the other. Threshold values for classifying the same and different programs are determined by analyzing similarity values of entire non-self programs. Similarity value of the same programs is relatively high.

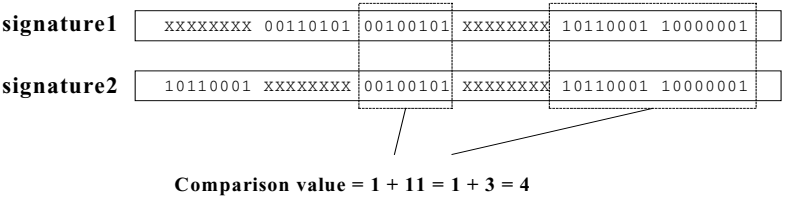

Fig. 5. Similarity between two signatures

6 Simulation

AIVDS extracts signatures of non-self programs based on self-programs. Signature size and similarity value is important factor in AIVDS. As the number of self-programs that has same comparison units at same position is increased, size of a signature of the non-self program is decreased. In the worst case, AIVDS cannot generate the signature of particular program, because every comparison units are marked as useless. The similarity value is used for classifying virus detector from suspicious signatures. In this chapter, we will show simulation result from different signature extraction region and comparison unit. The simulation is processed on several parameters as in Table. 1.

Table 1. Simulation parameters

Parameters	Variables
The number of self-programs	1385 execution files
The number of non-self programs	160 execution files (3 virus infected files)
Signature extraction region size	500Byte, 1Kbyte, 5Kbyte, 10Kbyte
Comparison unit size	1Byte, 2Byte, 3Byte

In Figure 6, graphs show the relation between signature size and two parameters; signature extraction region and comparison unit. The more signature extraction region and comparison unit increases, the more signature size increases. Larger signature includes richer information about related non-self program. But, when signature is used for scanning viruses, small signature is effective. Signatures that are larger than 1KB are not feasible for a virus-scanning signature. Moreover, the percentage of files that has zero signature is independent with size of signature extraction region and comparison unit. The number of files with zero signature size is almost 14 (8.75%) in all case. Therefore, we chose two effective sizes of signature extraction region; 1Kbyte and 500 Byte. We simulated comparison for signature selection about the extracted signatures of non-self programs on these two parameters.

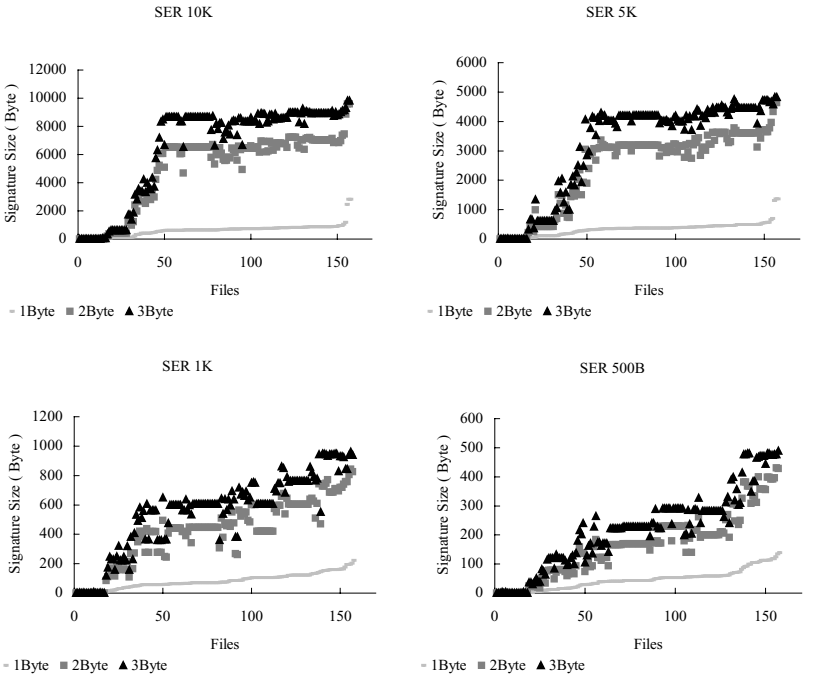

Fig. 6. Simulating using various sizes of SER: Graphs show signature size of Signature Extraction Region (SER) 10Kbyte, 5Kbyte, 1Kbyte, and 500Byte. Each graph has three parameters of comparison unit 1Byte, 2Byte, and 3Byte.

Similarity values between each signature of non-self programs are shown in Figure 7. When signature extraction region is 1Kbyte and comparison unit is 1Byte, 88% of signatures of non-self programs have similarity value zero. When signature extraction region is 500Byte and comparison unit is 1Byte, signatures who similarity value is zero are 92% in entire signatures of non-self programs. Since virus infected file is 1.875%, ideal percentage of signatures that similarity value is zero should be 98.125%. Then, we can classify non-self signatures into virus signatures, whose similarity value is greater than zero. But, the percentage of non-zero signatures is 6% even though the extraction region size is 1Kbyte and comparison unit size is 3 Byte. We need to determine threshold value to classify non-self programs into normal programs and viruses.

When signature extraction region is the same size, the more comparison unit is increase, the more similarity value is increase rapidly, because consecutive extracted signature is larger. When larger comparison unit is used, the gap of similarity value is larger. When signature extraction region is 1Kbyte and comparison unit is 3Byte, we can find easily threshold of similarity value 1.E+08 to classify signatures of non-self programs into virus, because ideal percentage of signatures of normal programs is 98.125%. When threshold of similarity value is 1.E+08, three signatures are selected by signature selector. Virus scanner using these signatures can detect virus-infected files.

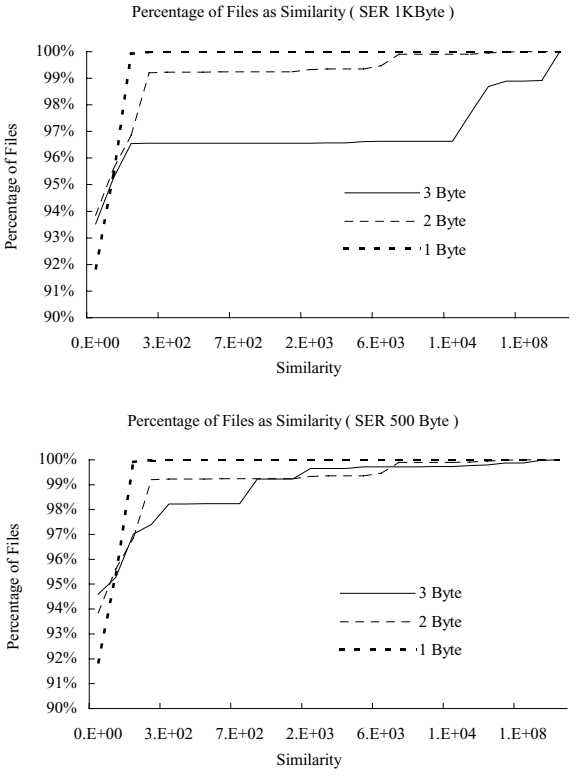

Fig. 7. Percentage of files as similarity: In the case of SER 1Kbyte and 500Byte, Percentage of files is showed. Each graph has three parameters of comparison unit 1Byte, 2Byte, and 3Byte.

Figure 8 shows the behavior of the synthetic virus. The sample of self-programs is the same as one of the previous simulation, and the parameters are selected in the general case; SER 1Kbyte, comparison unit 3Byte, and threshold value of similarity 1.E+08. Non-self program is inserted at each stage as in the previous simulation. The only difference is that the same synthetic virus inserted at every stage between 100 and 150 with the probability of 0.3. At the same time, inserted synthetic virus infects other self-programs at every stage with the probability of 0.3. During this period, the death rate of synthetic virus is 0.33. This figure shows that the number of similar – similarity value is higher than threshold value – signatures increase from stage 100 and decrease from stage 150. After stage 200, the number of similar signatures changes between 0 and 5, but it is very small compared with the period of virus propagation (between 100 and 200). Therefore, AIVDS is able to detect virus propagation behavior.

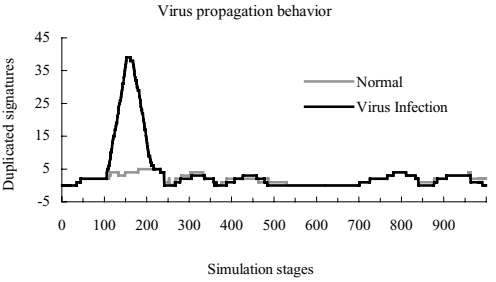

Fig. 8. Virus propagation behavior at the AIVDS. In case of SER 1Kbyte and comparison unit 3Byte, virus propagation started at the stage of 100, and stopped at the stage of 150.

7 Conclusion

Artificial immune system is based on distinguish "self" and "non-self" like biological immune system. But previous artificial immune system is not feasible for dynamic computer system, especially negative selection algorithm. In this paper, we proposed the Artificial Immune based Virus Detection System (AIVDS) that is feasible for dynamic computer system. AIVDS is a signature learning system to detect unknown virus. AIVDS produces signatures of non-self from suspicious program, and classify them into self-program and virus. Since virus tend to infect other programs, if similar non-self signatures are appeared frequently, AIVDS classifies those into signature of viruses for virus scanning. This process gives the adaptable capacity to AIVDS.

In the simulation of the proposed AIVDS with 1Kbyte of the Signature Extraction Region (SER) and 3Byte of the comparison unit, 94% of extracted signatures were completely different, in other words their similarity values are zero. The remaining 6% signatures including virus signatures had distinguished similarity values. Especially, 2% virus signatures had relatively high similarity values. The proposed AIVDS classifies suspicious non-self programs into normal programs and viral programs. Using threshold of similarity value, AIVDS can select virus signatures. Virus scanner using these signatures can detect virus-infected files correctly. It also detects the propagation behavior of virus.

References

[1] Eugene H. Spafford, "Computer Viruses as Artificial Life", in Artificial Life II, Addison-Wesley 1992.
[2] Richard Marko, "Heuristics: Retrospective and Future", Virus Bulletin Conference 2002
[3] Taras Malivanchuk, "The WIN32 worms: Classification and Possibility of Heuristic detection", Virus Bulletin Conference 2002
[4] Vesselin Bonchev, "Macro and Script Virus Polymorphism", Virus Bulletin Conference, 2002

[5] Gabor Szappanos, “Are There Any Polymorphic Macro Viruses at All?”, Virus Bulletin Conference, 2002
[6] Stephanie Forrest, “Self-Nonself Discrimination in a Computer”, IEEE Symposium on Research in Security and Privacy 1994
[7] J. P. Anderson, “Computer Security threat monitoring and surveillance”, Technical report, 1980
[8] Dorothy E. Denning , “An Intrusion-Detection Model”, 1986 IEEE
[9] David Wagner and Drew Dean, Intrusion Detection via Static Analysis, 2001 IEEE
[10] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, “A Sense of Self for Unix Processes”, 1996 IEEE
[11] Jeffrey O. Kephart, "A Biologically Inspired Immune System for Computers", High Integrity Computing Laboratory IBM Thomas J. Watson. Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems 1994.
[12] Kim, J., Bentley, P., 1999, "The Human Immune System and Network Intrusion Detection", submitted to EUFIT'99
[13] Digging For Worms, Fishing For Answers, 2002
[14] Dipankar Dasgupta and Nii Attoh-Okine, “Immunity-Based Systems : A survey”, IEEE International Conference 1997
[15] Anil Somayaji, Steven Hofmeyr and Stephanie Forrest, “Principles of a Computer Immune System”, 1997
[16] Robert E. Marmelstein, David A. Van Veldhuizen and Gray B. Lamont, “A Distributed Architecture for an Adaptive CVIS”,
[17] “The Digital Immune System”, Symantec technical report
[18] Stephanie Forrest, Anil Somayaji and David H. Ackley, “Building Diverse Computer Systems”, IEEE 1997
[19] Dipankar Dasgupta and Stephanie Forrest, “Novelty Detection in Time Series Data Using Ideas from Immunology”, International Conference on Intelligent System 1999
[20] Dipankar Dasgupta, “Artificial Neural Networks and Artificial Immune System : Similarities and Differences”, IEEE 1997
[21] Okamoto, T. and Ishida, Y., “Multiagent Approach Computer virus: An Immunity-Based System”, AROB 99
[22] P. D. Williams, K. P. Anchor, J. L. Bebo, G. H. Gunsch, and G. B. Lamont, “CDIS: Toward a computer immune system for detecting network intrusions,” in Proc. Fourth Int. Symp. Recent Advances in Intrusion Detection, Oct. 2001, pp. 117–133
[23] Leandro Nunes de Castro and Fernando Jose Von Zuben, *“Artificial Immune Systems: Part I – Basic Theory and Applications,”* Technical report, TR-DCA 01/99