Mathematical Biosciences and Engineering

MBE, 17(6): 7221–7233.
DOI: 10.3934/mbe.2020370
Received: 02 July 2020
Accepted: 12 October 2020
Published: 22 October 2020

http://www.aimspress.com/journal/MBE

Research article

Artificial immune intelligence-inspired dynamic real-time computer forensics model

Zairong Wang[1], Xuan Tang[2], Haohuai Liu[3,*] and Lingxi Peng[4]

[1] Data Recovery Key Laboratory of Sichuan Province, School of Computer Science, Neijiang Normal University, Sichuan 641100, China
[2] School of Administration, Guangzhou University, Guangzhou 510006, China
[3] School of Chemistry, Guangzhou University, Guangzhou 510006, China
[4] School of Mechanical and Electrical Engineering, Guangzhou University, Guangzhou 510006, China

* **Correspondence:** Email: huaihuai99@163.com, j619wtcf@163.com; Tel: +8613725156696; Fax: +862039366375.

Abstract: Dynamic computer forensics is a popular area in computer forensics that combines network intrusion technology with computer forensics technology. A novel dynamic computer forensics model is proposed based on an artificial immune system. Simulating the artificial immune mechanism, the definitions of self, non-self, and immunocyte in the network transactions are given. Then, detailed evolution processes for immature detectors, mature detectors, and memory detectors are given. Real-time network risk evaluation equations are constructed, which can compute the risk of each type of network attack. Finally, computer forensics is accomplished according to the real-time network risk. The immune cells dynamically capture the real-time computer system status of the invading antigen, including CPU utilization, memory utilization, network bandwidth utilization status, etc. Theoretical analysis and comparative experimental results demonstrate that the proposed model improves the real-time efficiency and performance with low technical requirements for technicians compared with existing models.

Keywords: network security; computer forensics; artificial immune system

1. Introduction

Existing computer forensics models are mostly static. Its research and network monitoring technology, such as intrusion detection and real-time network risk assessment, are separated from each other. Effective electronic evidence can only be extracted after an attack by analyzing the fuzzy traces from event logs, hard disk files, and the information left in the system. However, for some highly skilled network intruders or hackers, the relevant logs, hard disk files, and other traces can be completely destroyed or modified, resulting in the ineffectiveness of measures taken afterward, even for the best analysis technologies [1,2]. Therefore, the combination of network intrusion technology and computer dynamic forensics technology, which takes evidence when the system is invaded or attacked, has become a focus of much research in computer forensics.

Early real-time dynamic forensics approaches mainly used network intrusion detection systems (NIDS), honeypots, and other network security technologies or tools to analyze and detect real-time network data flow [3]. However, NIDS have a very high false negative and false positive rate, resulting in the inaccuracy or incomprehensibility of detection results and poor practicability. Moreover, NIDS do not process network data according to procedures prescribed by law and entirely rely on network data fragments for detection. From the legal point of view, the evidence is not complete, and the electronic evidence obtained by the technology does not meet the requirements of evidence collection [2].

Honeypots can provide some evidence for tracking the attacker and their behavior by simulating the vulnerability of the host and fooling the attacker with a false attack target. However, at present, this technology is not mature, and there is possible subjective deception. Therefore, the electronic evidence collected by honeypots has not been accepted as electronic legal evidence. Finally, Metasploit, a dynamic forensics tool, reconstructs the crime scene by memory extraction.

From the perspective of forensics, all existing tools have certain limitations [4]. In addition, these forensics methods cannot identify unknown attacks. Pi et al. proposed a method to obtain evidence from the Internet of things equipment data using mobile terminal data [5]. Hosseini et al. reduced the false alarm rate and improved detection efficiency significantly by compressing network worm traffic through digital signal processing [6]. Talib introduced a black box test method based on functional scenes [7]. Mohammed et al. proposed a computer forensics method based on big data [8]. Park proposed a forensics method for developing a synthetic corpus [9]. Yee-Yang et al. introduced a cloud storage service CloudMe, which is helpful for the forensics of big data endpoints supported by cloud computing [10]. These methods have a certain degree of effectiveness but are not suitable for all computer equipment or systems, and they are difficult to deploy in practical applications.

Network security approaches based on artificial immune systems have long been regarded as one of the most important and promising research directions in the academic field [11–14]. For dynamic forensics, Tao proposed an immune-based network monitoring model that extracts dynamic forensic evidence based on an immune intrusion detection system [15]. Based on dynamic clonal selection, Ding et al. proposed an immune system model for intrusion monitoring and dynamic forensics [4]. Yang et al. proposed a computer forensics method based on an artificial immune system which has better real-time processing capabilities and adaptability [16]. However, these dynamic forensic methods are all based on the "self/non-self" recognition model proposed by Forrest et al. [17], where data is judged to be legal or illegal, and then dynamic forensic evidence is extracted. The amount of electronic evidence obtained is vast and most of this is invalid due to its inability to effectively distinguish and measure the harm caused by different types of attacks and attack intensity. Based on Danger Theory, Peng et al. proposed an automatic intrusion response system model which effectively reduced the number of intrusion responses and the cost of the response [12]. Wojciech et al. reviewed

recent computer forensics technologies [18]. Qi improved the K-means clustering algorithm, and then applied it to locate the computer and the network intrusion crimes′ real-time forensics. In their simulation experiments, including their analysis in MATLAB, their method obviously improves upon the existing K-means clustering approach [19]. Jeferson et al. introduced the major concepts of computer forensics and then describe the tools and related technology used in this area [20]. They discuss one case study for a Linux system, which involves computer monitoring, hardware and software analyses, chain of custody, and steganalysis. Finally, they used the NuDetective tool and prepared a computer forensics report. Therefore, this aimed to show a complete criminal investigation so that end users can duplicate the steps described in this work with low-level techniques. Flora et al. proposed a new methodology to support investigators during the analysis process, correlating evidence found through different forensics tools. The methodology was implemented by a system which can add semantic assertion data generated by forensics tools during extraction processes. These assertions enable more effective access to relevant information and enhance the retrieval by including reasoning abilities [21]. Existing computer forensics systems are faced with the problems of low forensics efficiency and difficulties in reconstructing the attack scene. The large number of low-quality alarms and forensics greatly affect the burden of system administrators.

In order to solve above problems, this paper first defines the non-self and self immune cells of the immune system for network activities, and then proposes a computing maneuver that can reproduce the invasion or attack scenario to realize a real-time forensics model (*ANAIM*). The model combines the network intrusion detection and network real-time risk assessment, considers all kinds of attacks on the host and the overall network risk intensity in real time, and then dynamically captures the crime scene according to the specific risk to obtain electronic evidence. Theoretical analysis and experimental results demonstrate that the proposed model improves the real-time forensics efficiency and performance with low requirements for technicians. It provides a new and promising direction for computer forensics.

The remainder of the paper is as follows: section 2 gives a theoretical model, section 3 provides comparative experiments and analysis, and the conclusions and future work are given in section 4.

2. Theoretical model

The network activity domain is defined as $D = \{0,1\}^l$, in which l is a positive integer. The antigen set is defined as $Ag \subset D$. Ag is a binary string that represents the characteristics of a network transaction by extracting the source and destination IP address, port number, protocol type, and part of the packet content from IP packets in network activities. The normal activity of a network is $SelfSet \subset Ag$. The illegal activities or network attacks are *NonselfSet*, where $NonselfSet \subset Ag$.$SAg \subset Ag$. SAg is the antigen to be detected, *SelfSet* represents a normal network service transaction, and *NonselfSet* represents a network intrusion or attack. $SelfSet \cup NonselfSet=Ag$, $NonSelfSet \cap selfSet=\varnothing$.

The detector set of the immune system is defined as $ID=\{<d, age, count, s>|d \in D, age \in N, count \in N, s \in R\}$, where d is the detector's gene from the set of existing detectors, *age* is the detector's age, *count* is the number of matching detector, s is the risk of the detector, N is the set of natural numbers, and R is the set of real numbers. Here, the immune detector consists of mature detectors and memory detectors, $ID = ME \cup$ MT, and $ME \cap MT=\varnothing$.

$$MT = \{x \mid x \in ID, \forall y \in SelfSet(< x.d, y >\notin Match \wedge x.count < \beta)\} \tag{1}$$

$$ME = \{x \mid x \in ID, \forall y \in SelfSet(< x.d, y > \notin Match \wedge x.count \geq \beta)\} \tag{2}$$

The matching relation in the network activity domain D is defined as $Match = \{<x, y> | x, y \in D, f_{match}(x, y)=1\}$, where $f_{match}(x, y)$ is the matching degree of x and y. In the actual process, this is calculated by the Euclidean distance, Hamming distance, or r-continuous bits. The memory detector set, ME, is composed of detectors that do not match themselves, and the matching threshold of the antigen is greater than β. The mature detector set, MT, is composed of detectors that do not match with themselves, but the number of antigen matches does not reach the threshold β, which belongs to the set of natural numbers N. Similarly, $<d, age>$ can be used to define the immature detector set UM, which can be randomly generated or by the antibody gene pool.

$$UM = \{<d, age> | d \in D, age \in N\} \tag{3}$$

Figure 1 gives the model structure and main processes [15]. The dotted arrow in the middle of this figure indicates the direction of cell evolution from bottom to top. First, an immature cell, *UM,* is randomly generated, and then after self-tolerance, the rest will evolve into the mature cell set, *ME*. For the mature cells, *ME*, if they can detect a certain number of antigens in their certain lifecycle, they will evolve into memory cells, *MT*. Otherwise, they will die. The solid arrow in the middle indicates the direction of detecting antigens. The arrow on the left side indicates that the memory cells, *MT*, first conduct a risk assessment, and then carry out the dynamic forensics according to the risk.

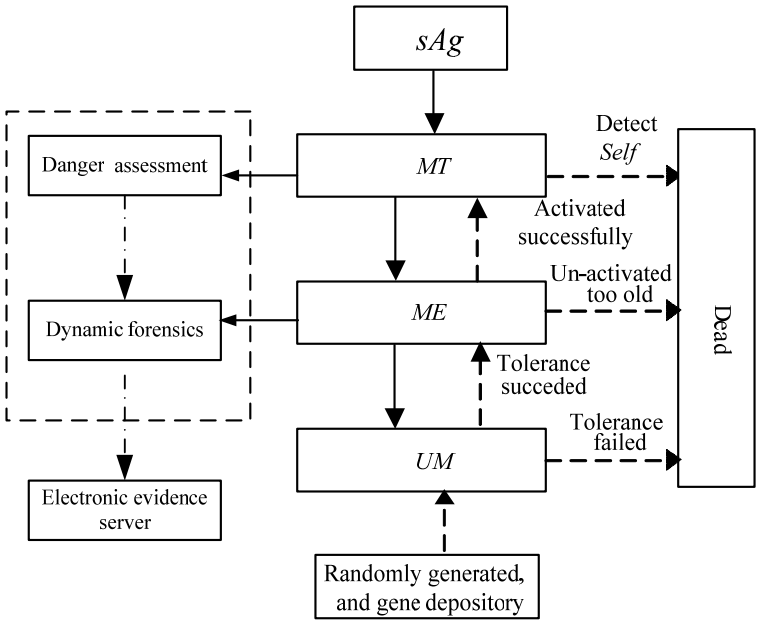

Figure 1. Model structure and main processes.

In the process of *UM* self-tolerance, after α number of tolerance cycles (α belongs to the set of natural numbers, N), the immature detectors that recognize self-antigens will be removed. Immature detectors will evolve into mature immune detectors.

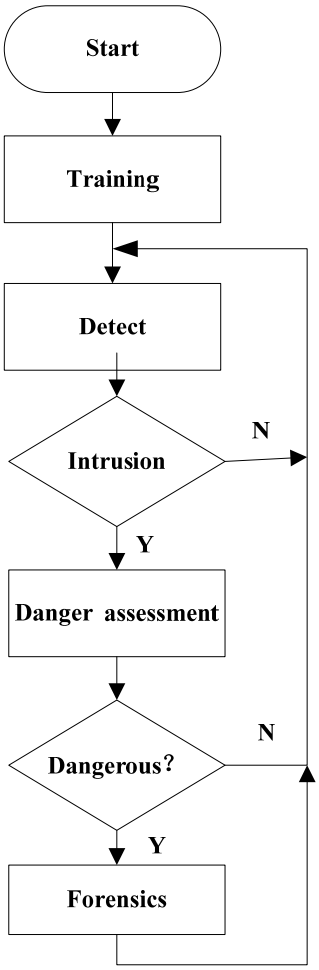

Figure 2. The flowchart of the model.

In Figure 2, the model is trained first and then carries out the detection. If an attack or intrusion is detected, the risk assessment is carried out. If the risk exceeds the given threshold, the forensics will be conducted, otherwise the detection will continue.

2.1. Immunocyte evolution

The evolution of immune cells in network activity can be divided into three stages. Stage I, from the beginning to the end of the first tolerance period α, the initial immature cell set $UM(0)$ and self-set $SelfSet(0)$ should be defined, in which immature cells evolve into mature cells.

Stage II begins from time $\alpha+1$ for the evolution of mature cells into memory cells, which is the self-learning process. Through clonal selection, mature cells produce a variety of memory cells that can detect non-self-antigens, among which self-antigen cells are finally used for immature tolerance.

Stage III is the production stage of the various parts of the immune system, from memory cells to the completion of system learning. This can be used for practical detection: memory cells are used for detection, mature cells are used for the detection of undetected antigens, and, finally, immature cells are used for self-tolerance through the remaining antigens.

2.2. Host dynamic risk calculation

For any memory detector, from time t-1 to time t, if an invasion or attack antigen is detected, its risk value s will increase according to formula (4). The risk will be calculated according to this formula, where η_1 (> 0, constant) is the initial risk value, and η_2 (> 0, constant) can be taken as the cumulative reward factor.

$$x.s(t) = \eta_1 + \eta_2 \bullet x.s(t-1) \tag{4}$$

However, if the memory detector does not detect the invasion or attack antigen for the moment, the risk value will be calculated according to formula (5), which indicates that the risk is decreasing.

$$x.s(t) = x.s(t\text{-}1) \bullet e^{-1} \tag{5}$$

For the random memory cells x and y, if $f_{\text{match}}(x.d, y.d) = 1$ is satisfied, then memory cells x and y have a definite similarity. The invasion or attack detected by these two kinds of memory cells could be the same type.

The host network risk index ($0 \le r_k(t) \le 1$) presents the network risk of host k for time t: $r_k(t) = 1$ indicates the extreme risk of the current host; $r_k(t) = 0$ indicates that the current host is not at risk. The larger the $r_k(t)$ value, the higher the risk the current system faces. The immune forensics system analyzes and extracts the intercepted network IP packets. The host network risk assessment is carried out when the memory cells match the extracted packets. Due to the different network risks caused by various kinds of attacks on hosts, μ_i is set as the risk weight of class i of the network attacks or intrusions, and the risk $r_{k,i}$ of class i ($1 \le i \le I$) of the network attacks for the host k for time t is calculated by formula (6), which is the product of the class i network risk and corresponding weights.

$$r_{k,i}(t) = \frac{2}{1 + e^{-\mu_i \left(\sum\limits_{MM_j \in A_i(t)} MM_j.s(t) \right)}} \text{-}1 \tag{6}$$

The overall network risk of the host k for time t is calculated by the following formula, which is the product of various network risks and corresponding weights.

$$r_k(t) = \frac{2}{1 + e^{-\sum\limits_{i=1}^{I} \mu_i \left(\sum\limits_{MM_j \in A_i(t)} MM_j.s(t) \right)}} \text{-}1 \tag{7}$$

It can be seen from the formula that the calculation of the real-time risk of the host security is linear with the number of memory cells, and its time-space cost is small. Therefore, it can achieve real-time performance in practical applications.

2.3. Risk-based dynamic forensics

The obtained dynamic electronic evidence is as follows:

$$\Psi \subset \{< t, x, y, s > | \, t \in N, x \in \text{Ag}, y \in \Omega, s \in \Omega\} \tag{8}$$

where t represents the moment when the evidence is obtained; x is the network IP packet intercepted

by the forensics system, which is composed of the original packet and the submitted data; *x.a* is equivalent to the evidence of the preliminary analysis and extraction (antigen presentation) of the original evidence, and *x.b* is the original evidence (original IP packet). *y* is the network environment for time *t*, which is equivalent to the cell status when the biological immune system captures the invading antigen, corresponding to the computer system process status, CPU utilization, memory utilization, network bandwidth utilization of the computer system, etc. *s* is the encryption signature of $<t, x, y>$ in the forensics model. For example, in MAC (Message Authentication Code) '+' connects strings. To ensure the authority and originality of the electronic evidence, the following is used:

$$s' = E_{kpriv}(H(\tau.t + \tau.x + \tau.y)) \tag{9}$$

where $<t, x, y>$ is first converted into a string, and the hash value *h* is calculated; then, the private key *kpriv* is used to encrypt the hash value *h* through the public key algorithm *E* to obtain the digital signature. The public key can be used to decrypt the evidence, and extract the hash digest. *E* is a signature encryption algorithm, such as DSS or RSA, *kpriv* represents the private key of the forensics model, and *H* is a one-way hash function. Theoretically, if *kpriv* is long enough, *s* will be more secure and can resist any attack. At the initial time, Γ is empty, and evidence will be collected according to the risk level with the appearance of system attacks, as shown below:

$$\begin{aligned} \Psi_{new}(t) = \{\psi \mid \psi \in \Psi \wedge (\psi.t = t, \psi.x = x, \psi.y = y', \psi.s = s', \\ r_m(t) > \theta_m \vee r_{m,i}(t) > \theta_{m,i}\} \end{aligned} \tag{10}$$

When obtaining evidence for $\Psi_{\text{new}}(t)$, one of two possible results will be reported. When the overall risk value $r_m(t)$ of the host exceeds the given threshold, this indicates that the overall risk degree of the host against the attack has seriously affected the normal operation of the host, θ_m. When the network risk $r_{m,i}(t)$ of the host against the class *i* attack exceeds the given threshold, $\theta_{m,i}$, then it shows that the class *i* attack and its variants detected by the host have risks that affect normal operation, which calls for real-time forensics.

There are two purposes for setting the risk threshold value in the model. On the one hand, it evaluates the real-time network risk of all detected attacks and similar attacks, so as to associate all attack information, reduce the false alarm, and capture the real-time operation of the system when the system is at risk, thus providing important information for the reproduction of attacks. In addition, many intrusion behaviors with low risk are ignored, which improves the low efficiency of the existing dynamic forensics model relying on the intrusion detection system and improving the efficiency and quality of dynamic forensics. The forensic system adopts an intrusion tolerance strategy, and low-risk intrusions are ignored to reduce the number of alarms. In this way, the proposed model adopts a strategy of intrusion tolerance, and low-risk intrusion will not affect the normal running of the computer.

3. Experiments

The experimentation was carried out in the Data Recovery Laboratory of Neijiang Normal University, a Sichuan Provincial Key Laboratory. The experimental environment provided is a 1000 m local area network with 50 computers connected to the Internet through a class C IP address 210.41.176.*. The server operating system is CentOS 7.6. The server provides email, WWW, FTP and

VPN services. The forensic system is also deployed on the server. The antigens in the network activities have a fixed length and are composed of 256-bit binary strings describing network transaction characteristics, such as source and destination IP addresses, protocol types, port numbers and packet contents extracted from IP packets. In the experimentation, the tolerance period of the antigen is set to α = 50 h, the activation parameter of the memory cell is set to β = 15; the matching relationship is calculated by r-continuous bit matching with r = 8. For the risk calculation parameters, η_1 is set to 0.001 and η_2 is set to 0.9998. The forensics risk thresholds θ_m and $\theta_{m,i}$ are set to 0.3. This level of risk will not affect the normal operation of the system. These parameters have been demonstrated to be effective in [15,23], therefore similar repeated experiments will not be conducted in this paper.

3.1. Real-time network risk assessment experiment

The immune-based intrusion detection system has good self-learning, is self-adaptive and diverse, and has good detection performance for unknown attacks. In order to verify that the model is effective for evaluating network risks, we carry out the TCP reset attack on the network, a common DDoS attack. The attack intensity curve and real-time risk curve of the model evaluation are shown in Figure 3 below. It can be seen in the figure that from 0 to 8 seconds, the intensity of the network attack increases, and the corresponding network risk is also on the rise. From 9 to 12 seconds, the intensity of the network attack decreased rapidly, and the risk decreases correspondingly.

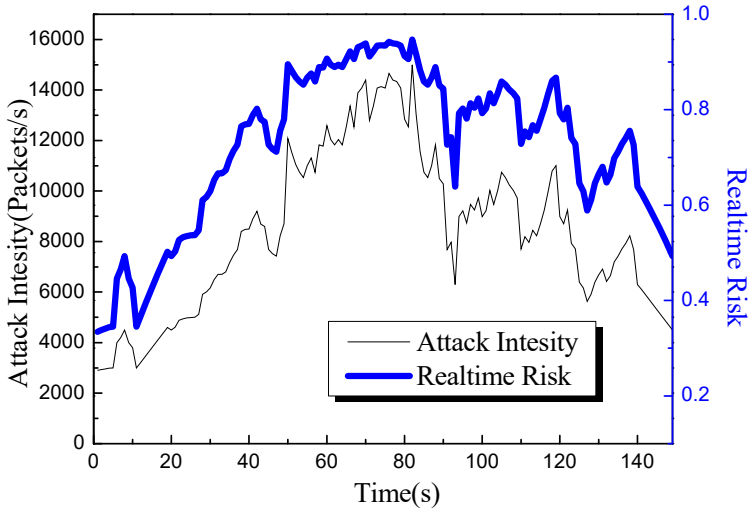

Figure 3. The intensity and network real-time risk curve of the TCP reset attack.

In order to further verify the effectiveness of the model for real-time risk assessment, a LAND attack is launched on the network. By constructing and sending packets with the same source address and target address, network equipment without a corresponding protection mechanism will be paralyzed. From the experimental results in Figure 4 below, it can be seen that when the network attack intensity increases, the real-time risk curve of the network will decrease. When the network attack intensity decreases, the real-time risk curve of the network decreases correspondingly, and the real-time risk curve of the network and the attack intensity also demonstrate a good consistency.

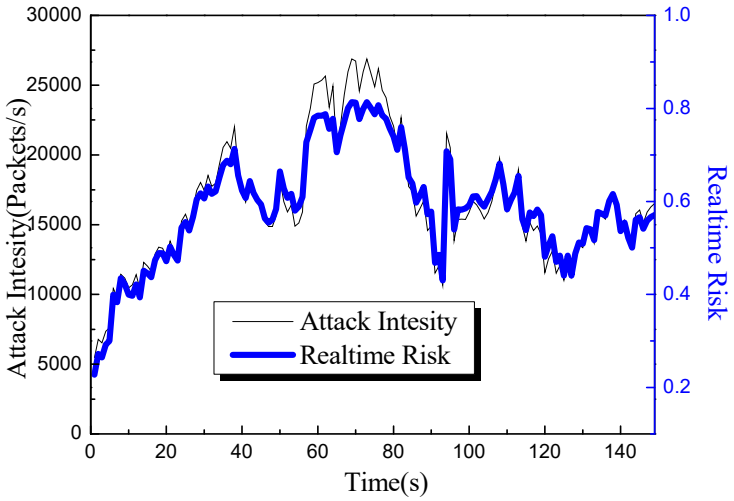

Figure 4. The intensity and network real-time risk curve of the LAND attack.

From the above two attack experiments it can be concluded that the real-time risk curve of the model is consistent with the network attack intensity, which shows that the model can effectively evaluate the real-time security of the whole network.

3.2. Comparative experiments

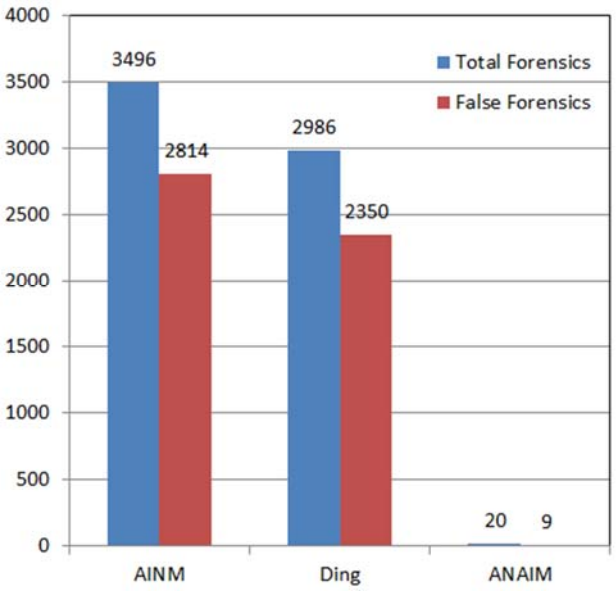

Figure 5. The number of comparisons of effective evidence.

Many studies have shown that intrusion detection systems based on immune systems are good at self-learning, self-adaptivity and diversity, and can effectively identify unknown attacks [22]. To verify the efficiency of the *ANAIM* model in dynamic forensics, simulation experiments are designed to

compare it with the *AINM* [15] and Ding [23], which are based on the "self/non-self" recognition model proposed by Forrest [17]; however, Ding's model extends this model. The simulation is designed to generate forensics based on the identification of the attack or non-attack. The experimental results are shown in Figure 5 above.

In Figure 5, it can be seen that the number of dynamic forensics of the AINM model is 3496, 2814 (80.5%) for false attacks and 682 (19.5%) for real attacks, while the total number obtained by the *ANAIM* model is only 20; of these, 9 (45%) are false alarms and 11 (55%) are real attacks. For Ding's model, 2986 are false alarms, 2350 (78.7%) are false attacks, and 636 (21.2%) are real attacks. Ding's model improves the efficiency, but *ANAIM* has greatly improved the forensics performance. It is evident that the ANAIM model dramatically improves the accuracy and efficiency compared with the AINM model. The other models can obtain the same results, but the efficiency is much lower than ANAIM. The greatest significance is that *ANAIM* combines similar alarms and greatly reduces the number of alarms. System administrators can then focus on analyzing this aggregated alarm information.

3.3. Dynamic forensics instance

Simulation results show that the *ANAIM* model can effectively distinguish and measure the harm caused by different types of attacks and their corresponding intensity, and the effectiveness and amount of electronic evidence obtained are better than the "self/non-self" model. In order to reflect the dynamic nature of model forensics, Table 1 gives an example of dynamic forensics in which *ANAIM* can capture dynamic real-time network activity, network flow, and connection and CPU utilization while the static forensics model does not.

Table 1. Instances of *ANAIM* forensics.

Time	Sept. 12, 2019 19:48:45
Network activity	TCP 127.0.0.1:2514 127.0.0.1:1110 ESTABLISHED TCP 210.41.176.XXX:2516 192.168.0.1:80 ESTABLISHED TCP 192.168.0.117:1194 192.168.0.1:80 SYN_RECV TCP 192.168.0.117:4556 192.168.01.:80 SYN_ RECV TCP 192.168.0.117:4559 192.168.0.1:80 SYN_ RECV TCP 192.168.0.117:4562 192.168.0.1:80 SYN_ RECV TCP 192.168.0.117:4568 192.168.0.1:80 SYN_ RECV TCP 192.168.0.117:4574 192.168.0.1:80 SYN_ RECV …
Network connection	7412
Network flow	63242 Packets/s
CPU utilization	10.3% user, 84.3% system, 0.0% nice, 5.4% idle
Swap	471294K av, 2457813K used, 95896K free, 112715K cached
Memory	220311K av, 1777208K used, 25895K free, 68398K buff
process	httpd, ids, mysqld, top…32 processes
….	…

Analysis of the simulation results shows that the *ANAIM* model can extract first-hand evidence

of the attack or intrusion scene clearly and accurately, and the electronic evidence is strong in persuasiveness, which can provide sufficient evidence for the reconstruction or restoration of the network attack or intrusion scene. This does not require professional computer expertise for the forensics personnel.

4. Discussions

Traditional static computer forensics technology mainly accomplishes the task by recovering and reconstructing residual attack information in the system, then sorting out all the file information in the computer to be verified, and extracting data for information mining or automatic analysis of the residual information. Finally, all the extracted data will be expressed in a form that can be understood by analysts, followed by data visualization and further analysis. Traditional static technology cannot accurately extract real-time evidence to describe the time of a specific attack and its surrounding environment and other detailed aspects; therefore, the extracted evidence is not persuasive. In addition, there is a high demand for the technological expertise and experience of forensics personnel. As a result, existing dynamic forensics methods are not useful due to immature technology and limited forensic tools [4,15,16].

Dynamic computer forensics technology can be combined with network intrusion detection. It can automatically and completely record the time, place, and process of the intrusion and the state of the whole network environment before and after the intrusion. A dynamic computer forensics mechanism can reproduce the whole intrusion process by organically integrating and sorting this information. This mechanism is better than the static forensics approach for obtaining evidence. It has advantages in the aspects of place, time and environment.

5. Conclusions

This paper proposes a dynamic forensics model based on artificial immune systems by reproducing the attack scene. The model can accurately and quantitatively calculate the host type and overall risk in real time and extract dynamic forensic evidence in accordance with the real-time risk. The forensics results can express the attack scene well with high efficiency, strong real-time performance, and low technical requirements for forensics personnel, providing a new direction for computer forensics. More experiments will be conducted in future work and the theoretical model will be further improved. More interesting and better results can be expected to be achieved.

Acknowledgments

This work was supported by Foundation of Guangzhou Yangcheng Scholars Project (No. 201831837), National Social Science Foundation ((No. 18BJL065), and Incubation Project of Neijiang Science and Technology Bureau ((No. 2019KJFH001).

Conflict of interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. L. Peng, Q. Zhang, *Dynamically real-time computer forensics paradiam with immune*, International Conference on Signal Processing (ICSP), 2014.
2. L. Wang, H. Qian, Computer Forensics and Its Future Trend, *J. Software*, **14** (2003), 1635–1644.
3. E. S. Pilli, R. C. Joshi, R. Niyogi, Network forensic frameworks: Survey and research challenges, *Digital Invest.*, **7** (2010), 14–27.
4. J. Ding, X. Liu, T. Li, S. Yang, P. Yang, Dynamic Computer Forensics Based on Artificial Immune System Against Network Intrusion, *J. Sichuan Univ. (Eng. Sci. Edition)*, **36** (2004), 108–111.
5. H. Pi, Electronic data forensics in the age of Internet of everything, *Chin. Inf. Secur.*, **5** (2019), 71–73.
6. S. Hosseini, A. Jahangir, M. Kazemi, Digesting Network Traffic for Forensic Investigation Using Digital Signal Processing Techniques, *IEEE Trans. Inf. Forensics Secur.*, **14** (2019), 3312–3321.
7. T. Abu, Testing closed source software: computer forensic tool case study, *J. Comput. Virol. Hack. Tech.*, **14** (2018), 167–179.
8. M. Asim, D. R. Mckinnel, A. Dehghantanha, R. M. Parizi, G. Epiphaniou, Big data forensics: Hadoop distributed file systems as a case study, in *Handbook of Big Data and IoT Security*, Springer, (2019), 179–210.
9. J. Park. Trede and VMPOP: Cultivating multi-purpose datasets for digital forensics – A Windows registry corpus as an example, *Digit. Invest.*, **26** (2018), 3–18.
10. Y. Teing, A. Dehghantanha, K. R. Choo, CloudMe forensics: A case of big data forensic investigation, *Concurr. Comp-Pract. E.*, **30** (2017), 1–12.
11. J. Zeng, X. Liu, T. Li, C. Liu, L. Peng, F. Sun, A self-adaptive negative selection algorithm used for anomaly detection, *Prog. Nat. Sci.*, **19** (2009), 261–266.
12. L. Peng, D. Xie, Y. Fu, W. Xiong, Y. Shen, Automated intrusion response system model based on danger theory, *J. Commun.*, **33** (2012), 136–144.
13. H. Deng, L. Peng, J. Zhang, C. Tang, H. Fang, H. Liu, An intelligent aerator algorithm inspired-by deep learning, *Math. Biosci. Eng.*, **16** (2019), 2990–3002.
14. C. Liang, L. Peng, An Automated Diagnosis System of Liver Disease using Artificial Immune and Genetic Algorithms, *J. Med. Syst.,* **37** (2013), 9932.
15. T. Li, An immune based model for network monitoring, *Chin. J. Comput.*, **29** (2006), 1515–1522.
16. J. Yang, T. Li, S. Liu, T. Wang, D. Wang, G. Liang, Computer Forensics System Based on Artificial Immune Systems, *J. Univers. Comput. Sci.*, **13** (2007), 1354–1365.
17. S. Forrest, A. S. Perelson, L. Allen, R. Cherukuri, Self-Nonself Discrimination in a Computer, in *IEEE Computer Society Symposium on Research in Security and Privacy*, IEEE Computer Society Press, (2002), 202–212.
18. M. Wojciech, C. Luca, W. Steffen, Recent Advancements in Digital Forensics, *IEEE Secur. Priv.*, **15** (2017), 10–11.
19. Y. Qi, Computer Real-Time Location Forensics Method for Network Intrusion Crimes, *Int. J. Network Secur.*, **21** (2019), 530–535.
20. J. dos Santos Almeida1, L. de Santana Nascimento, D. A. M. José, Computer Forensics: A Linux Case Study Applied to Pedophilia Crime Investigation in Brazil, *Int. J. CyberSecur. Digit. Foren*, **8** (2019), 31–42.
21. A. Flora, C. Aniello, C. Giovanni, N. Fabio, A semantic-based methodology for digital forensics

analysis, *J. Parallel Distr. Comput.*, **2020** (2020), 172–177.
22. S. Hofmeyr, S. Forrest, Architecture for an artificial immune system, *Evol. Comput.*, **8** (2000), 443–473.
23. J. Ding, An Extended Immune-Based Model for Computer Forensics, *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, 2009.

AIMS Press