Copyright © 2018 Tech Science Press CMC, vol.56, no.3, pp.381-399, 2018

Network Security Situation Awareness Framework based on Threat Intelligence

Hongbin Zhang[1, 2], Yuzi Yi[1, *], Junshe Wang[1], Ning Cao[3, *] and Qiang Duan[4]

Abstract: Network security situation awareness is an important foundation for network security management, which presents the target system security status by analyzing existing or potential cyber threats in the target system. In network offense and defense, the network security state of the target system will be affected by both offensive and defensive strategies. According to this feature, this paper proposes a network security situation awareness method using stochastic game in cloud computing environment, uses the utility of both sides of the game to quantify the network security situation value. This method analyzes the nodes based on the network security state of the target virtual machine and uses the virtual machine introspection mechanism to obtain the impact of network attacks on the target virtual machine, then dynamically evaluates the network security situation of the cloud environment based on the game process of both attack and defense. In attack prediction, cyber threat intelligence is used as an important basis for potential threat analysis. Cyber threat intelligence that is applicable to the current security state is screened through the system hierarchy fuzzy optimization method, and the potential threat of the target system is analyzed using the cyber threat intelligence obtained through screening. If there is no applicable cyber threat intelligence, using the Nash equilibrium to make predictions for the attack behavior. The experimental results show that the network security situation awareness method proposed in this paper can accurately reflect the changes in the network security situation and make predictions on the attack behavior.

Keywords: Situation awareness, stochastic game, cloud computing, virtual machine introspection, cyber threat intelligence, Nash equilibrium.

1 Introduction

With the rapid development of computer networks, network applications have penetrated

[1] School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050000, China.

[2] Hebei Key Laboratory of Network and Information Security, Hebei Normal University, Shijiazhuang, 050024, China.

[3] College of Information Engineering, Qingdao Binhai University, Qingdao, 266000, China.

[4] Department of Information Science & Technology, Pennsylvania State University, 1600 Woodland Rd. Abington, Pa, 19001, USA.

[*] Corresponding Author: Yuzi Yi. Email: 2201614041@stu.hebust.edu.cn;

Ning Cao. Email: Ning.cao2008@hotmail.com.

CMC. doi:10.3970/cmc.2018.03787 www.techscience.com/cmc

into various industries and daily life. In recent years, the rapid expansion of new network architectures such as cloud computing has further increased the scale of the network. At the same time, network security events have emerged in an endless stream, complex and targeted cyber-attack have affected many industries such as finance, energy, and medical care, caused serious security problems. Therefore, it is crucial that the detection method distinguishes accurately and timely between normal network flow and cyber-attack with limited compute resources. Early single-point detection and defense technologies are difficult to effectively analyze the synergy and the stage of cyber-attack. As the threat landscape continues to change, and with more advanced attackers than ever, security teams need all the help they can get to more effectively prevent, detect and respond to threats [Shackleford (2018)]. In order to adapt to the problems brought about by new types of cyber threats, and to assess the overall security status and security situation change trends of the network, the security situation awareness system has become a research hotspot at the present stage. The emergence of cyber threat intelligence (CTI) in recent years has brought new ideas to the study of situation awareness systems, CTI is referred to as the task of gathering evidence-based knowledge, including context, mechanisms, indicators, implications and actionable advice, about an existing or emerging menace or hazard to assets that can be used to inform decision regarding the subject's response to that menace or hazard [McMillan (2013)]. CTI describes the attack behavior, and provides context data for network attacks, and guides the defense of network attacks.

There are a large number of virtual machines (VM) in cloud computing, so the security of virtual machines is crucial to cloud computing. This paper takes the security status of VM as the situation analysis node, and uses virtual machine introspection (VMI) [Garfinkel (2003)] to monitor the running state of the target virtual machine (TVM). By analyzing the game process between attack and defense, the TVM security situation is obtained, and then achieves the network security situation of the cloud computing environment. In the prediction of attack behavior, a combination of CTI and Nash equilibrium [Nash (1951)] is used. When CTI is applicable, CTI is used as a prediction basis for attack behavior, the related context data of CTI is used to analyze the potential threats of the target system. If there is no applicable CTI, Nash equilibrium is used to predict the attack action.

The rest of this paper is organized as follows. In Section 2, we present the related work. Section 3 gives the preliminaries of this paper. Section 4 describes the use of stochastic game models to analyze network security posture. Section 5 presents a potential threat analysis method that uses CTI and Nash equilibrium. Section 6 experimentally verifies the feasibility of the proposed method. We conclude the whole paper in Section 7.

2 Related work

Research on the network security situation awareness, Bhatt et al. [Bhatt, Yano and Gustavsson (2014)] divides attacks into multiple attack stages. First, the alarm is verified through network configuration information and vulnerability information, then the validate valid alarm is matched with the known attack stage to identify the entire attack process. This attacks scenario-based method can efficiently identify the known attack behavior, but it cannot identify unknown attack behaviors. In the context of Markov models for security situation awareness, Farhadi et al. [Farhadi, Amirhaeri and Khansari (2011)] using Markov to calculate the state transition probability of the network and analyzing the attacking trend. The Markov model-based method requires that each stage of the multistep attack is continuous and has no steps to be lost, it requires longer observation sequences to train the parameters of Markov model. With the increase in the size of the network, the probability of state transition between attacks is difficult to calculate and the scalability is not ideal. In another work [Ye, Xu and Qi (2013)], the vulnerability of the target system is analyzed synthetically by constructing attack graphs, and the maximum probability of the attack path is calculated. This paper aimed at the algorithm complexity of attack graph construction, and proposed a method of target environment preprocessing, which can reduce the complexity of the algorithm in the process of attack path analysis. However, when the network environment changes, it needs to model the environment again, not well adapted to network changes. In the prediction of attack actions, Fachkha et al. [Fachkha, Bouharb and Debbabi (2013)] combining time series analysis techniques with probabilistic models, data mining and other techniques to analyze DDoS attack characteristics and behavior changes. Through the events that occur at time T, predict T+1, T+2, . . . , T+n events. Although this method can reduce the training overhead, it cannot effectively handle a large number of data sets, and the method requires strict assumptions for the data generation process. Wu et al. [Wu, Ota, Dong et al. (2016)] proposed a combination of fuzzy clustering and game theory, which improves the efficiency of forecasting, but requires high level of attack and defense modeling of the network, there are many factors to consider.

By analyzing the advantages and disadvantages of the above research, this paper proposed a security situation awareness method for cloud computing using stochastic game and CTI. The VMI is used to monitor the CPU status, memory and network information of the target virtual machine, quantifies the security situation of the cloud environment through the game process of attack and defense, and uses a combination of CTI and Nash equilibrium to predict the attack.

3 Preliminaries

Cloud computing is a computing method that provides dynamic and easily scalable virtualized resources and data to users over the Internet. Virtualization is the most important technology to support cloud computing. The concept of virtualization and virtual machines were proposed by IBM in the 1960s. It mainly aims to simplify management and optimize resources by re-planning limited and fixes resources according to different needs. According to the characteristics of cloud computing virtualization, we used VM as security situation analysis nodes and use VMI as the monitoring mechanism to collect TVM operational data.

VMI technology is a technology that obtains guest operating system (OS) bottom state information from external, the information obtained includes: CPU registers, I/O controller registers, memory, mass storage devices, and network traffic data. Through VMI technology, it is possible to effectively monitor or interfere with the guest OS running status in an Introspecting Virtual Machine (IVM). The VMI architecture is shown in Fig. 1.

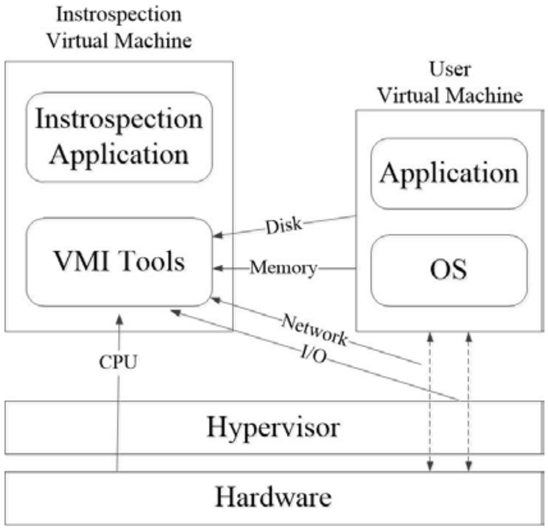

Figure 1: The architecture of VMI

IVM is highly decoupled and isolated from untrusted VM which are assumed to be unable to access or tamper the hypervisor. It has a complete view and can access to all guest OS states, and it is capable of modifying any of these states and interfering with every guest OS activity due to the interposition of the hypervisor between the guest OS and the underlying hardware [Hebbal, Laniepce and Menaud (2015)]. We use VMI to monitor CPU and memory usage, network transmission rate and delay rate, and use these data to determine the impact of attack on TVM. However, the state data obtained by using VMI is expressed in binary form. It is necessary to use the kernel data structure and other knowledge to obtain the high-level semantics of binary data, this semantic difference is called the semantic gap [Chen and Noble (2013)]. We selected LibVMI [Payne and Bryan (2012)] as an introspection tool in the existing methods. LibVMI is an introspection library that deals with this knowledge gap by providing a standard set of tools and API’s that are updated with releases of popular operating systems [Lamps, Palmer and Sprabery (2014)]. It also supports KVM hypervisor in addition to Xen and improves the overall performance by using multiple optimized caches. Notably, LibVMI integrates the popular memory forensics framework Volatility [Volatility (2018)], benefiting hence from its memory analysis capabilities [Hebbal, Laniepce and Menaud (2015)].

4 Stochastic game model

In the aspect of network security situation assessment, we selected the stochastic game model to model the network attack and defense process. The stochastic game is a dynamic game process in which one or more players participate and there is a transfer of state probabilities. In an offensive and defensive environment of the network, both offensive and defensive operations will lead to the transition of the target system's network system status, and then both offensive and defensive players continue to select an action strategy based on the new network status, and so on. The network security state reflects the impact of attack and defense strategies on the target system, and the network security state has a certain transition probability, which is consistent with the process described by the stochastic game. Therefore, we used the network attack and defense model based on stochastic game (AD-SG) to analyze the network security situation in the game phase.

Definition 1. AD-SG is a 6-tuple, $AD-SG = (P, S, A^a, A^d, U, \beta)$. The meaning of each element in the tuple is as follows:

P: It is the set of players in the game. In network attack and defense, the players are attacker P^a and defender P^d, so $P = \{P^a, P^d\}$.

S: It is the set of TVM network security status. $S = \{S_1, S_2, \dots, S_k\}$, TVM network security status is determined by both offensive and defensive strategies, among them, $S_k = S_{ij}^N$, it indicates the network state of node N when i and j are taken separate by both sides.

A^a: It is the set of attacker's optional strategies, $A^a = \{A_1^a, A_2^a, \dots, A_i^a\}$.

A^d: It is the set of defender's optional strategies, $A^d = \{A_1^d, A_2^d, \dots, A_j^d\}$.

U: It is the utility function of players, $U = \{U^a, U^d\}$. U^a indicating the attacker's utility function, U^d indicates the defender's utility function.

β: It is a status transition probability function for TVM security. It is determined by the attack success rate.

In the course of the game between the two parties, each pursuing the maximization of utility, any party adopting a strategy will produce costs and benefits, and utility is the difference between income and cost. We considered the increase in the cost of the other party's strategy as a result of its own strategy.

Attacker's utility function U^a:

$$U^a = AR - AC + DC \tag{1}$$

In the formula: AR is the reward from adopting strategies for attackers. AC is the cost of taking a strategy for attackers. DC is the cost of adopting a strategy for the defensive party.

Attack strategy reward AR:

$$AR = \beta \times AS(S_i^a) \times EA(S_i^a) \times EP(S_i^a) \tag{2}$$

Among: β is attack success rate, which derived from historical information and statistics. $AS(S_i^a)$ is the degree of damage to TVM by the attack strategy i, refer to MIT Lincoln Laboratory's privilege-enhanced multi-dimensional attack classification method [Fried, Graf, Haines et al. (2000)] to quantify the damage degree of different attacks, specific values are shown in Tab. 1. $EA(S_i^a)$ indicates the impact of the attack strategy on the CPU and memory usage of the TVM, dividing the impact level into 4 levels, corresponding to 2, 5, 8, 10. $EP(S_i^a)$ indicates the impact of the attack strategy on the network transmission rate and delay rate, also it is divided into four levels according to the degree of influence, corresponding to the value of 2, 5, 8, 10.

Table 1: Attack classification and damage degree

Classification	Description	AS
Root	Get administrator permissions	10
User	Get normal user permissions	5
Data	Unauthorized access or read and write data	3
DOS	Denial of service attack	2
Probe	Probe attack	0.5
Other	Other	*

Attack cost AC:

AC is referred to the costs incurred by an attacker when he or she takes an attack strategy, including operating costs, expertise, and the degree of sanctions that might be imposed after the attack was discovered. The greater the authority gained by the attacker or the more serious the impact on the target, the higher the operating cost and expertise cost of the attacker, and the greater the possibility of being discovered, so the higher the degree of sanctions that may be imposed, based on the above analysis, it can be seen that the AC has a positive correlation with the AS. In this paper, we let $AC = AS$, which is:

$$AC(S_i^a) = AS(S_i^a) \tag{3}$$

Defense costs DC:

According to the classification of defense strategies, the defense costs are quantified and the defense strategy is divided into: no defense measures $\emptyset$, monitoring protection measures D_S. prevent preventing measures D_F, repair protection measures D_R [Xi, Yun, Zhang et al. (2014)], the defense cost $DC\left(S_j^d\right)$ are 0, 4, 8, 10 respectively.

Through the above analysis, in the network security state in which the attacker adopts the strategy S_i^a and the defender adopts the strategy S_j^d, the attacker's utility U_{ij}^a:

$$U_{ij}^a = \beta \times AS(S_i^a) \times EA(S_i^a) \times EP(S_i^a) - AC(S_i^a) + DC\left(S_j^d\right) \tag{4}$$

In order to achieve the goal, the attacker causes losses to the target system, and the defender adopts a defensive strategy in order to reduce the loss of the system. According to the relativity of two parties' interests, a non-cooperative zero-sum game is used to describe the game process of both parties, so the defender's utility U_{ij}^d:

$$U_{ij}^d = AC(S_i^a) - \beta \times AS(S_i^a) \times EA(S_i^a) \times EP(S_i^a) - DC\left(S_j^d\right) \tag{5}$$

The utility functions U_{ij}^a and U_{ij}^d of the two sides respectively reflect the influence of the two parties' strategies on the security status of the network. In the current network security state, TVM's security situation can be expressed as $V_n = U_{ij}^d - U_{ij}^a$, the size of $|V_n|$ reflects the degree of security or dangerous state of the network. When $V_n > 0$, the network is in a secure state, and the greater the value of $|V_n|$, the more secure the network environment. When $V_n < 0$, the network is in a dangerous state. The larger the value of $|V_n|$, the more dangerous the network environment is.

The CIA security requirements model $Asset = (C, I, A)$ describes confidentiality, integrity, and availability of hosts, in this paper, confidentiality, integrity and availability are assessed according to important, general and unimportant three levels, which are 10, 5 and 1 respectively. Based on the CIA security requirements model, the attack damage D is introduced to indicate the impact of network attacks on the confidentiality, integrity, and availability of TVM. D can be represented by vectors: $D = (D_c, D_i, D_a)$, D_c, D_i, D_a respectively represent damage to confidentiality, integrity and availability, according to the degree of damage (low, medium, high) the value can be 1, 2, 3. Thus the weight of TVM can be obtained: $W_n = Asset \times D$.

The relative weight W_n' of node n can be represented as:

$$W_n' = W_n / \sum_{n}^{N} W_n \tag{6}$$

According to the above analysis, the security situation S of the cloud computing can be comprehensively represented by the security status of each node:

$$S = \sum_{n=1}^{N} W_n' \times V_n \tag{7}$$

Similarly, the size of $|S|$ reflects the degree of network security or dangerous state. When $S > 0$, the network is in a safe state. When $S < 0$, the network is in a dangerous state.

5 Potential threat analysis

In this paper, we use CTI and Nash equilibrium to analyze the potential threat in the target system and predict the attack. When the CTI exists the context-related data of security events in the target system, the context data is used as the basis for the attack behavior prediction. When the CTI is not applicable, the Nash equilibrium is used to predict the attack behavior. This section will describe the attack prediction methods in these two situations.

5.1 Attack prediction using threat intelligence

CTI includes a large amount of security event information. However, not all security event information is applicable to the current system state. In order to improve data accuracy and obtain contextual data related to security events, the concept of high-quality CTI is introduced in this paper, simultaneously, using system hierarchy fuzzy optimization method to obtain high-quality CTI. The definitions of internal CTI, external CTI and high-quality CTI are as follows:

Definition 2. Internal cyber threat intelligence (ICTI). It is derived from the security event information in the target system and is obtained by integrating relevant data in security devices such as security information and event management (SIEM) and intrusion detection systems (IDS).

Definition 3. External cyber threat intelligence (ECTI). It refers to the open source intelligence (OSINT) or the CTI provided by intelligence providers.

Definition 4. High-quality cyber threat intelligence (HCTI). The ECTI which exists contextual data or related information of security events in the target system, and it is of guiding significance to defense.

To accurately process CTI data, unified ICTI and ECTI formats before analyzing CTI. In this paper, we select Structured Threat Information Expression (STIX) [Barnum (2014)] as the ICTI and ECTI format. STIX is a language and serialization format for the exchange of CTI. It consists of several cyber threat information, such as network observables, indicators, events, tactics, techniques and procedures (TTP), attack targets, and threat initiation.

Several objects in the STIX are selected as the CTI analysis elements, and these analysis elements are used as CTI screening objects. We selected several key properties as analysis elements in five objects: Indicator, malware, observed data, tool, and vulnerability.

1) Indicators contain a pattern that can be used to detect suspicious or malicious cyber activity. we use labels and patterns in the indicator as analysis elements.

2) Malware is a type of TTP that is also known as malicious code and malicious software, using name and labels in the malware as one of the analysis elements.

3) Observed Data conveys information that was observed on systems and networks.

4) Tools are legitimate software that can be used by threat actors to perform attacks. Knowing how and when threat actors use such tools can be important for understanding how campaigns are executed.

5) A Vulnerability is “a mistake in software that can be directly used by a hacker to gain access to a system or network” [Surhone, Tennoe and Henssonow et al. (2010)].

The selected Objects and their properties are shown in Tab. 2:

Table 2: The selected Objects and their properties

Objects	Properties	Description
Indicator	labels	It is used to categorize indicators
	pattern	The detection pattern for indicators
Malware	name	A name used to identify the Malware sample
	labels	The type of malware being described
Observed Data	objects	The observed data in security events
Tool	name	The name used to identify the Tool
	labels	The kind(s) of tool(s) being described
Vulnerability	CVE-id	The Vulnerability identifiers

The relationship between these Objects are shown in Fig. 2:

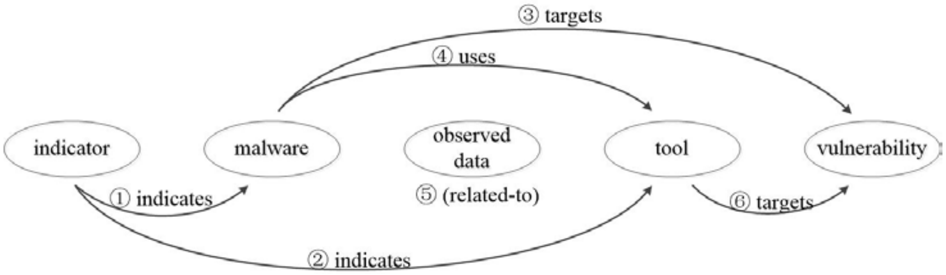

Figure 2: The relationship structure of the selected Objects

The Relationship ① and ② describes that the Indicator can detect evidence of the related malware and tool.

The Relationship ③ documents that this Malware being used to exploit the Vulnerability. The Relationship ④ documents that this Malware uses the related tool to perform its functions.

The relationship ⑤ meanings there are no relationships explicitly defined between the Observed Data object and other objects.

The Relationship ⑥ documents that this Tool being used to exploit the Vulnerability.

It is easy to determine whether the individual properties of the corresponding objects in the ICTI and ECTI are equal. Because of the close relationship between objects, in some cases, same objects in the different relationship can express different security events. Therefore, judging the matching degree of ICTI and ECTI is inaccurate by whether the properties are equal or not. The relationship between objects makes the ICTI and ECTI has fuzzy similarity. According to the hierarchical relationship between Objects and Properties in CTI, this paper adopts the system hierarchy fuzzy optimization method and uses the relative superiority degree of the target to judge the matching degree of ICTI and ECTI. Here is how to use the system hierarchy fuzzy optimization method to obtain HCTI:

1) First of all, the ECTI is classified, the probability that the security event in the target system completely matches with an ECTI is low. And the same type of CTI contains more abundant information, so the same kind of CTI can used to analyze the follow-up security events, in this paper, we use the CAPEC-id [The MITRE Corporation (2011)] of the attack pattern in CTI as the CTI classification standard.

2) Counting the objects' occurrence frequency in ECTI, using frequency as an element in the eigenvalue matrix, and set weights on indicators at all levels. Tab. 3 shows an example of data statistics.

Table 3: Objects and Properties weights and frequency of occurrence

Objects		Properties		Frequency of occurrence			
Name	W	Name	W'	Classification of ETI			
				ID 1	ID 2	ID 3	ID n
Indicator	w_1	labels	w_{11}	x_{11}	x_{12}	x_{13}	x_{1n}
		pattern	w_{12}	x_{21}	x_{22}	x_{23}	x_{2n}
Malware	w_2	name	w_{21}	x_{31}	x_{32}	x_{33}	x_{3n}
		labels	w_{22}	x_{41}	x_{42}	x_{43}	x_{4n}
Observed Data	w_3	objects	w_{31}	x_{51}	x_{52}	x_{53}	x_{5n}
Tool	w_4	name	w_{41}	x_{61}	x_{62}	x_{63}	x_{6n}
		labels	w_{42}	x_{71}	x_{72}	x_{73}	x_{7n}
Vulnerability	w_5	CVE-id	w_{51}	x_{81}	x_{82}	x_{83}	x_{8n}

In Tab. 3, Objects represent five subsystems, properties represent evaluation factors under each system, W is the weight of objects, W' is the weight of properties, X_{mn} is the frequency of occurrence of ICTI in ECTI, $m = 1,2,3, \ldots ,8$.

Let the subsystem i contain m_i evaluation factors, and the feature value vector of evaluation factor j is:

$$x_j = (x_{1j}, x_{2j}, \ldots, x_{m_i j})^T \tag{8}$$

Then, the eigenvalue matrix of the evaluation factors of the n items to be optimized for the subsystem i is represented as:

$$X_{m_i \times n}(i) = \begin{bmatrix} x_{11(i)} & x_{12(i)} & \cdots & x_{1n(i)} \\ x_{21(i)} & x_{22(i)} & \cdots & x_{2n(i)} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m_i 1(i)} & x_{m_i 2(i)} & \cdots & x_{m_i n(i)} \end{bmatrix} = (x_{kj}) \tag{9}$$

In the formula, $i = 1,2,3,4,5$; $k = 1,2, \ldots, m_i$; $j = 1,2, \ldots, n$.

3) Convert Eq. (9) to the target relative dominance degree matrix:

$$R_{m_i \times n}(i) = \begin{bmatrix} r_{11(i)} & r_{12(i)} & \cdots & r_{1n(i)} \\ r_{21(i)} & r_{22(i)} & \cdots & r_{2n(i)} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m_i 1(i)} & r_{m_i 2(i)} & \cdots & r_{m_i n(i)} \end{bmatrix} = (r_{kj}) \tag{10}$$

In CTI matching, the higher the frequency, the better the result, so in the target relative dominance degree matrix, r_{ij} can be obtained using Eq. (11):

$$r_{ij} = \frac{x_{ij} - \wedge_j x_{ij}}{\vee_j x_{ij} - \wedge_j x_{ij}} \tag{11}$$

In the formula, $\vee$ and $\wedge$ represent take large symbols and small symbols respectively, $\vee_j x_{ij}$, $\wedge_j x_{ij}$ represent the maximum and minimum eigenvalues of the target i respectively, $j = 1,2,3,\dots,n$.

4) Let the weight vector of m_i evaluation factors' in the subsystem i be:

$$W(i) = (w_{1(i)}, w_{2(i)}, \dots, w_{m_i(i)}) \tag{12}$$

Among, $\sum_{k=1}^{m_i} w_{k(i)} = 1$, $w_{k(i)}$ is the weight of the factor k for the subsystem i.

The target relative degree of superiority $u_{j(i)}$ of the subsystem i can be expressed as:

$$u_{j(i)} = \frac{1}{1 + \left\{\dfrac{\sum_{k=1}^{m_i}[w_{k(i)}(g_{k(i)} - r_{kj(i)})]^p}{\sum_{k=1}^{m_i}[w_{k(i)}(r_{kj(i)} - b_{k(i)})]^p}\right\}^{2/p}} \tag{13}$$

In the formula, $i = 1,2,3,4,5$; $k = 1,2,\dots,m_i$; $j = 1,2,\dots,n$; $g_{k(i)} = \vee_{j=1}^{n} r_{kj(i)}$; $b_{k(i)} = \wedge_{j=1}^{n} r_{kj(i)}$; p is the distance parameter, using Euclidean distance, at this time $p = 2$.

This results in the superiority vector of n schemes in the system i:

$$U(i) = (u_{1(i)}, u_{2(i)}, \dots, u_{m_i(i)}) \tag{14}$$

In the formula, $i = 1,2,3,4,5$.

5) The output of the unit system constitutes the input of the high-level unit system. Make:

$$(r_{ij}) = (u_{j(i)}) \tag{15}$$

There are:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \tag{16}$$

At this time:

$$u_j = \frac{1}{1 + \left\{\dfrac{\sum_{i=1}^{m}[w_i(g_i - r_{ij})]^p}{\sum_{i=1}^{m}[w_i(r_{ij} - b_i)]^p}\right\}^{2/p}} \tag{17}$$

This gives n target relative superiority vectors for high-level (Objects) unit systems:

$$U = (u_1, u_2, \dots, u_n) \tag{18}$$

6) According to the principle of maximum degree of superiority, the results of Eq. (18) can be used to analyze superior goals. But we can see from the analysis of the physical meaning of the fuzzy optimization model [Li (2016)], When $u_n > 0.5$, solution n has the necessity to participate in optimization, that is, if $\exists\, u_i > 0.5$, $max(u_i)$ is the target relative degree of superiority. In this paper, we consider $max(u_i)$ as HCTI.

5.2 Prediction of attack actions using nash equilibrium

Nash equilibrium means that the strategy of a player is an optimal response to the strategies of others. For every player, as long as other player does not change his strategy, he cannot improve his situation. In the network environment, the available vulnerabilities of the target system are limited. Therefore, the available policies of the two parties are also limited. The limited strategy determines that the transferable network security status is also limited. According to the Nash equilibrium existence theorem [Nash (1950)], there is an equilibrium point in the network offense and defense game model. Under the premise of a rational choice between both sides of the offensive and defensive sides, both parties hope to obtain the maximum benefit at the minimum cost. So the two sides will choose countermeasures according to each other's tactics, and the best countermeasure will form a Nash equilibrium.

Since the strategies adopted by both parties are not clear and unique, the Nash equilibrium under the hybrid strategy is used to express the two parties' strategies in the form of probability. The attacker selects a strategy with a probability distribution of $P_a=(P_{a1},P_{a2},\dots,P_{am})$, the defender selects a strategy with a probability distribution of $P_d=(P_{d1},P_{d2},\dots,P_{dn})$. Under the mixed strategy, the two parties' profit expectation is:

$$E_a(P_a,P_d)=\sum_{i}^{m}P_{ai}\,[\sum_{j}^{n}P_{dj}U_a(S_i^a,S_j^d)]=\sum_{i}^{m}\sum_{j}^{n}P_{ai}P_{dj}U_a(S_i^a,S_j^d) \tag{19}$$

$$E_d(P_a,P_d)=\sum_{j}^{n}P_{dj}\,[\sum_{i}^{m}P_{ai}U_d(S_i^a,S_j^d)]=\sum_{j}^{n}\sum_{i}^{m}P_{ai}P_{dj}U_d(S_i^a,S_j^d) \tag{20}$$

From the above analysis, it can be known that the network offense and defense game model have a mixed strategy (P_a^*,P_d^*) to reach the Nash equilibrium, where (P_a^*,P_d^*) satisfies:

$$\begin{cases}\forall\, P_{ai},\sum_{i=1}^{m}\sum_{j=1}^{n}U_a(S_i^a,S_j^d)P_{ai}^*P_{dj}^*\geq\sum_{i=1}^{m}\sum_{j=1}^{n}U_a(S_i^a,S_j^d)P_{ai}P_{dj}^*\\ \forall\, P_{dj},\sum_{i=1}^{m}\sum_{j=1}^{n}U_d(S_i^a,S_j^d)P_{ai}^*P_{dj}^*\geq\sum_{i=1}^{m}\sum_{j=1}^{n}U_d(S_i^a,S_j^d)P_{ai}^*P_{dj}\\ \sum_{i=1}^{m}P_{ai}=1,P_{ai}\geq 0\\ \sum_{j=1}^{n}P_{dj}=1,P_{dj}\geq 0\end{cases} \tag{21}$$

In summary, mixed strategy $P_{ai}^*=(P_{a1}^*,P_{a2}^*,\dots,P_{am}^*)$ is the best choice for the attacker and is the most likely strategy that the attacker adopts. The defender can take defensive measures based on the attacker's optimal strategy.

6 Experimental verification

Using the LLDOS1.0(inside) from the MIT Lincoln Laboratory's DARPA2000 as the experimental data set and the security event information of the data set is made into CTI to verify the system hierarchy fuzzy optimization method. Since the data set does not include defense measures, we added a demilitarized zone (DMZ) to the original network topology of the data set, according to the vulnerability information and defense measures of the servers in the DMZ, the Nash equilibrium attack prediction method is verified. The experimental network topology is shown in Fig. 3. In this section, the content of the experiment is described in detail.

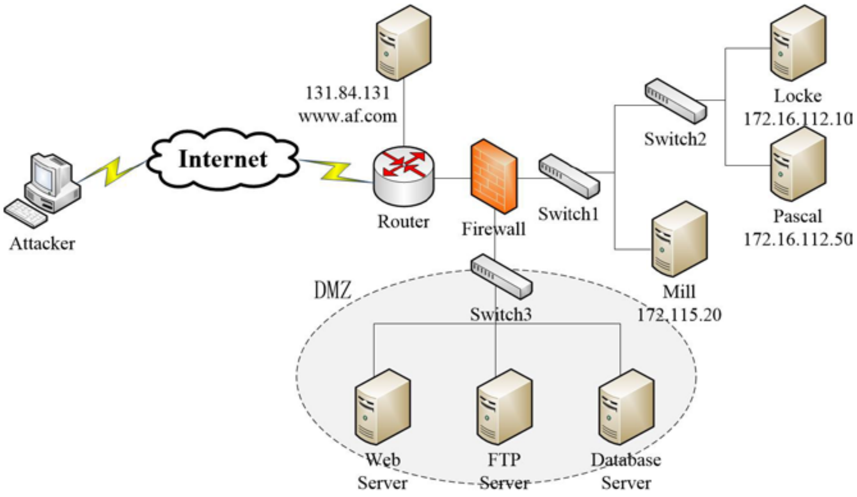

Figure 3: Network topology

6.1 Network security situation assessment

LLDOS 1.0 includes a complete distributed denial of service attack scenario. The attack is divided into five phases:

1) IPsweep from a remote site

2) Probe of live IP’s to look for the sadmind daemon running on Solaris hosts

3) Breakins via the sadmind vulnerability, both successful and unsuccessful on those hosts

4) Installation of the trojan mstream DDoS software on three hosts.

5) Launching the DDoS

In LLDOS 1.0, the attacker successfully invaded three hosts, they are mill (172.16.115.20), pascal (172.16.112.50), and locke (172.16.112.10), and use these three hosts to launch a DDOS attack on the target host www.af.mil (131.84.1.31). As the attacker took the exact same intrusion means for three hosts, we used mill (172.16.115.20) as an example to analyze the security situation of a single host. Using wireshark to screen mill's relevant network traffic and import traffic into snort to get alarm information: Mill was attacked by IPsweep at 9:51:36 and Sadmind Ping attacked at 10:08:07. The attacker used the remote buffer overflow attack to invade the host after determining that mill running the sadmind service, after several attempts, he successfully obtained root authority at 10:33:29, then established a connection with mill through Telnet, and installed DDOS software on the host. Since the data set does not contain any defense information, the

AD-SG model proposed in this paper is used to quantify the utility of the attacker to obtain the network security situation value. Mill's network posture values and situation changes are shown in Fig. 4.

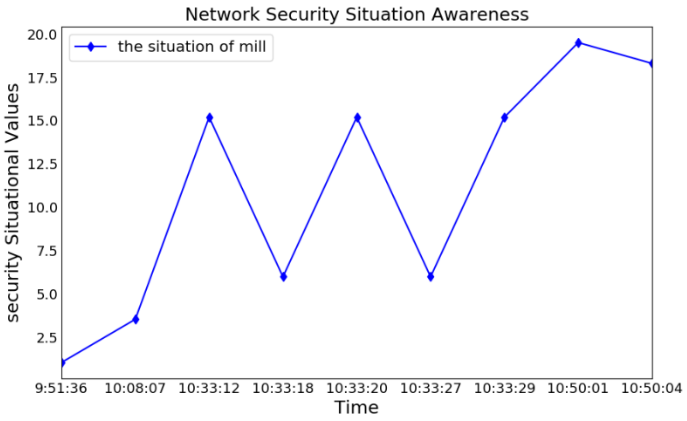

Figure 4: Mill's network security situation

In the overall network environment security situation analysis, weights are assigned to each host, and the CIA weights for mill, pascal, locke, and www.af.com are $Asset_1 = (5,5,5)$, $Asset_2 = (5,5,10)$, $Asset_3 = (1,5,10)$ and $Asset_4 = (5,5,10)$. Since each host is attacked by the same type of attacks during the same attack phase, the attack damages are divided into stages. In the first phase to the fifth phase, the attack damages are $D_1 = (2,1,1)$, $D_2 = (2,1,1)$, $D_3 = (2,1,2)$, $D_4 = (3,2,2)$, $D_5 = (1,2,3)$. After assigning weights, calculating the security situation of the entire network environment. The results are shown in Fig. 5.

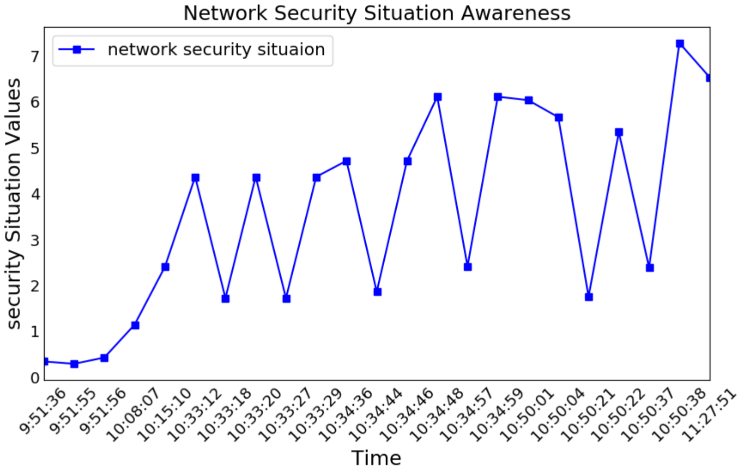

Figure 5: Security situation in the network environment

Fig. 5 reflects the impact of the attack on the network environment. It can be seen that the probe attack has little impact on the network environment, subsequent buffer overflow attacks increasing the network security situation further. The attacker successfully obtained the root privileges of the three hosts poses a greater threat to the network environment. After installing the DDOS tools on these three hosts and taking an attack on the target, the availability of the target is affected, as the threats of the other three hosts have not been lifted, the security situation of the entire network environment has further increased. From the experimental results, we can see that the method proposed in this paper can reflect the impact of the attack on the network security situation.

6.2 Attack prediction

In the following content, we will introduce the both case of existing HCTI and does not existing HCTI. In the presence of HCTI, the system hierarchy fuzzy optimization method was highlighted, the HCTI obtained by this method can be used as a basis for attack prediction. In the absence of HCTI, predicting of attacking behavior in DMZ using Nash equilibrium.

6.2.1 Threat intelligence extraction

The security event described by the data source can be divided into two parts: the attacker invades intranet hosts and uses hosts to launch the DDOS attack. The first part consists of phase 1, 2, 3 and 4, the second part consists of phase 5 alone. In this paper, the information of the first part is chosen as the preferred object of the ECTI, and the potential threat is analyzed through the optimization results, that is, the information of the second part.

Make the information of the first part into ICTI and add it to the ECTI, The ECTI used in the experiment can be divided into four categories according to CAPEC-id: CAPEC-24, CAPEC-47, CAPEC-185, CAPEC-122, the data source used for the experiment belongs to the CAPEC-47 category.

Table 4: Evaluation factors weights and frequency of occurrence

Objects		Properties		Frequency of occurrence			
Name	W	Name	W’	Classification of External TI			
				24	47	185	122
Indicator	0.2	labels	0.4	9	6	0	8
		pattern	0.6	2	3	1	2
Malware	0.2	name	0.35	2	1	0	0
		labels	0.65	11	8	0	4
Observed Data	0.2	objects	1	1	3	1	1
Tool	0.15	name	0.35	2	1	1	2
		labels	0.65	2	3	2	2
Vulnerability	0.25	CVE-id	1	2	1	0	1

The weights and frequency of evaluation factors are shown in Tab. 4.

According to Tab. 4, the evaluation characteristic matrix of the subsystem indicator is:

$$X_{2_1\times 4}(1) = \begin{bmatrix} 9_{(1)} & 6_{(1)} & 0_{(1)} & 8_{(1)} \\ 2_{(1)} & 3_{(1)} & 1_{(1)} & 2_{(1)} \end{bmatrix} \tag{22}$$

The target relative degree of identity matrix obtained by Eq. (11) is:

$$R(1) = \begin{bmatrix} 1_{(1)} & 0.67_{(1)} & 0_{(1)} & 0.89_{(1)} \\ 0.5_{(1)} & 1_{(1)} & 0_{(1)} & 0.5_{(1)} \end{bmatrix} \tag{23}$$

The weight vector for the subsystem indicator is:

$$\mathrm{W}'(1) = (0.4_{(1)}, 0.6_{(1)})^T \tag{24}$$

From the Eq. (13), the target relative degree of superiority vector for the subsystem indicator is:

$$U(1) = (0.73_{(1)}, 0.96_{(1)}, 0_{(1)}, 0.70_{(1)}) \tag{25}$$

Similarly, by calculating the subsystems of malware, observed data, tool, and vulnerability, the target relative degree of membership of high-level cells can be obtained:

$$R = \begin{bmatrix} 0.73 & 0.96 & 0 & 0.70 \\ 1 & 0.81 & 0 & 0.15 \\ 0 & 1 & 0 & 0 \\ 0.22 & 0.78 & 0 & 0.22 \\ 1 & 0.5 & 0 & 0.5 \end{bmatrix} \tag{26}$$

The weight vector of the second-level subsystems is:

$$\mathrm{W} = (0.2, 0.2, 0.2, 0.15, 0.25)^T \tag{27}$$

From Eq. (17), the target relative superiority vector of the ICTI and ECTI can be obtained as:

$$U = (0.70, 0.85, 0, 0.29) \tag{28}$$

Fig. 6 shows the results of the ECTI screening using system hierarchy fuzzy optimization method.

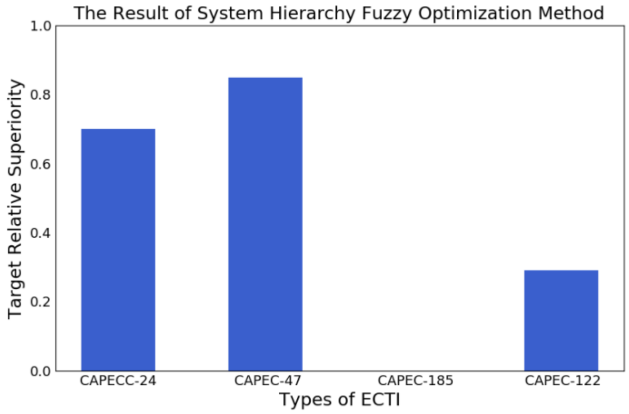

Figure 6: The result of system hierarchy fuzzy optimization method

According to the experimental results and the principle of higher priority, it can be seen that the CAPE-47 type ECTI has the greatest reference value, and the experimental results are in line with the actual situation. The closest to its superiority is CAPEC-24, because both security events are buffer overflow. Therefore, some data in security events, especially labels, will have the same situation. Although it does not reach the maximum degree of superiority, it still has some reference value. CAPEC-122 is privilege abuse security events, in the target security event, the attacker tries a certain number of Telnet connections by obtaining the root privileges of internal hosts. Therefore, the ICTI and ECTI contains some of the same data, but the relative degree of superiority is 0.29 and does not reach the threshold. So this type of CTI is not considered as reference value. CAPEC-185 is malicious software download security events, although individual data is the same as ICTI, the overall difference is too great. Therefore, the degree of superiority is 0, which does not have any reference value.

6.2.2 Nash equilibrium

In the network environment shown in Fig. 3, Web Server, FTP Server, and Database Server are located in the DMZ. The attacker is located in the external network. The firewall allows external hosts to access the Web Server and FTP Server. Only the Web Server and FTP Server can access the Database Server.

The vulnerability information of the three servers is shown in Tab. 5. According to the vulnerability information, the defensive party's optional defense measures are shown in Tab. 6.

Table 5: Vulnerability information

Host	Vulnerability information	Result	AC
FTP Server	1. Ftp.rhosts	User	5
Web Server	2. Apache Chunked Enc.	Root	10
	3. Wu-Ftpd SockPrintf()	Root	10
Database Server	4. Oracle TelCommand Execute	Root	10

Table 6: Defensive strategy

Defensive strategy	Strategy category	DC
No access to the port	D_F	8
Install Apache patch	D_R	10
Install Oracle patch	D_R	10
Stop the FTP service	D_F	8

According to the optional strategy of offense and defense, obtain the utility matrix of both parties through Eqs. (4) and (5):

$$\begin{bmatrix} 26.8,-26.8 & 133,-133 & 80.5,-80.5 \\ 62,-62 & 54,-54 & 115.5,-115.5 \\ 30.5,-30.5 & 129.6,-129.6 & 145.2,-145.2 \\ 14.9,-14.9 & 151.9,-151.9 & 116.8,-116.8 \end{bmatrix} \tag{29}$$

Calculating the Nash Equilibrium to obtain a mixed strategy probability distribution for both sides: $P_a^* = (0, 0.31, 0.69)$, $P_d^* = (0.54, 0.46, 0, 0)$. The mixed strategy $P_a^* = (0, 0.31, 0.69)$ is a prediction of attack behavior. According to the prediction, the attacker's most likely strategy is to exploit the Wu-Ftpd SockPrintf() vulnerability.

7 Conclusion and future work

This paper proposes a situational awareness method in cloud computing environment. With TVM's network security status as the analysis node, the impact of attack behavior on TVM is obtained by VMI, through the game gains of both parties, the network security situation of cloud environment is obtained. In the situation prediction, the CTI and Nash equilibrium are combined to predict the attack behavior. The CTI context data provides real security event information and has a high reference value; when the CTI is not applicable, analyze offensive and defensive alternative strategies, using Nash equilibrium to predict attack behavior, so as to make up for the absence of relevant contextual data in CTI. In the next step, the proposed method is applied to the real environment, and the deficiencies in verification are improved.

Acknowledgement: This research was supported in part by the National Natural Science Foundation of China under grant numbers 61672206, 61572170.

References

Barnum, S. (2014): *Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX^{TM}).* MITRE Corporation.

Bhatt, P.; Yano, E. T.; Gustavsson, P. M. (2014): Towards a framework to detect multi-stage advanced persistent threats attacks. *International Symposium on Service Oriented System Engineering*, pp. 390-395.

Chen, P. M.; Noble, B. D. (2013): When virtual is better than real [operating system relocation to virtual machines]. *Workshop on Hot Topics in Operating Systems*, pp. 133-138.

Fachkha, C.; Bouharb, E.; Debbabi, M. (2013): Towards a forecasting model for distributed denial of service activities. *IEEE International Symposium on Network Computing and Applications*, pp. 110-117.

Farhadi, H.; Amirhaeri, M.; Khansari, M. (2011): Alert correlation and prediction using data mining and HMM. *iSeCure*, vol. 3, no. 2, pp. 77-101.

Garfinkel, T. (2003): *A Virtual Machine Introspection Based Architecture for Intrusion Detection (Ph.D. Thesis)*. Computer Science Department, Stanford University.

Hebbal, Y.; Laniepce, S.; Menaud, J. M. (2015): Virtual machine introspection: Techniques and applications. *International Conference on Availability, Reliability and Security*, pp. 676-685.

Lamps, J.; Palmer, I.; Sprabery, R. (2014): WinWizard: Expanding Xen with a LibVMI intrusion detection tool. *International Conference on Cloud Computing,* pp. 849-856.

Li, C. X. (2016): *Fuzzy Mathematics Method and Application.* Chemical Industry Press, Beijing, China.

Lippmann, R. P.; Fried, D. J.; Zissman, M. A.; Graf, I.; Haines, J. W. et al. (2002). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *DARPA Information Survivability Conference and Exposition*, vol. 2, pp. 12-26.

McMillan, R. (2013): *Definition: Threat Intelligence*. Gartner.

Nash, **J.** (1950): Equilibrium points in n-person games. *Proceeding of the National Academy of Science*, vol. 36, pp. 48-49.

Nash, J. (1951): Non-cooperative games. *Annals of Mathematics*, vol. 54, no. 2, pp. 286-295.

Payne, Bryan, D. (2012): Simplifying virtual machine introspection using LibVMI. *Office of Scientific & Technical Information Technical Reports*, pp. 1-20.

Shackleford, D. (2018): *CTI in Security Operations: SANS 2018 Cyber Threat Intelligence Survey*. SANS Institute.

Surhone, L. M.; Tennoe, M. T.; Henssonow, S. F.; Corporation, M. (2010): *Common Vulnerabilities and Exposures*. Betascript Publishing.

The MITRE Corporation (2011): Common attack pattern enumeration and classification (CAPEC). http://capec. mitre.org/.

Volatility (2018)**:** Volatility Framework. http://www.volatilityfoundation.org/.

Wu, J.; Ota, K.; Dong, M.; Li, J.; Wang, H. (2016): Big data analysis based security situational awareness for smart grid. *IEEE Transactions on Big Data*, pp. 346-350.

Xi, R. R.; Yun, X. C.; Zhang, Y. Z.; Hao, Z. Y. (2014): An improved quantitative evaluation method for network security. *Chinese Journal of Computers*, vol. 38, no. 4, pp. 749-758.

Ye, Y.; Xu, X. S.; Qi, Z. C. (2013): Attack graph generation algorithm for large-scale network system. *Journal of Computer Research and Development*, vol. 50, no. 10, pp. 2133-2139.