Journal of Universal Computer Science, vol. 13, no. 9 (2007), 1354-1365
submitted: 12/6/06, accepted: 24/10/06, appeared: 28/9/07 © J.UCS

Computer Forensics System Based on Artificial Immune Systems

Jin Yang, Tao Li, Sunjun Liu, Tiefang Wang
Diangang Wang, Gang Liang
(School of Computer Science, Sichuan University, Chengdu, China
jinnyang@163.com, litao@scu.edu.cn, liusunjun@163.com
waltzjhrny@vip.sina.com, wangdg@vip.sina.com, gangliang56@163.com)

Abstract: The current computer forensics approaches mainly focus on the network actions capture and analysis the evidences after attacks, which always result in the static methods. Inspired by the theory of artificial immune systems (*AIS*), a novel model of Computer Forensics System is presented. The concepts and formal definitions of immune cells are given, and dynamically evaluative equations for self, antigen, immune tolerance, mature-lymphocyte lifecycle and immune memory are presented, and the hierarchical and distributed management framework of the proposed model are built. Furthermore, the idea of biology immunity is applied for enhancing the self-adapting and self-learning ability to adapt continuously variety environments. The experimental results show that the proposed model has the features of real-time processing, self-adaptively, thus providing a promising solution for computer forensics.

Key Words: Network security, Computer forensics, Artificial immune systems

Category: H.3.7, H.5.4

1 Introduction

Computer forensics is a new approach for the network security. The field of computer forensic science emerged as an opponent to the growth of computer crimes. Computer forensics is the application of computer investigation and analysis techniques in the interests of determining potential legal evidence [Osles 2001]. However, current solutions for computer forensics are mostly static methods [Moan 2004] [Bashaw 2003] [Srinivas 2003] [Reis 2002], which are only used to collect, analyze and extract evidences after intrusions. Most forensic systems are based on statistics trails or try to detect known attack patterns, deviations from normal behavior, or security policy abnormal. The methods of adaptive capture the potential sensitive traffic and real time analyses are seldom considered. Therefore, the current situation calls for an effective and adaptive analyzing system for computer forensics.

Artificial Immune Systems (*AIS*) is a now receiving more attention and is realized as a new research hotspot of biologically inspired computational intelligence approach after the genetic algorithms, neural networks and evolutionary computation in the research of Intelligent Systems [Castro 2003]. Burnet

proposed clone Selection Theory in 1958 [Burnet 1959]. Negative Selection Algorithm and the concept of computer immunity proposed by Forrest in 1994 [Forrest 1994b]. It is known that the Artificial immune system has lots of appealing features [Kim 1999] [Li 2004a] such as diversity, dynamic, parallel management, self-organization and self-adaptation that has been widely used in the fields such as [Li 2004b] [Forrest 1994a] data mining, network security, pattern recognition, learning and optimization etc.

Thus, we apply Immune theory as support and propose a promising solution for building computer forensics system. In this paper, the computer forensics models and the corresponding dynamic equations for self, antigen, dynamic computer forensics, immune tolerance, mature-lymphocyte lifecycle, and immune memory are introduced. We put forward a Computer Forensics System based on Artificial Immune Systems, i.e. the *CFSAIS*. Our experiment results show that it is a good solution for computer forensics. This paper is organized as follows: Section 2 briefly describes the system structure and presents the arithmetic of the Artificial Immune. The results from system testing are presented in Section 3, with particular investigation into the areas of efficiency and effectiveness. Finally, Sections 4 gives the summary and conclusions.

2 The Computer Forensics System Based on AIS

Computer forensic science is the science of acquiring, preserving, presenting data and analyzing information collected on networks. In this section we describe the architecture of *CFSAIS*. The complete system architecture is represented in Fig. 1 and we will show the description of the system functioning and cooperation scheme in detail.

There are many similarities between computer security system and biological immune system (*BIS*) [Li 2005c]. A biological immune system can produce antibodies to resist pathogens through B cells distributing all over the human body. And T cells can regulate the antibody concentration. Simulating biological immune system, we place a certain amount of immune cells (viz. *Sensor Detectors*) into the network, and perceive the surrounding environment. Distributed *Detectors* are deployed on the sensitive host that needs more security and protection in the network. These hosts are monitored and can be provided forensics analysis once there are some attacks on them. In other words, the *Sensor Detectors* (*SDs*) simulate the lymphocyte and used as a detector to recognize *nonself* antigens. As B-lymphocytes consist of mature and memory ones, the *SDs* are divided into mature and memory *SDs*. The memory *SDs* will match the anti-gens at first and eliminate *nonself* antigens. The memory *SDs* have an unlimited life-cycle except they match the newly created *selfs*. Obviously, a considerable number of memory *SDs* will be generated in the end. Mature cells either evolve into memory ones

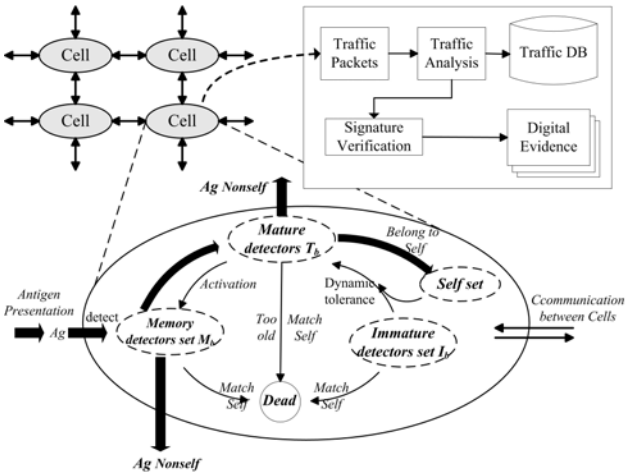

Figure 1: The Architecture of Computer Forensics System Based on AIS

or die before they exceed the lifecycle. As soon as the *SDs* detect an attack, the cells begin clone and generate a mass of similar cells in order to defend from fiercer network attacks and warn the dangerous level of the network. While the network danger become abating, the corresponding numbers of cell antibodies will decrease at the same time. The detectors' amount and type reflect the attack's intensity and type suffered by the network intrusion. In this model, the detectors can be categorized, accord-ing to the evolvement progress of the *SDs* themselves, into 3 types, viz. immature, mature and memory *SDs*. Fig.1 shows the cell structure and dynamic evolvement, and the relationship and the process of evolution of these *SDs* will be expatiated in detail in the following.

2.1 The Definition of Antigen, Antibody, Self and Non-self

Let $\Omega = \cup_{i=1}^{\infty}\{0,1\}^i$ be a set of binary strings. Let $R = \{\langle a, b\rangle | a \in D \wedge |a| = l \wedge b \in \Omega\}$, where $D = \{0,1\}^l$, l is a fixed natural number, and $|a|$ is the length of a. Given antigen set $Ag \subset R$, for $x \in Ag$ [Li 2005a]. $x.a$ is the antigenic determinant, $x.b$ is the original IP packets, which is the character of $x.b$ and consists of the source and destination IP addresses, port number, protocol type, flags, packet length, tcp/udp/icmp fields, etc. We define antigens (Ag) to be the features of network actions and services, given by:

$$Ag = \{ag | ag \in D\}. \tag{1}$$

The structure of an antibody is the same as that of an Antigen. For intrusion detection, the nonself set ($Nonself$) represents IP packets from a computer network attack, while the self set ($Self$) is normal sanctioned network service

transactions and nonmalicious background clutter. Set Ag contains two subsets, $Self \subseteq Ag$ and $nonself \subseteq Ag$ such that

$$Self \cup Nonself = Ag,\ Self \cap Nonself = \Phi. \tag{2}$$

For the convenience using the fields of a antigen x, a subscript operator "." is used to extract a specified field of x, where

$$x.fieldname = \texttt{the value of filed}\quad fieldname\, x. \tag{3}$$

In *CFSAIS*, all the detectors form a Set *Sensor Detector* called *SD*.

$$SD = \{\langle d, age, count\rangle | d \in D, age \in N, count \in N\}. \tag{4}$$

where d is the antibody gene that is used to match an antigen, age is the age of detector d, count (affinity) is the number of detector matched by antibody d, and N is the set of nature numbers. SD contains two subsets: mature and memory, respectively, the set Mat_{SD} and set Mem_{SD}. A mature SD is a SD that is tolerant to *self* but is not activated by antigens. A memory SD evolves from a mature one that matches enough antigens in its lifecycle [Li 2005b]. Therefore,

$$SD = Mat_{SD} \cup Mem_{SD},\ Mat_{SD} \cap Mem_{SD} = \phi. \tag{5}$$

$$Mat_{SD} = \{x | x \in SD, \forall y \in Self, (\langle x.d, y\rangle \notin Match \wedge x.count < \theta)\} \tag{6}$$

$$Mem_{SD} = \{x | x \in SD, \forall y \in Self, (\langle x.d, y\rangle \notin Match \wedge x.count \geq \theta)\} \tag{7}$$

where $\beta(> 0)$ represents the activation threshold. $Match$ is a match relation defined by

$$Match = \{\langle x, y\rangle x, y \in D, f_{match}(x, y) = 1\} \tag{8}$$

In the course, θ is the threshold of the affinity for the activated *detectors*. The affinity function $f_{match}(x, y)$ may be any kind of *Hamming*, *Manhattan*, *Euclidean*, and *r-continuous matching*, etc. In this model, we take *r-continuous matching* algorithm to compute the affinity of mature *detectors*. The matching functions utilize the following definitions:

$$f_{match}(x, y) = \begin{cases} 1 & \exists i, j, j - i \geq r \wedge 0 < i < j \leq l, x_i = y_j, \cdots, x_j = y_j \\ 0 & otherwise \end{cases} \tag{9}$$

The *r-continuous* matching is commonly used method for measuring the distance between bit strings with the goal of producing a better similarity coefficient.

Let $\Gamma \subset \{\langle t, x, y, s\rangle | t \in N, x \in Ag, y \in \Omega, s \in \Omega\}$ represent the digital evidences, where t is the evidence collecting time; x is the captured IP packets, $x.b$ is the original evidence (original IP packets), and $x.a$ is the evidence extracted

from $x.b$; y depicts the network environment in the host at time t, s denotes the digital signature of the evidence:$s = E_{k_{private}}(H(t + x + y))$, where E is the signature algorithm such as RSA, $k_{private}$ is a private key, H is a hash function, '+' is the operator for string connection. For $\forall \tau \in \Gamma$, the following equation (10) can verify evidence τ:

$$f_{verify}(\tau) = \begin{cases} 1 & D_{k_{public}}(\tau.s) = H(\tau.t + \tau.x + \tau.y) \\ 0 & otherwise \end{cases} \tag{10}$$

where $D_{k_{public}}(\tau.s)$ denotes decryption computing with the public key k_{public} and the corresponding public key algorithm E, thus the original hash value h is returned. However, $H(\tau.t + \tau.x + \tau.y)$ is to compute the hash value $h^{'}$ with the same method shown in signature equation. If $h = h^{'}$, then the evidence τ is integrity and valid, other-wise, τ is destroyed or altered, and unbelievable. The evidence is built of several blocks including time-stamps corresponding to the creation time of the alert message, the detection time of the intrusion, the alarm information such as IP packets, network connection, network flux, CPU status, system status, user status, processes status, swap status, and memory status etc. Our system monitors the network activities and classifies an input set (Ag) into *Self* and *Nonself*. When finding an intrusion, the corresponding evidences will be collected immediately. The goals of our Computer forensics system are self-adapting response and forensics in real-time. Therefore, the following shows in detail the dynamic evolution models for self, antigen, dynamic computer forensics, immune tolerance, mature-lymphocyte lifecycle, and immune memory.

2.2 The Dynamic Model of Self

In a real-network environment some network services and activities are often change, which were permitted in the past but may be forbidden at the next time.

$$Self(t) = \begin{cases} \{x_1, x_2, \cdots, x_n\} & t = 0 \\ Self(t-1) - Self_{variation}(t) \cup Self_{new}(t) & t \geq 1 \end{cases} \tag{11}$$

$$\begin{aligned} Self_{variation}(t) = \{x | x \in Self(t-1) \wedge \exists y \in B(t-1) \\ (f_{check}(y, x) = 2 \wedge f_{costimulation}(x) = 0)\} \end{aligned} \tag{12}$$

$$f_{check}(y, x) = \begin{cases} 2 & f_{match}(y, x) = 1 \wedge x.a \in Self(t-1) \\ 1 & f_{match}(y, x) = 1 \wedge x.a \notin Self(t-1) \\ 0 & otherwise \end{cases} \tag{13}$$

$$Self_{new}(t) = \{y | y \texttt{ is the new self element collected at time } t\} \tag{14}$$

Equation (11) stimulates the dynamic evolution of self-antigens, where $x_i \in \Re(i \geq 1, i \in N)$ is the initial self element defined. $Self_{new}$ is the set of newly

defined elements at time t, and $Self_{variation}$ is the set of mutated elements. $f_{check}(y, x)$ is used to classify antigens as either *self* or *nonself*: if x is a self-antigen, return 0; if x is a *nonself* one, return 1; if x is detected as *nonself* but was detected as a self-antigen before, then it may be a *nonself* antigen (needs to be confirmed), and return 2. $f_{costimulation}(x)(x \in Ag)$ simulates the co-stimulation in a biological immune system and indicates whether x is a self-antigen by an external signal. If x is confirmed self, return 1, otherwise, return 0. It is usually from the administrator. There are two advantages in this model. 1) *Self immune surveillance*: The model deletes mutated self-antigens ($Self_{variation}$) in time through surveillance. The false-negative error is reduced. 2) *The dynamic growth of Self*: The model can extend the depiction scope of *self* through adding new self-antigens ($Self_{new}$) into *Self*. Therefore, the false-positive error is prevented.

2.3 The Dynamic Mature Immune Model

$$Mat_{SD}(t) = \begin{cases} \phi & t = 0 \\ Mat_{SD}^{'}(t) \cap Mat_{new}(t) - Mat_{active}(t) - Mat_{dead} & t \geq 1 \end{cases} \tag{15}$$

$$Mat_{SD}^{'}(t) = Mat_{SD}^{''}(t) - S(t) \cup S^{'}(t). \tag{16}$$

$$\begin{aligned} Mat_{SD}^{''}(t) = \{y | y \in SD, x \in Mat_{SD}(t-1), x.age < \lambda, y.d = x.d, \\ y.age = x.age + 1, y.count = x.count\}. \end{aligned} \tag{17}$$

$$S(t) = \{x | x \in Mat_{SD}^{''}(t), \exists y \in SD(t-1), \langle x.d, y \rangle \in Match\}. \tag{18}$$

$$S^{'}(t) = \{y | y \in SD, x \in S(t), y.d = x.d, y.age = x.age, y.count = x.count + 1\}. \tag{19}$$

$$Mat_{new}(t) = \{y | y \in SD, y.d = x.d, y.age = 0, y.count = 0, x \in I_{maturation}(t)\}. \tag{20}$$

$$Mat_{active}(t) = \{x | x \in S^{'}(t), x.count \geq \beta\}. \tag{21}$$

$$\begin{aligned} Mat_{dead}(t) = \{x | x \in Mat_{SD}^{'}(t) \wedge (x.age > \lambda, x.count < \beta)\} \\ \cup \{x | x \in Mem_{SD}^{''}(t) \wedge \exists y \in SD(t-1), \langle x.d, y \rangle \in Match\}. \end{aligned} \tag{22}$$

Equation (15) depicts the lifecycle of the mature detector, simulating the process that the mature detectors evolve into the next generation. All mature detectors have a fixed lifecycle (λ). If a mature detector matches enough antigens ($\geq \beta$) in its lifecycle, it will evolve to a memory detector. However, the detector will be eliminated and replaced by new generated mature detector if they do not match enough antigens in their lifecycle. $Mat_{new}(t)$ is the generation of new mature *SDs*. $Mat_{dead}(t)$ is the set of *SDs* that haven't match enough antigens ($< \beta$) in lifecycle or classified self antigens as *nonself* at time t. $S^{'}(t)$ simulates that the mature *SD* undergo one step of evolution. $S^{''}(t)$ indicates that the mature *SD* are getting older. $Mat_{active}(t)$ is the set of the least recently used

mature *SD* which degrade into Memory *SD* and be given a new age $T > 0$ and count $\beta > 1$. Because the degraded memory *SD* has better detection capability than mature *SDs*, it is better to form a memory *SDs*. When the same antigens arrive again, they will be detected immediately by the memory *SDs*. In the mature detector lifecycle, the inefficient detectors on classifying antigens are killed through the process of clone selection. Therefore, the method can enhance detection efficiency when the abnormal network behaviors intrude the system again.

2.4 The Dynamic Memory Immune Model

$$Mem_{SD}(t) = \begin{cases} \phi & t = 0 \\ Mem_{SD}^{'}(t) \cap Mem_{new}(t) - Mem_{from-other}(t) & t \geq 1 \end{cases} \tag{23}$$

$$Mem_{SD}^{'}(t) = Mem_{SD}^{''}(t) \cup Mem_{clone}(t) - M_{dead}(t). \tag{24}$$

$$\begin{aligned} Mem_{SD}^{''}(t) = \{y | y \in Mem_{SD}, y.d = x.d, y.age = x.age + 1, \\ y.count = x.count, x \in Mem_{SD}(t-1) - Mem_{clone}(t)\}. \end{aligned} \tag{25}$$

$$M_{dead}(t) = \{x | x \in Mem_{SD}^{''}(t), \exists y \in SD(t-1), f_{match}(x.d, y) = 1\}. \tag{26}$$

$$\begin{aligned} Mem_{clone}(t) = \{x | x \in Mem_{SD}, y \in Mem_{clone}(t), x.d = y.d, \\ x.age = 0, x.count = y.count + 1\}. \end{aligned} \tag{27}$$

$$\begin{aligned} Mem_{new}(t) = \{x | x \in Mem_{SD}, y \in Mem_{active}(t), x.d = y.d, \\ x.age = 0, x.count = y.count\}. \end{aligned} \tag{28}$$

$$\begin{aligned} Mem_{from-other}(t) = \{x | x \in Mem_{SD}, y \in \cup_{i=(1,\cdots,k), i \neq k} Mem_{clone}^{i}(t), \\ x.d = y.d, x.age = 0, x.count = 0\}. \end{aligned} \tag{29}$$

$$Mem_{SD} = \{x | x \in SD, \forall y \in Self, (\langle x.d, y \rangle \notin Match \wedge x.count \geq \theta)\} \tag{30}$$

Equation (23) depicts the dynamic evolution of memory detector. $Mem_{SD}^{'}(t)$ simulates the process that the memory *SDs* evolve into the next generation ones. $Mem_{new}(t)$ is the set of memory *SDs* that are activated by antigens lately. These mature detector matched by an antigen will be activated immediately and turn to a memory detector. $Mem_{dead}(t)$ is the memory detector that be deleted if it matches a known self antigen. $Mem_{clone}(t)$ is the reproduced memory *SDs* when the detector distinguish a antigens. $Mem_{form-other}(t)$ is the memory *SDs* that transformed from other computers. The k indicates that the ID number of the computer. Therefore, dynamic model of immune is to generate more antibodies and enhance the ability of self-adaptation for the system.

2.5 The Antibody Cross

In order to keep the variety of individual as well as the optimal solution can be achieved, we divide the antibody gene to n gene bits set and utilize multi-point cross process. For example, we select two gene by random such as $G_1 = \{g_1, g_2, \cdots, g_n\}$, $G_2 = \{g_1^{'}, g_2^{'}, \cdots, g_n^{'}\}$. Select some points randomly, and then form two-point pair with some probability (p) to cross operation, to generate cross point set, and then to generate new gap of set $G_{new} = \{g_1, g_2, \cdots, g_i^{'}, \cdots, g_n\}$. Select cross point according to binomial distribution

$$P\{X = k\} = \binom{n}{k} p^k (1-p)^{n-k}, k = 0, 1, 2, \cdots, n \qquad 0 < p < 1. \tag{31}$$

$E(X) = np$, $D(X) = np(1-p)$, where X is the numbers of cross points. Then the G_1 and G_2 turn into the offspring G_{new} by the cross process.

2.6 The Antibody Variation

In order to prevent algorithm from converging prematurely, we take variation operation to the gene set $G_1 = \{g_1, g_2, \cdots, g_n\}$ after the cross process. Select variation point randomly and varied with some variation probability (p_m) to generate new generation $G_{new} = \{g_1, g_2, \cdots, g_i^{'}, \cdots, g_n\}$. Select variation point according to Poisson distribution

$$P_m\{X = k\} = \frac{\lambda^k e^{-\lambda}}{k!}, \qquad k = 0, 1, 2, \cdots \tag{32}$$

$E(X) = D(X) = \lambda > 0$, where X is the numbers of variation points. Then the G_1 turn into the offspring G_{new} by the variation process.

2.7 Dynamic Computer Forensic

$$\Gamma(t) = \begin{cases} \phi & t = 0 \\ \Gamma(t-1) \cup \Gamma_{new}(t) & t > 0 \end{cases} \tag{33}$$

$$\begin{aligned} &\Gamma_{new}(t) = \{\tau | \tau \in \Gamma \wedge (\tau.t = t, \tau.x = x, \tau.y = y^{'}, \tau.s = s^{'}, \\ &s^{'} = E_{k_{private}}(H(\tau.t + \tau.x + \tau.y)), x \in Ag_{nonself}(t))\} \end{aligned} \tag{34}$$

where $\Gamma_{new}(t)$ denotes the new evidences caught at time $t(t > 0)$, x is the intrusion IP packets caught by lymphocytes including two parts: $x.b$, the original evidence(the original IP packets), and $x.a$, the evidence extracted from $x.b$; $y^{'}$ depicts the snapshot of the environment in the host machine at time t, including the status of network, CPU, memory, and processes in the system etc.; and $s^{'}$ denotes the digital signature of the evidence, insuring its integrality, originality and authority.

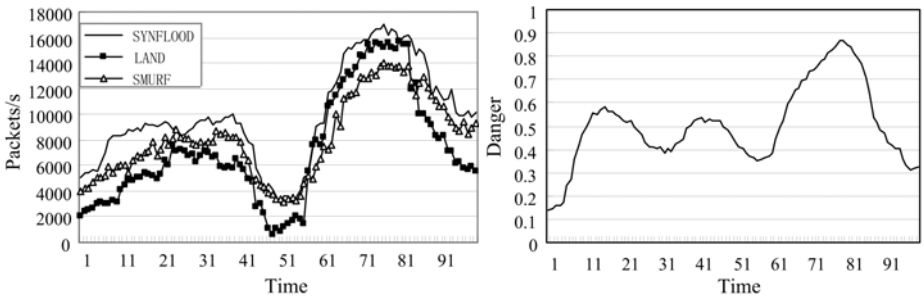

Figure 2: The left figure is the network suffering from the three typical incursions for instance and the right is the line of the network dangers obtained by *CFSAIS* at these incursions.

3 Experimental Results and Analysis

3.1 Experimental Environment and Evaluation Indicators

Experiments of attack simulation were also carried out in our Laboratory. To prove the intrusion detection performance, and reduce both false positive error rate and false negative error rate in contrast to the traditional *NIDS* techniques, we developed some series experiments. An antigen was defined as a fixed length binary string composed of the source/destination IP address, port number, protocol type, IP flags, IP overall packet length, TCP/UDP/ICMP fields, and etc. The network was attacked by 20 kinds of attacks, such as Syn Flood, Land, Smurf, and Teardrop. A total of 20 computers in a network were under surveillance. The task aimed to detect network attacks. Here are the coefficients for the model. We use *r-contiguous* bits matching rule ($r = 8$) for computing the affinity, $n = 40$ (the size of initial self set), and $\xi = 4$ (the number of new generated immature cells). The activation threshold is β; tolerance period is λ.

3.2 Results and Analysis

Figure 2 illustrates the levels of 3 kinds of attacks and depicts the evaluation of the network danger in *CFSAIS*. Danger changes when attack levels changes. The rise in attack levels is accompanied by a corresponding increase in danger, as implies the serious of network security. On the other hand, if attack levels decline, danger decreases accordingly after seconds of delay. Therefore, the network can stays on guard even when the attacks occur once again during a very short time. And Table 1 shows a portion of the evidences extracted by *CFSAIS* in real time. The evidences shown in table 1 indicate the network attack situation: The attack happened at 20:22:17 on Apr.9 2006. host 192.168.0.24 sent lots of requests to port 23 in host 192.168.0.1 (target). A large number of half-connection were

Table 1: Portion of Evidences collected by *CFSAIS* for Syn flood

	Result of Evidences
Attack time	Apr 9 20:22:17 2006
IP packets	tcp52:54:ab:39:02:db→00:20:ed:63:16:e6 192.168.0.24:256→192.168.0.1:23.S. tcp52:54:ab:39:02:db→00:20:ed:63:16:e6 192.168.0.24:512→192.168.0.1:23.S. tcp52:54:ab:39:02:db→00:20:ed:63:16:e6 192.168.0.24:768→192.168.0.1:23.S. . . .
Tasks	110 total, 2 running, 108 sleeping, 0 stopped, 0 zombie
Network flux	93715 packets/second
CPU status	20.1% user, 72.2% system, 0.0% nice,7.6% idle
System status	lower 90.3%
Users status	root, ids, ftp, ⋯, the total number is 16
Processes	56 processes: 55 sleeping, 1running, 0zom
Swap status	522072k av, 26432k used, 495640k free
Memory status	512292k av, 93063k used, 419228k free,0k shrd, 4542k actv

PID	USER	PR	NI	VIRT	RES	SHR	STAT	% CPU	% MEM	TIME+
5642	ids	15	0	60812	2500	1300	S	2.0	5.1	0:32.81
5646	ids	16	0	42204	7604	6484	S	2.0	1.5	0:00.35
1	root	16	0	2444	560	480	S	0.0	0.1	0:00.64
2	root	34	19	0	0	0	S	0.0	0.0	0:00.00
. . . . .										
1727	root	15	0	0	0	0	S	0.0	0.0	0:00.07

established between these two machines and the information were shown include the current network status, the processes, the resource of the host, the users status etc. The experiment shows that our *CFSAIS* can collect evidences of attack accurately and in a timely fashion.

4 Conclusions

In this paper, we have presented a model of computer forensics system based upon the theory of artificial immune system, and we have also illustrated the advantages of this model than traditional models. The concepts and formal definitions of immune cells are given. And we have quantitatively depicted the dynamic evolutions of self, anti-gens, immune-tolerance, and the immune memory.

Additionally, the model utilized a distributed and multi-hierarchy framework to provide an effective solution for the network intrusion. Finally, the experimental results show that the proposed model is a good solution for computer forensics.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under Grant No. 60373110, 60573130 and 60502011, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20030610003, the New Century Excellent Expert Program of Ministry of Education of China under Grant No. NCET-04-0870, and the Innovation Foundation of Sichuan University under Grant No. 2004CF10.

References

[Bashaw 2003] Bashaw, C.: "Computer Forensics in Today's Investigative Process"; "Proc. of 15th FIRST Conf. Computer Security Incident Handling & Response", Ottawa (2003)

[Burnet 1959] Burnet F. M.: "The Clone Selection Theory of Acquired Immunity"; Gambridge University Press, Cambridge (1959)

[Castro 2003] De Castro L. N., Von Zuben F. J., De Deus J. G. A.: "The Construction of a Boolean Com-petitive Neural Networks Using Ideas from Immunology"; Neurocomputing, 50(2003), 51-85

[Forrest 1994a] Forrest S., Perelson A. S.: "Self-nonself Discrimination in a Computer"; IEEE Symposium in Security and Privacy, Oakland, CA (1994), 202-213

[Forrest 1994b] Forrest S., Perelson A. S., Allen L., Cherukuri R.: "Self-Nonself Discrimination in a Computer"; "Proceedings of IEEE Symposium on Re-search in Security and Privacy", Oakland (1994)

[Kepler 1993] Kepler T. B., Perelson A. S.: "Somatic Hyper Mutation in B Cells: An Optimal Control Treat-ment"; Theoret Biol, (1993), 37-64

[Kim 1999] Kim J., Bentley P.: "The Artificial Immune Model for Network Intrusion Detection"; 7th European Congress on Intelligent Techniques and Soft Computing (1999)

[Li 2004a] Li T.: "Computer Immunology"; Publishing House of Electronics Industry, Bejing (2004)

[Li 2004b] Li T.: "An Introduction to Computer Network Security"; 1st edition, Publishing House of Electronics Industry Beijing (2004)

[Li 2005a] Li T.: "An Immune Based Dynamic Intrusion Detection Model"; Chinese Science Bulletin, 50(2005), 2650-2657

[Li 2005b] Li T.: "An Immunity Based Network Security Risk Estimation"; Science in China Ser. F Information Sciences, 48(2005), 557-578

[Li 2005c] Li T.: "An Immune-Based Model for Computer Virus Detection"; Lecture Notes in Computer Science, (2005), 59-71

[Moan 2004] Moan J.: "Computer Forensics in a Global Company"; in Proc. of 16th FIRST Conf. Computer Security Incident Handling & Response, Budapest (2004)

[Osles 2001] Osles L.: "Computer Forensics. The key to solving the crime" (2001)

[Reis 2002] Reis M. A., Geus P. L.: "Standardization of Computer Forensic Protocols and Procedures"; "Proc. of 14th FIRST Conf. Computer Security Incident Handling & Response", Hawaii, 1(2002), 15-20

[Srinivas 2003] Srinivas M., Andrew H., Sung.: "Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques"; International Journal of Digital Evidence, 1 (2003)