Hindawi
Security and Communication Networks
Volume 2018, Article ID 7412627, 13 pages
https://doi.org/10.1155/2018/7412627

Research Article

Security Measurement for Unknown Threats Based on Attack Preferences

Lihua Yin,[1] Yanwei Sun ,[2,3] Zhen Wang,[4] Yunchuan Guo ,[2] Fenghua Li,[2,3] and Binxing Fang[5]

[1]*Cyberspace Institute of Advanced Technology (CIAT), Guangzhou University, Guangzhou, China*
[2]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[3]*School of Cyberspace Security, University of Chinese Academy of Sciences, Beijing, China*
[4]*School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China*
[5]*Institute of Electronic and Information Engineering of UESTC in Guangdong, Dongguan, China*

Correspondence should be addressed to Yunchuan Guo; guoyunchuan@iie.ac.cn

Received 12 January 2018; Revised 14 March 2018; Accepted 10 April 2018; Published 20 May 2018

Academic Editor: Huaizhi Li

Security measurement matters to every stakeholder in network security. It provides security practitioners the exact security awareness. However, most of the works are not applicable to the unknown threat. What is more, existing efforts on security metric mainly focus on the ease of certain attack from a theoretical point of view, ignoring the "likelihood of exploitation." To help administrator have a better understanding, we analyze the behavior of attackers who exploit the zero-day vulnerabilities and predict their attack timing. Based on the prediction, we propose a method of security measurement. In detail, we compute the optimal attack timing from the perspective of attacker, using a long-term game to estimate the risk of being found and then choose the optimal timing based on the risk and profit. We design a learning strategy to model the information sharing mechanism among multiattackers and use spatial structure to model the long-term process. After calculating the Nash equilibrium for each subgame, we consider the likelihood of being attacked for each node as the security metric result. The experiment results show the efficiency of our approach.

1. Introduction

Security measurement matters to every stakeholder in network security and involves all the stages and aspects of the entire life cycle. There would be no effective security awareness and actions without accurate security measurement. The existing security measurements mainly focus on the relationship between exploits and system vulnerabilities, and their security measurements of unknown threats like zero-day loophole are very limited. In addition, zero-day attacks targeting governments and corporates are growing with time. An increasing number of hackers, motivated by their persistent love for technology or tempted by profits, are attempting to discover and propagate zero-day exploits. In 2016, 10822 vulnerabilities were found in China, and 2203 of them were zero-day vulnerabilities (http://www.cert.org.cn/publish/main/upload/File/2016CNVDannual1.pdf), which may cause serious consequences. According to *The Hacker News*, hackers exploited the zero-day vulnerability to attack Bangladesh's central bank in 2016 and stole over $80 million from the Federal Reserve Bank (http://thehackernews.com/2016/03/bank-hacking-malware.html). In such case, it poses a great challenge as to carry out effective measurements of threats posed by zero-day vulnerabilities to help system administrators better understand and guard against them.

Current security measurements are mostly around known vulnerabilities. They get the result of the measurement after analyzing the attacks and coming up with corresponding rules. Such measurement can be a distortion from real-life situation. For example, some vulnerabilities proven highly threatening according to CVSS' measurement are not actually exploited too much by the attackers. The work proposed by Wang et al. [1] has similar problems; this is one of the few measurement works targeting hidden vulnerability. The

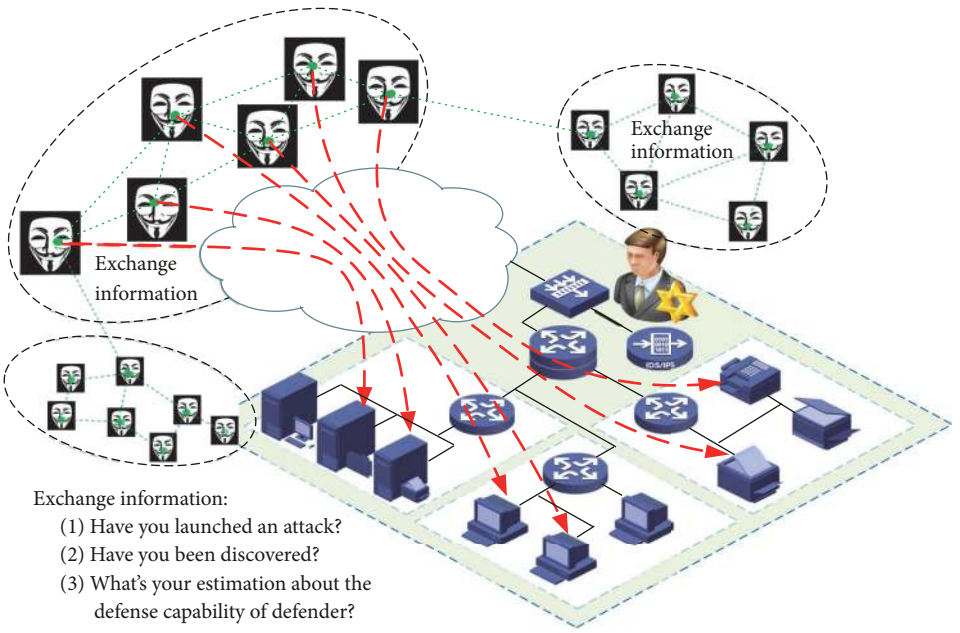


Figure 1: Multiattackers versus a certain company.

paper uses attack graph to analyze and decide the minimal number of vulnerabilities needed to achieve a set goal, assuming that there is a hidden vulnerability on any of the nodes. The number will be used as a reference for security measurements. This study measures only from the dimension of complexity, not taking the differences in the attackers' preference, and the result is only a theoretical one, so there is not too much reference value for the result. For example, if there is a network with weak protection and only one hole is needed to break into the network, then the measurement result should be highly risky according to the method above. However, the attackers will not attack a network that they deem of no value; Honeynet is a case in point. Therefore, when measuring unknown threats, we need to include the attacking decision of the attacker as a significant metric and take the degree of complexity, possible risks, benefits, and so on into consideration. This way, we can analyze the decision the attacker is most likely to take and get a more realistic measurement result. That is to say, if, during a certain period, the attackers are eager to attack a node, then we will think this node is facing serious threats at this time; otherwise, the result should not be labeled as highly threatening.

The security measurement of unknown threats with the attack behavior preferences in mind faces the following three major challenges:

(i) Modeling zero-day attacks from the time dimension: current studies express unknown vulnerabilities using attack graph, which are mainly from the spatial dimension. But zero-day attack is a long-term process. The zero-day exploit (also called cyber resources) will always be available unless the vulnerability is fixed by the defender. As the process goes on, the attacker needs to make a tradeoff between risk and profit for each time point: if the vulnerability is exploited, the attacker may get some profit but the risk of being noticed may also increase. On the contrary, if the zero-day vulnerability is exploited too late or not exploited, there is also a chance for the defender to fix it, leaving no chances for the attacker. So the first challenge is how to model this process properly from the dimension of time.

(ii) Identify and calculate the factors that change over time: in the course of the zero-day attack and defense, the factors that influence the attacker's decision-making include the potential profits, the attacker's knowledge of the defending party, and the risks related to the attack. A comprehensive and reasonable expression and calculation of these factors are the basis for predicting attack decision.

(iii) Predict the attack decision and measure the unknown threat: after setting the relevant parameters, how to use these parameters to accurately predict the attacker's possible decision at each time point determines the accuracy and reliability of the entire security measurement.

To overcome these challenges, this paper, which is based on our previous work [2], uses long-term game theory to predict the behavior of the attacker (the attack-defense scenario is shown in Figure 1) and then propose a new security metric based on the prediction. The main contributions are as follows:

(i) We present a multiple game model: we consider zero-day attacks as a long-term process. The attacker's decision (attack or wait) at any point in time will have a corresponding effect on the later game process. Therefore, we discuss the continuous game process in a certain period of time.

(ii) We discuss the factors that affect the attack and defense: we discuss the factors that affect the attack and defense, particularly the change of the attacker's understanding of the defensive ability of the defender. At the beginning of the game, attackers know little about the defender. But with the game process, their understandings become more and more accurate due to their observation and information sharing. We design a learning mechanism to simulate the correction process.

(iii) We propose a new security metric based on attack prediction: from the attacker's point of view, we calculate the Nash equilibria for each subgame. Using the result as an important reference, we propose a new security metric for unknown threat.

The rest of the paper is organized as follows. Section 2 surveys the related work. Section 3 introduces the preliminaries. Section 4 introduces long-term game formulation. Section 5 discusses the details about the security measurement method. Section 6 reports experimental results, and Section 7 gives the conclusions.

2. Related Work

There have been plenty of research studies on network security metrics. They focus on different aspects, such as metrics of system vulnerabilities, metrics of defense power, metrics of situations, and metrics of attack or threat severity [3]. In detail, when assessing the risk of malware threat, Hardy et al. [4] proposed the targeted threat index combining the social engineering and technical sophistication. Thakore [5] provided a set of metrics such as coverage, redundancy, confidence, and cost to quantitatively evaluate monitor deployments. Kührer et al. [6] focused on the effectiveness of malware blacklists and showed that the current blacklist is insufficient to protect against the variety of malware threats. There are other studies focused on evaluating the strength of IDS or other security products [7, 8], strength of user password [9, 10], and so on. However, the effects of all the metrics mentioned above are not ideal when faced with unknown threats.

In order to make a better understanding of zero-day attack, some researches focus on analyzing the attack itself such as detecting and identifying the attack. To identify the unknown files, Avasarala et al. [11] introduced the class-matching approach. Mishra and Gupta [12] proposed a hybrid solution which uses the concept of CSS matching and URI matching to defend against zero-day phishing attacks. Wang et al. proposed some representative works on measuring the zero-day attack [1, 13, 14]. To evaluate the robustness of networks, [13, 14] modeled network diversity as a security metric and then proposed two complementary diversity metrics. The paper [1] conducted the evaluation process based on how many zero-day vulnerabilities are required to compromise a network asset. However, all these works are conducted based on attack graph and do not consider the attacker behavior.

Table 1: Payoff in the two-player one-shot game.

	Protect	Not protect
Attack	$-C_a, -C_d$	$G - C_a, -G$
Not attack	$0, -C_d$	$0, 0$

Analyzing the attacker behavior is of great importance when measuring the network security. Ekelhart et al. [15] developed a simulation-driven approach which took attack strategies and attacker behavior into consideration. Al-Jarrah and Arafat [16] used the time delay neural network which embedded the temporal behavior of the attacks to maximize the recognition rate of network. Mitchell and Chen [17] proposed specification-based IDS which can adapt to different attacker types such as reckless, random, and opportunistic attackers. In this way, it could get a higher detection accuracy. Allodi and Massacci first pointed out that not all the vulnerabilities were equally exploited by the attacker [18] and then focused on the choice of attackers [19]. By validating the actual "traces" attacks left on real systems, they claimed that the real attacker would behave less powerful than we thought and would not exploit every vulnerability. The attackers would strategically choose the busy periods and some certain vulnerabilities, while the efforts of security professionals were diffused across many vulnerabilities [20, 21]. Based on this observation, Dumitraş [22] proposed a novel metrics that enabled a more accurate assessment of the risk of cyberattacks. Bozorgi et al. [23] used machine learning method with high dimensional feature vectors input to predict the vulnerability which was most likely to be exploited by the attacker. All these analyses, however, are from the perspective of defender, ignoring the information sharing mechanism among attackers where the mechanism is the most important part during the attack and can guide attackers to changer their strategies dynamically.

3. Preliminaries

Before introducing the long-term game, we begin with the simple attack-defense game. The players of the game are attacker and defender. Liang and Xiao [24] divide the game applications into two subclasses: general analysis and the specialized analysis. In general analysis, the networks are often not specific but abstract, and the strategy set of attacker SA = {attack, not attack}; meanwhile, the set of defender SD = {protect, not protect}. Let C_a represent the attack cost, C_d represent the defense cost, and G represent the profit of a successful attack. In this section, G is a fixed value. Their payoffs are as shown in Table 1. In [25], a *Remainder cost* was defined to indicate the damage that the attack brought to the system after the defender implemented the defense strategy, and $Remainder\ \ cost = G \times \varepsilon$ where $\varepsilon \in [0, 1]$. For simplicity, we assume that $\varepsilon = 0$ in this section; that is, if the defender implemented the defense strategy, there is no damage to the system and no reward to the attacker.

According to Table 1, we can see that when the attack cost is fixed, if the attacker and the defender are completely rational, both of them will make their decisions by calculating

the Nash equilibrium, and the Nash equilibrium is related to the parameter discussed below:

(1) If $G \leq C_a$, the attacker will not attack and the defender will not protect. In this case, we can say the network is pretty safe.

(2) If $C_d \geq G > C_a$, the attacker will attack, but the defender still not protect. In this case, we can say the network is of great danger.

(3) If $G > C_a$ and $G > C_d$, no pure Nash equilibrium exists, but there is a mixed Nash equilibrium; that is the attacker will choose to launch an attack with the probability of $P_A = C_d/G$, and the protecting probability $P_D = (G - C_a)/G$. In this case, we can say the network is a little bit safer than case two, but more dangerous than case one.

It can be seen from the above model that if the defender decides to protect the target, the attacker cannot finish his attack successfully, consequently getting no reward. As time goes on, this one-shot game is repeated time after time, and there is no necessary correlation between each of them. This is a simple case for security metrics, but it does not apply to the attacker who holds the zero-day exploits of certain target, mainly for the following two reasons. First, the payoff is much different. According to the stealth of zero-day exploits [26], most of the software and the security products cannot detect the existence of it and thus difficult to resist the zero-day attacks effectively. So, the attacker can get the corresponding profit only if he/she launches an attack. Second, the strategy is much different. Compared with the one-shot game, attacker with zero-day exploits is more concerned about the persistence of the resource, he/she has to make sure if this resource will still be useful after this attack and then makes a tradeoff between risk and profit. Therefore, the key points to the entire game process are the attacker's decision and the defense capability. In next section, we will discuss the detail of the long-term game.

4. Long-Term Game Formulation

We analyze the attack-defense scenario between multiple attackers and a single target. The target could be a company or an organization, and it contains a lot of nodes. And we assume that each node has at least one cyber resource (where the node has no resource is out of our discussion). All of these vulnerable nodes are protected by the same administrator, so we assume that all the nodes share the same defense capability. We define an attack-defense game as a combination of one resource and one node. The attacker who owns more than one resource means that he will be involved in more than one attack-defense game. So the relationship is a many-to-many mapping. As mentioned above, our discussion takes place over a certain period of time since the zero-day attack is a long-term process. We assume that it takes one-time tick to complete an attack, and if the vulnerability was not discovered by the administrator this time, it will still be useful to the attacker next time. For each time tick, we compute the probability of being attacked for every node and take these results as the security metric. In this section, we first introduce the game formulation and some key parameters and then we focus on the attackers learning strategy.

4.1. Attack-Defense Game

Cyber Resource. We call a zero-day exploit or a set of zero-day exploits a cyber resource to the attacker [27]. A cyber resource could help attacker to finish a certain attack. If one or more exploits are fixed or expired, which could cause the failure of the attack, then we say the resource is expired.

The Number of Long-Term Game. In this attack-defense scenario, there are total m attackers, n cyber resource, and q nodes. During the game, q is a constant, but m and n will change from time to time. Compared with the number of attackers, we are more concerned about the number of resources, because n resource means n attack-defense games. We use $n(t)$ to represent the total number at time tick t. There are three main aspects that will influence the number. First, at the beginning of time tick t, $n_{\text{new}}(t)$ is used to denote the newly joined resource. Second, at the end of time point t, $n_{\text{exp}}(t)$ is used to denote the number of expired resources. Third, according to the defense capability, some of the resources are randomly eliminated at the end of time point t, and the number is notated as $n_{\text{dis}}(t)$. Then the total number of attack-defense games at time point $t + 1$ is

$$n(t+1) = n(t) + n_{\text{new}}(t) - n_{\text{dis}}(t) - n_{\text{exp}}(t). \quad (1)$$

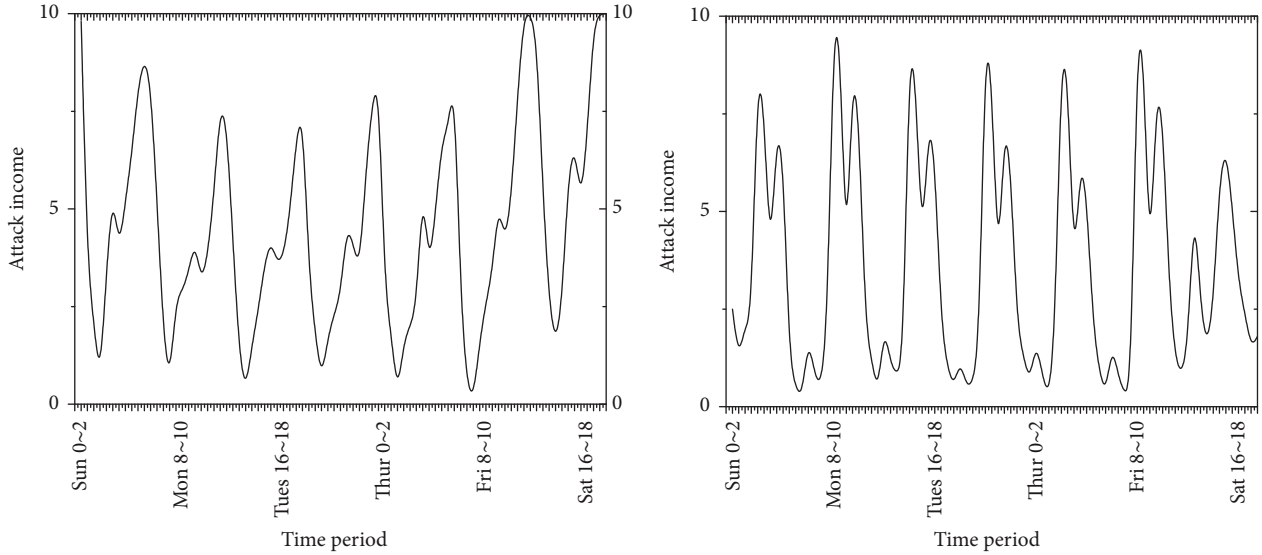
The Duration of Each Long-Term Game. Different from the one-shot game, zero-day attack-defense game is a long-term game. It consists of multiple subgames over a period of time. The game will go on as long as the resource is still useful. There are mainly three possibilities to terminate the game:

(1) The resource is expired. We use notation L to represent the lifecycle of certain resource. Different vulnerabilities have different lifecycles. For example, compared with buffer overflow and executable code, other vulnerabilities such as PHP vulnerability or SQL injection often have longer lifecycles [28]. If one of these vulnerabilities expired, we say the resource is expired.

(2) The resource was found and patched due to the attack action. We call this the passive-defense capability of the administrator, denoted as P_a, indicating the probability of discovering the vulnerability *after* being attacked.

(3) The resource was found and patched when nothing happens. We call this the initiative-defense capability of the administrator, denoted as P_b, indicating the probability of discovering the vulnerability *before* being attacked.

P_a and P_b are determined by the capability of the administrator itself, and both are fixed values which are known to the administrator and unknown to the attackers during the



(a) $g^i(t)$ of attacking a video platform

(b) $g^i(t)$ of attacking a bank

FIGURE 2: Sample of $g^i(t)$.

game process. Each attacker has its own assessment about these two values, denoted as $P_a^i(t)$ and $P_b^i(t)$. After each subgame, the attacker will revise the assessment by observing the state of other neighbors and exchanging information with each other. This is consistent with the values of most hackers who believe that all the information should be free. They think everyone has the rights to access information, so, many hacker groups form a unique "black ecosystem" to share information more effectively. According to the director of Baidu security laboratory (Baidu is the predominant search engine in China), the information sharing mechanism among such black ecology is much better than those among white ecosystem. The details about the update rules will be discussed at the next subsection.

The Strategy Set for Each Subgame. The same as the one-shot game mentioned above, the strategy set of attacker SA = {attack, not attack}; meanwhile, the set of defender SD = {defend, not defend}.

The Payoff for Each Subgame. For any resource i, at any time t, the payoff of the subgame consists of three parts: the attack cost denoted as C_a^i, the one-time attack income $g^i(t)$, and the long-term profit expectancy from time t to the end of the long-term game, denoted as $E^i(t)$. $E^i(t)$ is dependent on four factors including $g^i(t)$, $P_a^i(t)$, $P_b^i(t)$, and the attacker's action last subgame. When making a decision, what the attacker really cares about is not only the one-time attack income for the attack, but also the long-term profit expectancy during the whole lifecycle. The specific parameter settings are discussed in the next subsection. For the administrator, the loss also includes two parts, the defense cost denoted as C_d and the loss caused by the attack. In order to reduce the complexity of this model, it is assumed that the attacking loss equals the negative of the attacker's attack revenue.

Game Rules. For each subgame, given the payoffs, both attacker and administrator determine their decisions by calculating the Nash equilibrium. At the end of each subgame, if the resource is still useful, attacker will revise the assessment by observing the state of other neighbors and exchanging information with each other in order to recalculate the payoffs for next subgame.

4.2. Some Key Parameters. Notations lists the key parameters used in our long-term game model. In this paper, we assume that C_a^i, C_d, P_a, P_b, and L are fixed value, and we mainly discuss the following three parameters.

Gain Function of Time $g^i(t)$. For certain resource i, gain function of time $g^i(t)$ represents the one-time attack income that the attacker can get when he attacks the node at the certain time t. This $g^i(t)$ could be a fixed value or a function of time, and that depends on the target type. For example, if the attacker wants to attack some e-commerce platforms, he can get better rewards in some specific days such as Black Friday. For the sake of discussion, Figure 2 shows the instantaneous yield curves of attacks against a live video platform and a bank over different time periods. It can be seen that, for a bank attack, the attack revenue on weekdays is higher than that on weekends, and the attacks on working hours are higher than the off-hours. On the contrary, the attack revenue is relatively higher on weekends and late nights when the attack target is changed to the video platform. In general, we assume that $g^i(t)$ is affected by the target's visiting traffic, business process, and customer's work schedule. The specific values are beyond the scope of this article. This article assumes that both attackers and administrator have a good understanding of the target of the attack, so the value of $g^i(t)$ is known.

Profit Expectancy $E^i(t)$. For the certain resource i, we use $E^i(t)$ to represent the profit expectancy from time t to the end of the lifecycle, so $E^i(t) = \sum_{\sigma=t}^{L} P_A(\sigma) g'(\sigma)$, where $g'(\sigma) = \max\{g(\sigma) - C_a, 0\}$ and $\sigma \in [t, L]$. So, we need to compute all the $P_A(\sigma)$ for $\sigma = t, t+1, \ldots, L$. If notation Q is used to indicate that the resource is still available at time point σ and notation R is used to indicate that the attacker will choose to attack, then $P_A(\sigma) = P(Q) * P(R)$. So we have

$$P_A(\sigma) = \left[P_A(\sigma-1) * (1-P_a) + (1 - P_A(\sigma-1)) * (1-P_b)\right] * P_A(t-1). \tag{2}$$

$P_A(\sigma-1) * (1-P_a)$ means that the attacker has attacked last time but has not been discovered, and $(1 - P_A(\sigma-1)) * (1-P_b)$ means that the attacker has not attacked last time and has not been discovered. To simplify the computation, it is assumed that the defender will always be protected when calculating $P_A(\sigma)$. And $P_A(t-1)$ used in (4) is an approximate value. Because the exact value of $P_A(t)$ which is determined by calculating the Nash equilibrium of the subgame at time point σ could not be known, the probability $P_A(t-1)$ is used instead. According to the above recursive formula, we can get

$$P_A(\sigma) = \left[P_A(t-1) + \frac{(1-P_b) * P_A(t-1)}{(P_b - P_a) * P_A(t-1) - 1}\right] \cdot \left[(P_b - P_a) * P_A(t-1)\right]^{t-1} - \frac{(1-P_b) * P_A(t-1)}{(P_b - P_a) * P_A(t-1) - 1}. \tag{3}$$

The Estimation of Administrators Defense Capability $P_a^i(t)$ *and* $P_b^i(t)$. During the game, the attacker does not know the real P_a and P_b, but each attacker has its own assessment about these two values, denoted as $P_a^i(t)$ and $P_b^i(t)$. After each subgame, the attacker will update the assessment by observing the state of other neighbors and exchanging information with each other. The update rule includes the following main steps.

(i) Initializing the assessment randomly at the beginning of the game: the initial assessments are random because the attacker knows little about the defender.

(ii) Calculating the observed result of P_a and P_b: at the end of the time point t, the attacker observes his neighbors and counts the numbers of them (1) who had attacked this time and been discovered and (2) who had not attacked this time and been discovered and then calculates the observed result of P_a and P_b.

(iii) Combining the observed result, neighbors' assessment with his previous assessment to be his new assessment: when combining these three results, the reference value difference should be considered. For the neighbor who has survived longer, its assessment has a higher reference value. What is more, at the beginning of the game, the observed result plays an important role. However, as the game goes on, this importance diminished. Because the observed samples, that is, attacker's neighbors, are limited, the observed result has a strong randomness.

Some new parameters and notations are introduced as follows. Let P_{aD} denote the probability of the attacker being eliminated after attack, so $P_{aD} = P_a \times P_D$. Similarly, P_{bD} is used to denote the probability of being eliminated without attack, so $P_{bD} = P_b \times P_D$. For any attacker $i \in AT$ at the end of time point $t \in [0, L_i]$, $P_a^i(t)$ and $P_b^i(t)$ represent the attacker's assessment of P_a and P_b, $P_a^i(0)$, and $P_b^i(0)$ are the initial estimates. We use $s = \{s_1^t, s_2^t, \ldots, s_{k_t}^t\}$ to denote the neighbors' strategy and $f = \{f_1^t, f_2^t, \ldots, f_{k_t}^t\}$ to denote whether these neighbors are found by the defender or not, where k_t is the total number of neighbors of i, $s_j^t \in Sa$, $f_j^t \in 0, 1$. AD is used to denote the neighbors who had attacked this time and had been found, so $AD = \{j \mid s_j^t = 1 \wedge f_j^t = 1, j \in [0, k_t]\}$. ND is used to denote the neighbors who were not attacked and had been found, so $ND = \{j | s_j^t = 0 \wedge f_j^t = 1, j \in [0, k_t]\}$. $P_D^t(j)$ is used to denote the P_D calculated by neighbor j. Let P_{obaD}^t denote the observed value of P_{aD}, so $P_{obaD}^t(i) = |AD| / \sum s_j^t$; let P_{obbD}^t denote the observed value of P_{bD} so $P_{obbD}^t(i) = |ND|/(k - \sum s_j^t)$. After introducing above notations, this paper presents three plans for revising the parameters P_a and P_b; we will make a brief introduction.

Plan 1: Average Summation. Record all the P_a^t and P_b^t for each neighbor and resource i itself, and then take the average with $P_{obbD}^t(i)$ to calculating P_{aD}^t, donated as

$$\overline{P_{aD}^{t+1}} = \frac{\sum_{j=1}^{k_t} P_a^t(j) P_D^t(j) + P_a^t(i) P_D^t(i) + P_{obaD}^t(i)}{k_t + 2} \tag{4}$$

and then calculate

$$\overline{P_D^{t+1}} = \frac{\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)}{k_t + 1} \tag{5}$$

so

$$P_a^{t+1} = \frac{\overline{P_{aD}^{t+1}}}{\overline{P_D^{t+1}}} = \frac{\left(\sum_{j=1}^{k_t} P_a^t(j) P_D^t(j) + P_a^t(i) P_D^t(i) + |AD| / s_j^t\right)(k_t + 1)}{(k_t + 2)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)},$$
$$P_b^{t+1} = \frac{\overline{P_{bD}^{t+1}}}{\overline{P_D^{t+1}}} = \frac{\left(\sum_{j=1}^{k_t} P_b^t(j) P_D^t(j) + P_b^t(i) P_D^t(i) + |ND| / \left(k - s_j^t\right)\right)(k_t + 1)}{(k_t + 2)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}. \tag{6}$$

Plan 2: Staged Average Summation. In the first half of the resource life cycle, revise P_a and P_b according to Plan 1, and remove $P_{oabD}^t(i)$ and $P_{obbD}^t(i)$ when calculating P_{aD}^{t+1} and P_{bD}^{t+1} in the second half. That is because when this long-term game

goes to a certain stage, the estimated value $P_a^{t+1}(i)$ and $P_b^{t+1}(i)$ are already close to the actual value, but there is a large uncertainty in the observations, so we removed the $P_{oabD}^t(i)$ and $P_{obbD}^t(i)$ in the second half.

$$P_a^{t+1}(i) = \begin{cases} \dfrac{\left(\sum_{j=1}^{k_t} P_a^t(j) P_D^t(j) + P_a^t(i) P_D^t(i) + |AD| / s_j^t\right)(k_t+1)}{(k_t+2)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 \le \dfrac{L}{2}, \\ \dfrac{\sum_{j=1}^{k_t} P_a^t(j) P_D^t(j) + P_a^t(i) P_D^t(i)}{\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 > \dfrac{L}{2}, \end{cases}$$

$$P_b^{t+1}(i) = \begin{cases} \dfrac{\left(\sum_{j=1}^{k_t} P_b^t(j) P_D^t(j) + P_b^t(i) P_D^t(i) + |ND| / \left(k - s_j^t\right)\right)(k_t+1)}{(k_t+2)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 \le \dfrac{L}{2}, \\ \dfrac{\sum_{j=1}^{k_t} P_b^t(j) P_D^t(j) + P_b^t(i) P_D^t(i)}{\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 > \dfrac{L}{2}. \end{cases} \tag{7}$$

Plan 3: Staged and Weighted Average Summation. Based on plan two, we introduce the concept of neighbor weight; that is, the longer the neighbor exists, the greater reference value it can provide. It should be noted that the notation t in $P_a^t(i)$ and $P_b^t(i)$ is the existing time of resource i but not the existing time of certain neighbor. So we use $T = \{t_1, t_2, \ldots, t_k\}$ to represent the existing time of k neighbors. Then the formula of plan three is as follows:

$$P_a^{t+1}(i) = \begin{cases} \dfrac{\left(\sum_{j=1}^{k_t} P_a^t(j) P_D^t(j) t_j + P_a^t(i) P_D^t(i) t + |AD| / s_j^t\right)(k_t+1)}{\left(\sum_{j=1}^{k_t} t_j + t + 1\right)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 \le \dfrac{L}{2}, \\ \dfrac{\left(\sum_{j=1}^{k_t} P_a^t(j) P_D^t(j) t_j + P_a^t(i) P_D^t(i) t\right)(k_t+1)}{\left(\sum_{j=1}^{k_t} t_j + t\right)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 > \dfrac{L}{2}, \end{cases}$$

$$P_b^{t+1}(i) = \begin{cases} \dfrac{\left(\sum_{j=1}^{k_t} P_b^t(j) P_D^t(j) t_j + P_b^t(i) P_D^t(i) t + |AD| / s_j^t\right)(k_t+1)}{\left(\sum_{j=1}^{k_t} t_j + t + 1\right)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 \le \dfrac{L}{2}, \\ \dfrac{\left(\sum_{j=1}^{k_t} P_b^t(j) P_D^t(j) t_j + P_b^t(i) P_D^t(i) t\right)(k_t+1)}{\left(\sum_{j=1}^{k_t} t_j + t\right)\left(\sum_{j=1}^{k_t} P_D^t(j) + P_D^t(i)\right)}, & t+1 > \dfrac{L}{2}. \end{cases} \tag{8}$$

5. Measure the Network Security

5.1. Calculate the Nash Equilibrium for Each Subgame. After calculating the above parameters, we can fill the payoffs matrix for each subgame (see Table 2) where

$$\begin{aligned} G_A^D &= g'(t) + (1 - P_A) E(t+1), \\ G_{NA}^D &= (1 - P_b) E(t+1), \\ G_A^{ND} &= g'(t) + E(t+1), \\ G_{NA}^{ND} &= E(t+1). \end{aligned} \tag{9}$$

Here is a brief discussion of the game's Nash equilibrium, also the optimal timing selection guidelines:

(1) At some time t, when $g'(t) \le 0$ and $C_d < P_b E(t+1)$, attacker will not attack but the defender will protect. When $g'(t) \le 0$ and $C_d > P_b E(t+1)$, attacker will not attack and the defender will not protect.

Because $g'(t) \le 0$ and $P_a > P_b$, so $G_A^{ND} < G_{NA}^{ND}$; the best choice for attackers is not attacking whatever the defending choice is. But in terms of the defenders, if $C_d < P_b E(t+1)$, that means there is some probability of discovering the vulnerability by expending a little defending cost, so the defender will defend. But if the cost is high, that is, $C_d > P_b E(t+1)$, the defender will not defend.

(2) At some time t, when $g'(t) > (P_a - P_b)E(t+1)$ and $C_d < P_a E(t+1)$, attacker will attack and the defender will protect. When $g'(t) > (P_a - P_b)E(t+1)$ and $C_d \ge P_a E(t+1)$, attacker will attack and the defender will not protect.

Because $g'(t) > (P_a - P_b)E(t+1)$, that means the profit of this attack is higher than the loss caused by the discovery of the attack, so the best choice for attackers is attacking whatever the defending choice is. But in terms of the defender, under this circumstance, if

Table 2: Payoff in the multiplayer evolutionary game.

	Protect	Not protect
Attack	$G_A^D, -G_A^D - C_d$	$G_A^{ND}, -G_A^{ND}$
Not attack	$G_{NA}^D, -G_{NA}^D - C_d$	$G_{NA}^{ND}, -G_{NA}^{ND}$

$C_d < P_a E(t+1)$, the defender will defend, but if $C_d \geq P_a E(t+1)$, that means the defending cost is higher than the loss caused by being attacked, so defender will not protect.

(3) At some time t, when $0 < g'(t) \leq (P_a - P_b)E(t+1)$ and $C_d \geq P_a E(t+1)$, the defender will not protect and the attacker will attack.
Because $C_d < P_a E(t+1)$ and $P_a > P_b$, so $-G_A^D - C_d < -G_A^{ND}$ and $-G_{NA}^D - C_d < -G_{NA}^{ND}$; due to the high cost of protection, the defender will not defend whatever the attacking choice is. Under this circumstance, the attacker will choose attack.

(4) At some time t, when $0 < g'(t) \leq (P_a - P_b)E(t+1)$ and $C_d < P_b E(t+1)$, the defender will protect and the attacker will not attack.
Because $C_d < P_b E(t+1)$ and $P_a > P_b$, so $-G_A^D - C_d > -G_A^{ND}$ and $-G_{NA}^D - C_d > -G_{NA}^{ND}$; due to the low cost of protection, the defender will defend whatever the attacking choice is. Under this circumstance, the attacker will not choose attack.

(5) At some time t, when $0 < g'(t) \leq (P_a - P_b)E(t+1)$ and $P_b E(t+1) < C_d < P_a E(t+1)$, there is no pure Nash equilibrium, but only mixed Nash equilibrium; that is, the attacker will attack with the probability of $(C_d - P_b E(t+1))/((P_a - P_b)E(t+1))$, and the defender will defend with the probability of $g'(t)/((P_a - P_b)E(t+1))$.
We use X to denote the probability of attack for the attacker and Y to denote the probability of defending. So the expected utility function of the attacker is

$$\begin{aligned} U_A &= X\left[YG_A^D + (1-Y)G_A^{ND}\right] \\ &\quad + (1-X)\left[YG_{NA}^D + (1-Y)G_{NA}^{ND}\right], \\ U_D &= Y\left[X\left(-G_A^D - C_d\right) + (1-X)\left(-G_{NA}^D - C_d\right)\right] \\ &\quad + (1-Y)\left[X\left(-G_A^{ND}\right) + (1-X)\left(-G_{NA}^{ND}\right)\right]. \end{aligned} \tag{10}$$

Differentiate the above-mentioned function:

$$\begin{aligned} \frac{\partial U_A}{\partial X} &= \left[YG_A^D + (1-Y)G_A^{ND}\right] \\ &\quad - \left[YG_{NA}^D + (1-Y)G_{NA}^{ND}\right]. \end{aligned} \tag{11}$$

Let $\partial U_A/\partial X = 0$; we get

$$Y = \frac{G_{NA}^{ND} - G_A^{ND}}{G_A^D + G_{NA}^{ND} - G_A^{ND} - G_{NA}^D} = \frac{g'(t)}{(P_a - P_b)E(t+1)}. \tag{12}$$

Similarly,

$$\begin{aligned} \frac{\partial U_D}{\partial Y} &= \left[X\left(-G_A^D - C_d\right) + (1-X)\left(-G_{NA}^D - C_d\right)\right] \\ &\quad - \left[X\left(-G_A^{ND}\right) + (1-X)\left(-G_{NA}^{ND}\right)\right]. \end{aligned} \tag{13}$$

Let $\partial U_D/\partial Y = 0$; we get

$$\begin{aligned} X &= \frac{G_{NA}^D - G_{NA}^{ND} + C_d}{-G_A^D - G_{NA}^{ND} + G_A^{ND} + G_{NA}^D} \\ &= \frac{C_d - P_b E(t+1)}{(P_a - P_b)E(t+1)}. \end{aligned} \tag{14}$$

5.2. Measure the Network Security Using the Nash Equilibrium. After calculating the Nash equilibrium for each subgame, we use this result as a reference for the safety measurement of unknown threat.

At any time t, for the jth node in the target, let nonnegative vector $p_j(t) = [p_{j,1}(t), \ldots, p_{j,K_j}(t)]^T$ denote the probability distribution of K_j unknown vulnerabilities, where $p_{j,m}(t) \in [0,1]$, $m = 1, \ldots, K_j$, and $p_{j,m}(t)$ represents the probability of mth resource being used by the attacker. So we can compute the probability that the node j is being attacked at time t:

$$P_j(t) = 1 - \prod_{n=1}^{K_j}\left(1 - p_{j,n}(t)\right). \tag{15}$$

Assume $P(t) = [P_1(t), P_2(t), \ldots, P_q(t)]^T$ which models the attack probability distribution of all nodes in the target; $W(t) = [w_1(t), w_2(t), \ldots, w_q(t)]$ represent the weight of each node in the target. We use $S(t)$ to denote the final result of the security measurement:

$$S(t) = P_j(t) \cdot W(t) = \sum_{n=1}^{q} P_n(t) \cdot w_n(t). \tag{16}$$

From the above equation we can see that the higher the value of $S(t)$, the greater the threat level. In next section, we will show how to calculate this score in detail.

6. Experiment

We evaluate the effectiveness of the proposed evolutionary games by synthetic data set. All experiments are conducted on a Windows 7 system with Intel Core i7-6700 3.4 GHz CPUs and 8 G memory.

Exp1: Numbers of Subgames. We first conduct our experiment and observe the total number of subgames. We provide ten types of gain functions with vary monotonicity and codomain. We set the lifecycle of each resource as 20 (for each subgame, it equals 1), and the total number of subgames for the beginning is 12000 (different attackers may keep the same resource), distributed at a $200 * 100$ matrix. The numbers of new joins can be regarded as a statistic process obeying Gaussian distribution. We pick different P_a and P_b

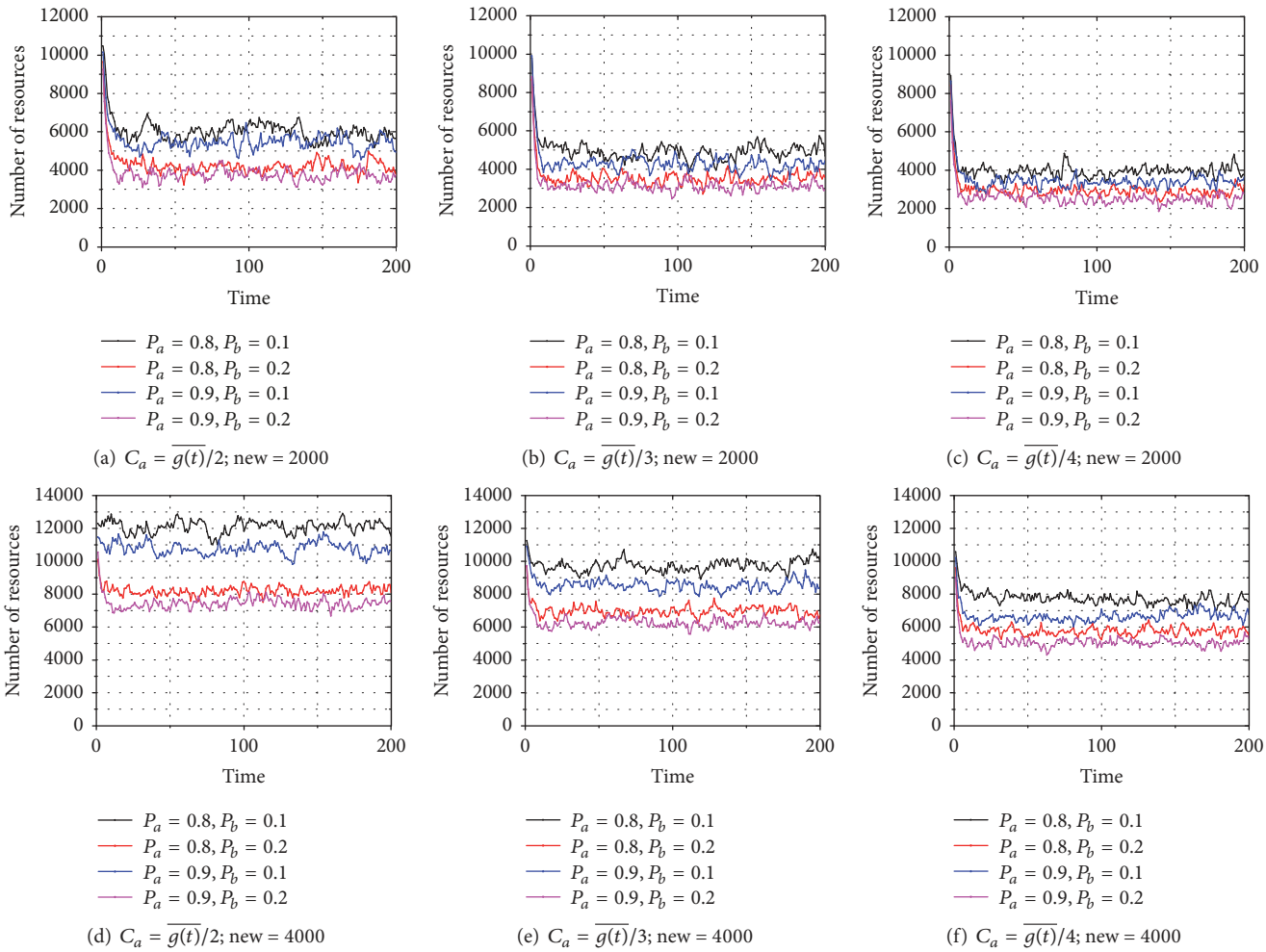


FIGURE 3: Numbers of resources with varying parameters C_a and new_{dis}.

to see whether the number could get stable with different number of new joins. Figure 3 shows the total number of subgames where the average of new joins equals 2000 and 4000; meanwhile, we set different attack cost.

We can see that the number of subgames can get stable in all these six experiments. We also can see there are so many factors that affect the number of subgames during the game. For example, different attack cost values lead to different experimental results. The average number in Figure 3(a) is higher than that in Figures 3(b) and 3(c). This is because when the cost of the attack is too high, the attacker's probability of attack will drop drastically, so the probability of being discovered is relatively lower. We can also find that when P_b equals 0.1 (notated as the black line and blue line), the number is greater than those where P_b equals 0.2 (the red line and pink line). This is because the greater P_b is, the greater the number of discovered vulnerabilities is. However, compared with P_b, the difference of P_a does not have a major impact on the change of total number. That is reasonable because during the whole game process, the probability of attack is much less than the probability of waiting, so the impact of P_a is much less.

Exp2: Measure the Network Security Using the Attack Probabilities. After calculating the attack probability for each vulnerability at each time, we use these results as important references to measure the network security. This experiment aims to show how to measure the network security using the attack probabilities. Due to the space limitation, it is impossible to list all the nodes and the attack probabilities of the game, so we use a lite version to explain our method. We assume that the target has seven nodes and each node has at least one vulnerability. Table 3 lists the probabilities of these nodes at the certain time period. The attack probability for each vulnerability is computed through the game Nash equilibrium, and the attack probability for each node, recorded as "tot" in the table, is computed through (15). From the table we can see that, for some vulnerability, the attack probability decreased to zero and never increase, such as the vulnerability 2 in node 1, which means this vulnerability had been fixed at certain time. And for some other vulnerability, the attack probability stayed at zero for a period and then increased such as vulnerability 2 in node 2, which means the vulnerability is newly discovered at time t_6. We take node 1 as an example; there are total 2 vulnerabilities, and at

Table 3: The attack probability for 7 nodes.

(a)

Time	Node1			Node2				Node3		Node4			
	vul1	vul2	tot	vul1	vul2	vul3	tot	vul1	tot	vul1	vul2	vul3	tot
t_1	0.25	0.06	0.3	0.16	0	0.13	0.27	0.34	0.34	0.22	0.13	0.16	0.43
t_2	0.15	1	1	0.75	0	0.16	0.79	0	0	0.35	0.41	0.75	0.9
t_3	0.34	0	0.34	0.18	0	0.09	0.25	0	0	0.49	0.27	0.11	0.67
t_4	0.17	0	0.17	0.26	0	0.05	0.3	0.15	0.15	0.59	0.34	0.23	0.79
t_5	0.22	0	0.22	0.57	0	0.16	0.64	0.05	0.05	0.61	0.02	0.64	0.86
t_6	0.3	0	0.3	0.63	0.15	0	0.69	0	0	0	0.17	0.37	0.47
t_7	0.51	0	0.51	0.67	0.26	0.03	0.76	0	0	0	0.06	0.19	0.23
t_8	0.06	0	0.06	0.18	0.21	0.15	0.45	0.16	0.16	0	0.19	0.28	0.41
t_9	0	0	0	0.38	0.44	0.12	0.69	0.22	0.22	0	0.25	0.44	0.58
t_{10}	0.14	0	0.14	0.47	0.37	0.17	0.72	0.13	0.13	0	0.08	0.63	0.66

(b)

Time	Node5					Node6				Node7			
	vul1	vul2	vul3	vul4	tot	vul1	vul2	vul3	tot	vul1	vul2	vul3	tot
t_1	0.44	0.34	0.11	0.08	0.7	0.66	0.34	0.85	0.97	0.74	0.34	0.14	0.85
t_2	0.23	0.19	0.25	0.16	0.61	0.34	0.64	0.12	0.79	0.16	0.19	0.22	0.46
t_3	0.12	1	0.65	0.2	1	0.62	0.35	0.34	0.84	0.26	0.08	0.3	0.52
t_4	0.29	0.05	0.21	0.22	0.58	0.15	0.84	0.06	0.87	0.69	0.09	0.34	0.81
t_5	0.56	0.12	0.76	0.38	0.94	0.08	0.71	0.18	0.78	0.14	0.15	0.42	0.58
t_6	0.18	0.2	0.43	0.61	0.85	0	0.34	0.31	0.54	0.71	0.34	0.44	0.89
t_7	0.34	0.14	0.23	0.42	0.75	0.15	0	0.44	0.52	0.29	0.37	0.38	0.72
t_8	0.12	0.08	0.05	0.4	0.54	0.35	0	0.6	0.74	0.03	0.24	0.39	0.55
t_9	0.74	0.66	0.01	0.06	0.92	0.28	0	0.18	0.41	0.47	0.12	0.27	0.66
t_{10}	0	0	0	0	0	0.91	0	0.2	0.93	0.18	0.09	0.16	0.37

Table 4: The final scores.

Time	Node1		Node2		Node3		Node4		Node5		Node6		Node7		score
	p	w	p	w	p	w	p	w	p	w	p	w	p	w	s
t_1	0.3	0.2	0.27	0.1	0.34	0.1	0.43	0.2	0.7	0.1	0.97	0.1	0.85	0.2	0.544
t_2	1	0.2	0.79	0.1	0	0.1	0.9	0.2	0.61	0.1	0.79	0.1	0.46	0.2	0.691
t_3	0.34	0.2	0.25	0.1	0	0.1	0.67	0.2	1	0.1	0.84	0.1	0.52	0.2	0.515
t_4	0.17	0.2	0.3	0.1	0.15	0.1	0.79	0.2	0.58	0.1	0.87	0.1	0.81	0.2	0.544
t_5	0.22	0.2	0.64	0.1	0.06	0.1	0.86	0.2	0.94	0.1	0.78	0.1	0.58	0.2	0.574
t_6	0.3	0.2	0.69	0.1	0	0.1	0.47	0.2	0.85	0.1	0.54	0.1	0.89	0.2	0.54
t_7	0.51	0.2	0.76	0.1	0	0.1	0.23	0.2	0.75	0.1	0.52	0.1	0.72	0.2	0.495
t_8	0.06	0.2	0.45	0.1	0.16	0.1	0.41	0.2	0.54	0.1	0.74	0.1	0.55	0.2	0.393
t_9	0	0.2	0.69	0.1	0.22	0.1	0.58	0.2	0.92	0.1	0.41	0.1	0.66	0.2	0.472
t_{10}	0.14	0.2	0.72	0.1	0.13	0.1	0.66	0.2	0	0.1	0.93	0.1	0.37	0.2	0.412

time t_1, we compute each probability of vulnerability being exploited, that is, 0.25 and 0.06. After that we compute the probability of node 1 being attacked, that is, 0.3. Similarly, we can get the probability of being attacked for each node at time t_1. Finally, we measure the network security through (16) and the final score of the system at time t_1 to t_{10} is shown in Table 4.

7. Conclusion

This paper focuses on measuring the network security with unknown threats. Although there are a lot of research studies on network security metrics, most of them are not ideal when faced with unknown threats. To help administrator have a better understanding about the potential zero-day attack, we analyzed the behavior of attackers and predict their attack timing. Due to the stealth and persistence feature, we modeled the zero-day attack as a long-term game. We specified and computed all the key parameters during the game and then got the Nash equilibrium for each subgame. We use these results as important references to measure the network security. The experiment showed the efficiency of our approach.

Main Parameters and Descriptions

C_a^i: Attacking cost
C_d: Defending cost
P_a: Real passive-defending capability of the administrator
P_b: Real initiative-defending capability of the administrator
$P_a^i(t)$: The estimation of administrator's passive-defense capability at time t
$P_b^i(t)$: The estimation of administrator's initiative-defense capability at time t
L: Resource lifecycle
$g_n^i(t)$: The gain function of time for certain attacker n
$E_n^i(t)$: Profit expectation from time t to the end of lifecycle.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key R&D Program of China (no. 2016YFB0800702) and DongGuan Innovative Research Team Program (no. 201636000100038).

References

[1] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel, "K-zero day safety: a network security metric for measuring the risk of unknown vulnerabilities," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 1, pp. 30–44, 2014.

[2] Y. Sun, L. Yin, Y. Guo, F. Li, and B. Fang, "Optimally selecting the timing of zero-day attack via spatial evolutionary game," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pp. 313–327, Springer, 2017.

[3] M. Pendleton, R. Garcia-Lebron, J.-H. Cho, and S. Xu, "A survey on systems security metrics," *ACM Computing Surveys*, vol. 49, no. 4, article no. 62, 2016.

[4] S. Hardy, M. Crete-Nishihata, K. Kleemola et al., "Targeted threat index: Characterizing and quantifying politically-motivated targeted malware," in *Proceedings of the in USENIX Security Symposium*, pp. 527–541, 2014.

[5] U. Thakore, *A quantitative methodology for evaluating and deploying security monitors [Ph.D. thesis]*, 2015.

[6] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*, vol. 8688, pp. 1–21, Springer, 2014.

[7] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, "Evaluating computer intrusion detection systems: a survey of common practices," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–41, 2015.

[8] N. Boggs, S. Du, and S. J. Stolfo, "Measuring drive-by download defense in depth," in *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*, vol. 8688, pp. 172–191, Springer, 2014.

[9] X. De Carné De Carnavalet and M. Mannan, "A large-scale evaluation of high-impact password strength meters," *ACM Transactions on Information and System Security*, vol. 18, no. 1, 2015.

[10] B. Ur, S. M. Segreti, L. Bauer et al., "Measuring real-world accuracies and biases in modeling password guessability," in *Proceedings of the USENIX Security Symposium*, pp. 463–481, 2015.

[11] B. R. Avasarala, J. C. Day, D. Steiner et al., "System and method for automated machine-learning, zero-day malware detection," US Patent 9,292,688, March 2016.

[12] A. Mishra and B. B. Gupta, "Hybrid solution to detect and filter zero-day phishing attacks," in *Proceedings of the In Proceedings of the Second International Conference on Emerging Research in Computing, Information*, pp. 373–379, 2014.

[13] L. Wang, M. Zhang, S. Jajodia, A. Singhal, and M. Albanese, "Modeling network diversity for evaluating the robustness of networks against zero-day attacks," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 494–511, 2014.

[14] M. Zhang, L. Wang, S. Jajodia, A. Singhal, and M. Albanese, "Network Diversity: A Security Metric for Evaluating the Resilience of Networks Against Zero-Day Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 1071–1086, 2016.

[15] A. Ekelhart, E. Kiesling, B. Grill, C. Strauss, and C. Stummer, "Integrating attacker behavior in IT security analysis: a discrete-event simulation approach," *Information Technology and Management*, vol. 16, no. 3, pp. 221–233, 2015.

[16] O. Al-Jarrah and A. Arafat, "Network intrusion detection system using attack behavior classification," in *Proceedings of the*

5th International Conference on Information and Communication Systems, ICICS 2014, pp. 1–6, April 2014.

[17] R. Mitchell and I.-R. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 593–604, 2014.

[18] L. Allodi and F. Massacci, "Comparing vulnerability severity and exploits using case-control studies," *ACM Transactions on Information and System Security*, vol. 17, no. 1, article no. 1, 2014.

[19] L. Allodi, F. Massacci, and J. M. Williams, "The work-averse cyber attacker model," 2016.

[20] K. Nayak, D. Marino, P. Efstathopoulos, and T. Dumitraş, "Some Vulnerabilities Are Different Than Others," in *Proceedings of the International Workshop on Recent Advances in Intrusion Detection*, vol. 8688, pp. 426–446, Springer International Publishing.

[21] S. Mitra and S. Ransbotham, "Information disclosure and the diffusion of information security attacks," *Information Systems Research*, vol. 26, no. 3, pp. 565–584, 2015.

[22] T. Dumitraş, "Understanding the vulnerability lifecycle for risk assessment and defense against sophisticated cyber attacks," in *Cyber Warfare*, pp. 265–285, Springer, 2015.

[23] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond heuristics: Learning to classify vulnerabilities and predict exploits," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2010*, pp. 105–113, ACM, July 2010.

[24] X. Liang and Y. Xiao, "Game theory for network security," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 472–486, 2013.

[25] W. Jiang, B.-X. Fang, Z.-H. Tian, and H.-L. Zhang, "Evaluating network security and optimal active defense based on attack-defense game model," *Jisuanji Xuebao/Chinese Journal of Computers*, vol. 32, no. 4, pp. 817–827, 2009.

[26] L. Bilge and T. Dumitras, "Before we knew it: An empirical study of zero-day attacks in the real world," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012*, pp. 833–844, ACM, October 2012.

[27] R. Axelrod and R. Iliev, "Timing of cyber conflict," *Proceedings of the National Acadamy of Sciences of the United States of America*, vol. 111, no. 4, pp. 1298–1303, 2014.

[28] M. Shahzad, M. Z. Shafiq, and A. X. Liu, "A large scale exploratory analysis of software vulnerability life cycles," in *Proceedings of the 34th International Conference on Software Engineering, ICSE 2012*, pp. 771–781, IEEE Press, June 2012.