A Layered Multi-agent Detection Model for Abnormal Intrusion Based on Danger Theory

Huang Xiao Tao , Li Sha ,Huang Li Qun
College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
E-mail: huangxt@hust.edu.cn, dudulee1980@sohu.com,huanglq2002cn@163.com

Abstract: **A layered multi-agent detection model for abnormal intrusion, based on danger theory, is presented according to the research on the danger theory and artificial immunity. The model, with three layers in the frame, conducts the real time monitoring and danger judgment on the host computer and network resource before it recognizes the *nonself*, and then the danger signal activates the immunity recognition. The danger judgment conducted by cloud model can recognize the harmful *self* and harmful *nonself* effectively, which ensures the system safety and improves the performance of detection system. Thus, the probability of misinformation and omission is decreased to some extent.**

Keywords: danger thoery; cloud model; artificial immunity; multi-agent; intrusion detection

I. INTRODUCTION

The artificial immunity technique refers to the system that can resist the outside intrusion by simulating the human immunity function on computers. Its emergence has developed the abnormal intrusion detection, which provides the adaptive protection for computers. However, some difficult problems occurred during the research in artificial immunity. The false positive and false negative, for example, are present due to the very definite boundary of traditional self-nonself recognition, as shown in Fig. 1.

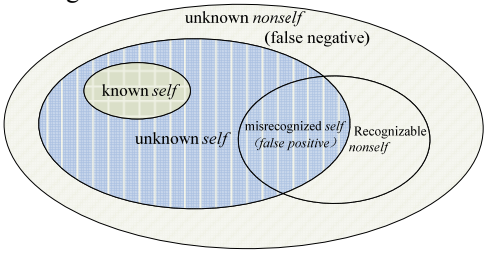

Fig. 1 Self-nonself overlay model

The danger theory in immunology, firstly established by immunologist Matzinger [3], was introduced into the artificial immunity system in 2002 by Doctor Uwe Aickelin [4] of University of Norttingham, Britain. Although some researchers at present have conducted some important research in applying the danger theory in the intrusion detection [5, 6], there is still no clear definition for the word *danger* in the intrusion detection system. Furthermore, some nonself, recognized by danger signal and artificial immunity model coordinately, may do little harm to the host computers or network system. Thus, so many non-danger alarms produced will affect the operation efficiency of system.

The conception *danger* is defined and a layered multi-agent detection model for abnormal intrusion is provided in this paper using the uncertain transformation model, cloud model. This model can recognize *harmful self* and *harmful nonself* of different danger level effectively, which ensures the workability of host computers and network system and improves the detecting efficiency to some extent.

II. CLOUD MODEL

A. General

The cloud model refers to the uncertain conversion model between some qualitative definition and its quantity by linguistic terms [7].

Definition for *cloud*: suppose U is a quantitative domain with precise values, C is the qualitative definition in the U. If the quantitative value performs $x \in U$ and x is a random value of qualitative definition C, then the certainty degree of x to C, namely $\mu(x) \in [0, 1]$, is a stability-prone random number:

$$\mu: U \to [0, 1] \quad \forall x \in U \quad x \to \mu(x)$$

Thus the distribution of x in U is defined *cloud*, each x called a cloud droplet [8, 9].

The numerical characteristic of one-dimensional normal cloud can be described by the expectation *Ex,* entropy *En* and excess entropy *He*, briefly written as *C=U(Ex, En, He)*, where *Ex* indicates the cloud center, specifying the center value of corresponding fuzzy concept; *En* measures the fuzzy degree of concept whose size can reflect the element number accepted by fuzzy concept in the domain; *He* indicates the discrete degree, namely the entropy of entropy.

As a complexly new model, the normal cloud model is developed from the normal distribution of probability theory and bell-shaped membership function of fuzzy set, which weakens the precondition of normal distribution. The random factors to determine the uncertainty, for example, can not be evenly small and can interact with each other. The excess entropy is here used to measure the deviation from normal distribution, expanding normal distribution to *generalized normal distribution* [10]. The change of system parameters chosen in this paper meets the requirements of *generalized normal distribution*, which can be described by normal cloud model.

B. Cloud Generator and Rule Generator

The cloud generator, briefly named CG, refers to the algorithm modularized by software or inserted in hardware to generate cloud model. The CG establishes the mapping

relation between qualitative values and quantitative values, basically including the normal cloud generator, backward cloud generator, antecedent cloud generator and consequent cloud generator, as shown in Fig. 2. The arithmetic for the four cloud generators above can be found in [7].

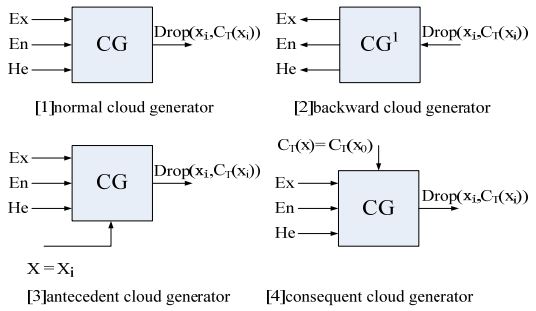

Fig. 2 Cloud generators

The single rule under single condition can be expressed as if A then B, where A and B are qualitative concepts. The single rule can be made if one antecedent cloud generator is connected with another consequent one as shown in Fig. 3, which is called single condition single rule generator.

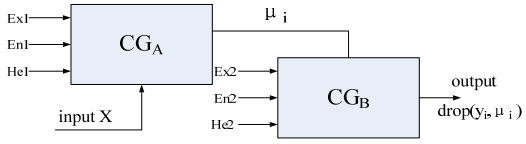

Fig. 3 Single rule under single condition

III. DANGER JUDGMENT

The cloud model is introduced to conduct transformation between qualitative concepts and quantitative values, because the judgment on danger levels, such as *dangerous, moderately dangerous, moderately safe* and *safe*, is uncertain, while the data resource of host computer and network is quantitative.

Definition 1： *Cloud of system resource variables: CLOUD=<SV, Ex, En, He, f, n>*

where $SV=S\_h \cup S\_n$ denotes the variable of system resource; *S_h* is the variable set of monitored host computer, $S\_h=\{C, M, S, I \ldots\}$, where *C* represents occupancy rate of CPU, *M* is occupancy rate of RAM, *S* is number of exchange space available, *I* is operation of I/O; *S_n* indicates the variable set of monitored network resource, $S\_n=\{D, L, W, F, \ldots\}$, where *D* is network delay, *L* is package loss rate, *W* is bandwidth, *F* is flow. *n* denotes the number of cloud droplets.

The cloud of system resource variables defined here refers to multi-dimensional cloud because the variable set *SV* includes many resource variables. The single rule under multi conditions (If A1,A2,… Then B), generally the multi-dimensional cloud generator, is suitable to calculate the multi-dimensional cloud of system resource. However, a great number of multi dimension clouds will be needed if all the linguistic terms are described in the multi-dimensional domain of space related to antecedent, which is too complicate to realize. Thus, the Boolean calculation of cloud model can be used to simplify the multi-dimensional cloud into the complex of many one-dimensional clouds, if the linguistic terms under multi conditions are too complicate, which realizes the qualitative reasoning by:

$$CLOUD = Cloud\_sv_1 \cap Cloud\_sv_2 \cap \cdots \cap Cloud\_sv_p$$

where *p* denotes the variable number of *SV*, $\cap$ not the *simple logic And*, but the conceptual calculation of cloud model to *soft And*.

The simplified one-dimensional clouds are calculated one by one as follows.

Firstly, the danger levels are classified as {*dangerous*, *moderate dangerous*, *common dangerous*, *moderate safe*, *safe*}. Secondly, the system resource variables are sampled according to the responding danger level. Finally, the backward normal cloud algorithm is used to compute the numerical characteristics (*Ex, En, He*) of danger level variables.

Algorithm for one-dimensional backward cloud (take variable *C* for example):

Input: sample point c_i, where $i=1, 2\ldots, m$.

Output: (*Ex, En, He*).

Algorithm steps:

① the sample mean $\overline{C} = \frac{1}{m}\sum_{i=1}^{m} c_i$ and sample variance $S^2 = \frac{1}{m-1}\sum_{i=1}^{m}(c_i - \overline{C})^2$ of set c_i are calculated.

② $Ex = \overline{C}$;

③ $En = \sqrt{\frac{\pi}{2}} \times \frac{1}{m}\sum_{i=1}^{m} | c_i - Ex |$;

④ $He = \sqrt{S^2 - En^2}$。

The numerical characteristics of each one-dimensional cloud are computed by the algorithm above, and then a group of numerical characteristics (Ex_j, En_j, He_j) is determined, where $j=1, 2, \ldots, p$. Thus, the one-dimensional cloud for each danger level is obtained.

The rule generators are created according to the single rule under multi conditions, as shown in Fig. 4. Furthermore, some rules are specified as follows:

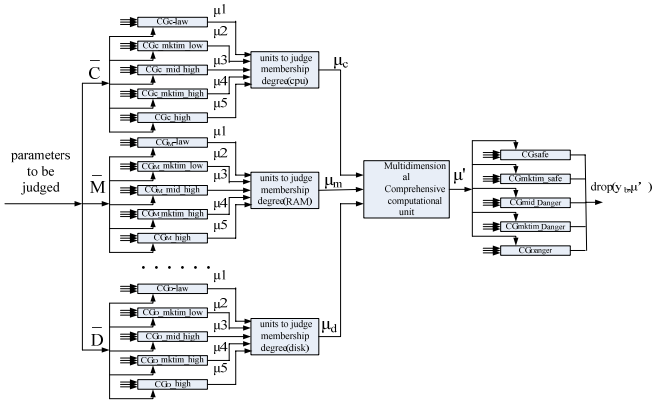

Fig. 4 Danger-perceivable cloud rule generator

Rule 1: If occupancy rate of CPU is high and that of RAM is high, then the danger level of system is high.

Rule 2: If occupancy rate of CPU is high and network flow is large, then the danger level of system is high.

Rule 3: If occupancy rate of CPU and RAM is low and network flow is small, then the danger level of system is low.

Rule 4: If CPU operates fast and occupancy rate of RAM is low and network delay is small and package loss rate is low, then the danger level of system is low.

……

Default rule: If the input rule is not among the ones above, then the system is safe.

The designed detection rules should cover all the input cases so that the detection system can be kept stable. After the rule design is finished, the qualitative rules will be mapped to the quantitative variables, and the cloud rule generator is used to realize these predefined rules [11]. The system resource variables are sampled in a longer time, and the numerical characteristics of each system resource variable are calculated by one-dimensional backward cloud. Then the membership degree of one-dimensional cloud to danger level is computed, and thus the danger level of sample can be obtained by the detection rules.

Ⅳ. Model Structure and System Design

A. Layer structure of model

Base on danger theory of immunity and intelligent agent technique, a structure of network detection system on abnormal intrusion is provide in this paper to monitor and detect the distributed network architecture, as shown in Fig. 5.

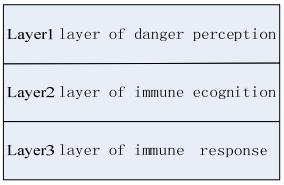

Fig. 5 Layer structure of model

(1) Layer of danger perception

Compared with traditional immunity-based intrusion detection system, the layer of danger perception is added into the new system before the traditional *self-nonself* is recognized, which learns from the way that the doctors diagnose the patients. When viruses or other processes intrude the host computers or network, some symptoms such as network congestion, high package loss rate and high occupancy rate of CPU or slow response will emerge. The danger level is judged by monitoring host computers and network resource, and then a danger signal is created to activate the immunity recognition.

(2) Layer of immune recognition

The mature detector is activated after it receives the danger signal from the layer of danger perception, and then the *self-nonself* is recognized to determine the *harmful self* and *harmful nonself*.

(3) Layer of immune response

The layer of immunity response reacts to the detected intrusion. Whether interaction with user is conducted and any measure is taken or not depends on the rules established in advance.

B. Design of layered multi-agent system

Each host computer in this system model introduces an immunity agent (IA) which acts like a lymphocyte of human body, and the whole network is just like human body. The IA is made up of a multi-agent system, as shown in Fig. 6.

Definition 2：*Immunity agent IA=<ID, MAS, KB, AB>*

where *ID* denotes the identification of *IA*; *MAS* means multi-agent system that makes up the immunity agent, *MAS*={*MA*, *GA*, *AgBA*, *DA*, *DMA*, *RA*, *CA*}; *KB* is the knowledge base of immunity agent; *AB* indicates the detector base.

(1) Monitoring agent (MA): It is responsible to conduct a real-time monitoring on system resource of host computers and network. The resource variable cloud of each danger level can be obtained by cloud model through initial learning. And then sampling regularly system resource and judging danger level are performed. The danger signal will be sent if danger is determined.

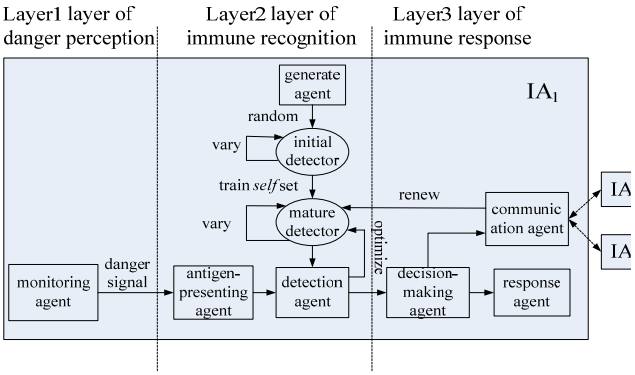

Fig. 6 Layered multi-agent detection model for abnormal intrusion based on danger theory

The monitoring agent involves the cloud calculating and detection rule generating, so the cloud controller, a key component, is needed to be designed, whose basic fundamental is shown in Fig. 7.

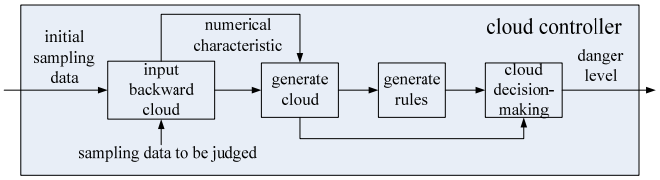

Fig. 7 Basic fundamental of cloud controller

The cloud controllers control various signals from intrusion detection system by cloud rule generators, involving the following steps:

① determine the input variables and output variables of cloud controller;

② design the control rules of cloud controller, including content and quantity of rules.

③ select the expression types of antecedent cloud and consequent cloud;

④ determine the parameters of cloud controller (such as quantitative factor and numerical characteristics) and the domain for input and output variables;

⑤ design the control algorithm of cloud controller;

⑥ select the suitable sampling time and the quantity of sample point, which directly affects the cloud quality. The larger the point quantity is, the smaller the error will be.

(2) Generating agent (GA): It is responsible to collect the normal behavior information of the network and host computers. The initial detectors are generated randomly before the mature detectors are created by negative selection algorithm.

Definition 3：*Mature detector D=<Gene, B_T, L_T, S>*

where $Gene = g_1 g_2 \ldots g_{nn}$ denotes gene coding of the detector, n being the gene number, $g_i (1 \le i \le n)$ as gene being a meaningful string such as protocol type, port number, resource/destination IP address, $g_i = \{0,1\}_i^{\ k}$, k being the length of g_i; $L\_T = n * T$

($n=0,1,2,\ldots$) indicates the lifetime of detector, T being the unit lifecycle of detector; B_T is the generation time of detector; S means the activation flag of detector.

(3) Antigen-presenting agent (AgBA): This agent is activated by the received danger signal, and then characteristics of the host computer and network in current system is extracted and coded.

Similar to the gene coding of mature detector, the antigen can be defined as $Ag=a_1a_2\ldots a_n$, $a_i=\{0,1\}_i^k$. The antibody shares the same string length and data structure with antigen.

(4) Detecting agent (DA): The presented antigen is detected by modified r-contiguous matching algorithm [12]. If the matching number reaches the activation threshold, the immunity response is conducted and upgraded into a memory detector.

The matching function between detector and antigen is

$$Match(D,Ag)=\sum_{i=1}^{n}P_i, \qquad P_i=\begin{cases}1, & g_i=a_j\\ 0, & \text{others}\end{cases}$$

The affinity function is $f(D,Ag)= Match(D,Ag)/N'$, where N' denotes the gene number of detectors or antigens.

(5) Decision-making agent (DMA): It is responsible to conduct a decision-making on detection results, and then judge by user-made interaction rules whether it interacts with users or not.

(6) Responding agent (RA): It takes the responding measures by the decision from RA.

(7) Communicating agent (CA): It is mainly responsible to conduct communication among AIS agents to improve the operation efficiency and detection accuracy of model and detect effectively the coordinated attacks.

C. Processing flow of multi-agent system

The following steps are generally performed in this system to conduct the detection and interaction inside AIS agent and among AIS agents:

(1) *GA* generates initial detector randomly and trains its tolerance by *self* set, thus creating mature detectors;

(2) *MA* conducts a real-time collection on the system resource data of host computers and determines the danger level by cloud model. If danger exists, a danger signal will be sent to activate *AgBA*;

(3) After *AgBA* is activated, characteristics of the host computer behaviors of current system is extracted and coded by gene chip based binary. Then the presented antigen dada are sent to *DA*.

(4) *DA* firstly takes quick recognition on the presented antigen dada by memory detector. The mature detector will be used if recognition fails. *DMA* will be immediately informed if the *nonself* is found, and then AB will be optimized according to the detection results.

(5) *DMA* activates the response agent according to the danger level of intrusion. If danger signals exist but detector can't recognize *nonself*, the *CA* will communicate with other *IA* and send presented antigen dada to it to seek for effective antigen detector, then renewing the local detector set.

V. CONCLUSIONS

This paper introduces the danger theory and multi agent technique into the intrusion detection, and then presents a layered multi-agent detection model for abnormal intrusion based on danger theory. The new model provided in this paper attempts to ensure the usability of host computers and network. The model can recognize effectively the harmful self and harmful nonself, according to the hazard of outside intrusion on the host computers and network judged by cloud model. The model can decrease the probability of omission and misinformation, which ensures the usability of host computers and network, and improves detection performance of system. However, only the system variable related with system usability is considered because the cloud model can't solve completely the problems to quantize and sample the system resource variables. In the future research, further consideration on the integrity and security of system will be taken, and the types of system variable should be increased and optimized.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China under Grant 60873225, 60773191, 70771043, National High Technology Research and Development Program of China under Grant 2007AA01Z403, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE20080718.

REFERENCES

[1]. Forrest S, Perelson A S, Allen L, Cherukuri R. Self- nonself Discrimination in a Computer[C]. In: Proc IEEE Symposium on Research in Security and Privacy, Okaland, CA, 1994, pp: 202-212.

[2]. Dasgupta D. Immunity - based Intrusion Detection System: A General Framework[C]. Proc. of the 22 NISSC , 1999.

[3]. Matzinger P. Tolerance, Danger and the Extended Family[C]. Annual Review of Immunology, 1994, pp: 991 - 1045.

[4]. Aickelin U and Cayzer S (2002): The Danger Theory and Its Application to Artificial Immune Systems[C]. Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS-2002), Canterbury, UK, pp: 141-148.

[5]. Anjum Iqbal, Mohd Aizani Maarof. Towards Danger Theory Based Artificial APC Model: Novel Metaphor for Danger Susceptible Data Codons[C]. Artificial Immune Systems: Third International Conference, ICARIS 2004, Catania, Sicily, Italy, September 13-16, 2004, pp: 54-65.

[6]. Aickelin U, Bentley P, Cayzer S, et al. Danger Theory: The Link between AIS and IDS[C]. Proceedings ICARIS-2003, 2nd International Conference on Artificial Immune Systems, LNCS 2787, 2003, pp: 147-155.

[7]. Li Deyi, Meng Haijun, Shi Xuemei. Membership clouds and membership cloud generators [J]. Journal of Computer Research and Development, 1995, Vol.32, No.6, pp: 16-21.

[8]. Song Yuanyi, Li Deyi, Yang Xiaozong, et al. The cloud scheduler politics of multiprocessor multitask real time systems [J]. Chinese Journal of Computers, 2000, Vol.23, No.10, pp:1107-1113.

[9]. L i Deyi, Han J, Shi X. Knowledge Representation and Discovery Based on Linguistic Models[C]. In: Lu H J, Motoda H eds. KDD: Techniques and Applications. Singapore: World Scientific Press, 1997, pp:3-20.

[10]. Garland M，Willmott A，Heckbert P.Hierarchical face clustering on polygonal surfaces[C].In： Proceedings of ACM Symposium on Interactive 3D Graphics，Research Triangle Park，North Carolina，2001, pp:49-58.

[11]. Zhao Weiwei, Li Deyi. Intrusion detection using cloud model [J]. Computer Engineering and Applications, 2003, No.26, pp:158-160

[12]. Xu Chun, Li Tao, Liu Sunjun, et al. The research for an improved dynamic clonal selection algorithm applied to intrusion detection [J], Journal of Air Force Engineering University (Natural Science Edition), Vol.7, No.3, 2006, pp: 50-54.