mathematics

MDPI

Article

Research on Optimization of Array Honeypot Defense Strategies Based on Evolutionary Game Theory

Leyi Shi [1,2,3,*,†], Xiran Wang [1,2,†] and Huiwen Hou [2]

[1] College of Oceanography and Space Informatics, China University of Petroleum, Qingdao 266580, China; z19160027@s.upc.edu.cn
[2] College of Computer Science and Technology, China University of Petroleum, Qingdao 266580, China; z20070032@s.upc.edu.cn
[3] Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
* Correspondence: shileyi@upc.edu.cn
† These authors contributed equally to this work.

Abstract: Honeypot has been regarded as an active defense technology that can deceive attackers by simulating real systems. However, honeypot is actually a static network trap with fixed disposition, which is easily identified by anti-honeypot technology. Thus, honeypot is a "passive" active defense technology. Dynamic honeypot makes up for the shortcomings of honeypot, which dynamically adjusts defense strategies with the attack of hackers. Therefore, the confrontation between defenders and attackers is a strategic game. This paper focuses on the non-cooperative evolutionary game mechanism of bounded rationality, aiming to improve the security of the array honeypot system through the evolutionarily stable strategies derived from the evolutionary game model. First, we construct a three-party evolutionary game model of array honeypot, which is composed of defenders, attackers and legitimate users. Secondly, we formally describe the strategies and revenues of players in the game, and build the three-party game payoff matrices. Then the evolutionarily stable strategy is obtained by analyzing the Replicator Dynamics of various parties. In addition, we discuss the equilibrium condition to get the influence of the number of servers N on the stability of strategy evolution. MATLAB and Gambit simulation experiment results show that deduced evolutionarily stable strategies are valid in resisting attackers.

Keywords: evolutionary game theory; array honeypot; multi-party game; bounded rationality; evolutionarily stable strategy; proactive defense; dynamic honeypot

check for updates

Citation: Shi, L.; Wang, X.; Hou, H. Research on Optimization of Array Honeypot Defense Strategies Based on Evolutionary Game Theory. *Mathematics* **2021**, *9*, 805. https://doi.org/10.3390/math9080805

Academic Editor: Vasyl Martsenyuk

Received: 5 March 2021
Accepted: 1 April 2021
Published: 8 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the popularity of the Internet, the security of network systems is threatened by various viruses and worms. Security precautions and network countermeasure technology are attracting more and more attention. Network security has become an important factor of national security [1], and security defense capabilities urgently need to be enhanced. Traditional network security defense technologies are basically passive defenses. For example, a firewall can discover its own vulnerabilities only after being attacked by attackers. Therefore, passive defense has been unable to resist various attack threats in today's network world. Fortunately, proactive defense technology has made up for the shortcomings of passive defense technology in recent years. It has gradually become a critical tool of network security protection systems.

Honeypot [2–4] is a kind of proactive defense technology. It simulates a real system and provides hackers with a flawed system as their attack target to interfere and confuse the hacker. Hence, honeypot can deceive the hacker and learn their purpose and means of attack. Moreover, it can detect unknown attacks. Thus, honeypot has been an important method of proactive defense technology. However, honeypot is merely a static network

trap, which is fixed disposition. It cannot adapt to the rapid changes of network services. Moreover, honeypot may be effective only for those reckless attackers but not the veterans. Once the attacker saw through the trap and bypassed it, the honeypot would immediately lose its value. Therefore, how to improve the dynamic characteristics and sweetness of honeypot has become a research hotspot.

Spitzner [5] proposed a dynamic configuration honeypot [6]. When the network environment changes, it can timely recognize these changes and automatically adjust the deployment of honeypot to resist external attacks. Therefore, dynamic configuration honeypot can deceive intruders through adaptive configuration changes to ensure the security of network system. However, dynamic configuration honeypot is still a static trap in essence. It can still be identified and bypassed by attackers through anti-honeypot technology [7]. Then the dynamic configuration honeypot loses its value.

Mimicry honeypot [8] is also a kind of dynamic honeypot whose content changes. It was proposed through the imitation and evolution of species in nature. Mimicry honeypots use nature's warning coloration and protective coloration schemes in network security defense, and evolve with changes in the network environment. In the mimicry honeypot defense system, intruders may access real services, honeypot services, and fake honeypot services. Among them, the honeypot service refers to the service full of vulnerabilities deployed by the defender to deceive the attacker, which belongs to the protective coloration scheme. Fake honeypot service refers to the real service deployed by the defender similar to the honeypot service to deceive the attacker, which belongs to the warning coloration paradigm. However, mimicry honeypot is only dynamically changeable in content, which is difficult to meet the needs of network confrontation.

Array honeypot [9–11] is another interesting dynamic honeypot, which was proposed by the idea of the battle array. It is a dynamic deception method composed of real systems. Array honeypot is coordinated by multiple machines, and the tasks of each host can be arbitrarily switched to form a constantly changing trap array. Not only can it provide normal services, but these normal services can also be used as honeypot services. In this way, array honeypot can effectively defend and confuse attack behaviors. Then the proactive network defense can be realized.

The three honeypots above are all dynamic honeypots. Although the forms are different, they all have a common concern, which is the research of deception mechanism. This paper focuses on the game analysis of the deception mechanism of array honeypot, and infers reasonable defense strategies to improve security defense capabilities of the system. There are three challenges:

- Bounded rationality: People are facing an uncertain world, and their computing and cognitive abilities of the environment are limited. To simulate a more realistic offensive and defensive process, we need to use the game theory of bounded rationality to reason about the offensive and defensive process of array honeypot system.
- Dynamic: The biggest feature of array honeypot is dynamic changes. Considering that traditional game theory may not be able to simulate the dynamic characteristics of array honeypot but only a static representation. Therefore, it is necessary for us to find a game theory that can truly interpret the offensive and defensive process of array honeypot.
- Multi-party game: When people use game theory to infer the process of network attack and defense, they usually only consider attackers and defenders. However, actual participants are the defender, the attacker and the legitimate user for array honeypot. Consequently, we need to find a three-party game method to deduce the offensive and defensive process of array honeypot.

Game theory studies decision-making when the behavior of subject directly interacts and the equilibrium problem of decision-making. In the traditional game process, players are in a completely rational ideal state. However, people are always in a state of bounded rationality in the real world.

Evolutionary game theory [12,13] is a kind of game mode with bounded rationality. In the process of game, it uses learning mechanism to constantly adjust the strategy of participants, so that the system reaches equilibrium. To solve the challenges above, we combine evolutionary game theory with array honeypot, and infer more effective array honeypot defense strategies to improve the security defense capabilities of the system.

In this paper, we construct a three-party evolutionary game model of array honeypot based on the evolutionary game theory of bounded rationality, which is composed of defenders, attackers and legitimate users. By analyzing the three-party [14,15] Replicator Dynamics, the evolutionarily stable strategies can be obtained. In addition, we can get the influence of the number of servers N on the evolutionarily stable strategy through discussing the equilibrium conditions. Finally, MATLAB and Gambit simulation results show that the evolutionarily stable strategies are effective in resisting attackers.

The main contributions of this paper are summarized as follows:

- We design a three-party evolutionary game model of array honeypot. Then we formally describe strategies and revenues of both players in the game, and the payoff matrix of the three-party evolutionary game is constructed.
- We obtain the Jacobian matrix from the partial derivatives of the Replicator Dynamics. The evolutionarily stable strategies can be obtained by analyzing the eigenvalue matrix, which is got from the Jacobian matrix.
- We use the simulation function of MATLAB and Gambit system to verify the effectiveness of our scheme. The experimental results prove that the deduced defense strategies will become stable after population evolution, and it can effectively resist attacks from attackers.

The rest of this paper is organized as follows: In Section 2, we analyze the related work of honeypot and game theory. Section 3 describes the relevant knowledge of game theory. The system model and evolutionary game analysis are defined in Section 4. In Section 5, we use MATLAB and Gambit simulation experiments to verify the effectiveness of the proposed strategies. Section 6 concludes this paper.

2. Related Work

In recent years, some scholars have applied game theory [16,17] to the process of offensive and defensive confrontation. They built various system game models to solve network security problems in different fields, and achieved some research results. For example, La et al. [18] considered the deceptive behaviors of offensive and defensive participants, and constructed a Bayesian game model with incomplete information to solve the offensive and defensive problems in honeynets. Based on the characteristics of offensive and defensive confrontation in the dynamic target defense environment, Liu et al. [19] proposed single-stage and multi-stage offensive and defensive game models based on incomplete information dynamic games, and presented a refined Bayesian equilibrium solution algorithm. Ge et al. [20] proposed a Stackelberg Security Game defense strategy generation method based on incomplete information, which solved the shortcomings of the lack of active defense capabilities of both parties in the attack and defense game in the research of network defense strategy.

Guan et al. [21] proposed an improved Bayesian model based on the relative historical revenues of participants. Among them, the offensive and defensive process was modeled as a repeated game, and the Bayesian model was used to describe the attacker's uncertain behavior. For the model, defenders decided whether to direct visitors to a regular server or honeypot server, thereby reducing attacks. Boumkheld et al. [22] defined a cyber deception game between the defender and the attacker, and the behavior between them was modeled as a Bayesian game with complete but imperfect information. Among them, the defender determined which of the low-interaction honeypot, high-interaction honeypot, or real system was installed in the system. And the attacker decided whether to attack the system according to the type of equipment he faced. Zhang et al. [23] proposed a multi-stage offensive and defensive signal game model from the reality of network offense

and defense. They solved a multi-stage offensive and defensive game equilibrium solution, and proposed an optimal proactive defense strategy selection algorithm.

Shandilya et al. [24] proposed a sufficiently general game model to capture various modes of interaction. The model promoted random game with incomplete information. And incomplete information was a wrong perception of the current state of the player due to wrong sensors. To simulate this imperfect information, they used the Euclidean distance between sensor outputs. To protect the SDN network from anti-honeypot attacks, Du et al. [25] proposed a pseudo-honeypot game strategy with theoretical performance guarantee. It proved the Bayesian Nash equilibrium under the pseudo-honeypot game strategy and the pseudo-honeypot game strategy can achieve the optimal balance between legitimate users and attackers. Shi et al. [26] deduced and analyzed the Bayesian Nash equilibrium strategies in the mimicry honeypot deception game based on the dynamic game theory of non-cooperative and incomplete information. The conditions for the mimicry honeypot model to apply the warning coloration and protective coloration schemes were given in the deception game, which proved that mimicry honeypot has better initiative, effectiveness and deception.

All the research above was based on the complete rationality of individuals. In fact, what people face is a complex and full of unknown world, and they are always in a state of bounded rationality. Therefore, these works were not consistent with the reality.

Evolutionary game theory is different from traditional game theory. It is based on a state of bounded rationality, and the behavior of participants is carried out under incomplete information. Evolutionary game is a dynamic evolution theory that emphasizes dynamic balance. Therefore, some scholars have begun to apply evolutionary game theory to the field of network security, and evolutionary game has become another important method to predict network offensive and defensive behavior.

Tian et al. [27] proposed a subjective APT-honeypot game model to study the offensive and defensive interactions. This model deployed honeypots to protect grid bus nodes. It proved that the attack and defense strategies had Bayesian Nash equilibrium under the condition of bounded rationality. Cheng et al. [28] took into account the bounded rationality in real society, and applied evolutionary game theory to the study of attack and defense costs. They analyzed the Replicator Dynamics and evolutionarily stable strategies of both offensive and defensive parties, and obtained the evolutionary rule of the network offensive and defensive confrontation process. Based on the non-cooperative evolutionary game theory, Zhu et al. [29] proposed an offensive and defensive evolutionary game model in the case of asymmetric information between the offensive and defensive parties, which had a learning mechanism. They combined the offensive and defensive utility function to demonstrate the existence and uniqueness of the Nash equilibrium in the offensive and defensive process of non-cooperative evolutionary game theory. Huang et al. [30] constructed an attack and defense evolutionary game model based on the non-cooperative evolutionary game theory. And it put forward a solution method of evolutionarily stable equilibrium from the bounded rational constraints of network attack and defense. On this basis, an optimal defense strategy selection algorithm was designed.

Although the works above studied game theory based on bounded rationality condition, they did not consider the game analysis of the offensive and defensive process of array honeypot.

Li et al. [31] formally described the revenues and strategies of array honeypot based on game theory of non-cooperative and incomplete information. And the effectiveness of the proposed array honeypot defense strategies was proved. However, the proposed strategies were actually deduced under completely rational and static conditions, while array honeypot was actually dynamically changing.

According to the bounded rationality and dynamic evolution characteristics of evolutionary game theory, we use evolutionary game theory to infer reasonable defense strategies of array honeypot. Then the security defense capabilities of the system can be improved.

Our research is different from the works above:

- We deduce the defense strategies of array honeypot based on evolutionary game theory of bounded rational, which can be more in line with real offensive and defensive scenarios.
- We use evolutionary game theory to deduce the defense strategies of array honeypot, then the evolution process can truly reflect the dynamic characteristics of array honeypot.
- The evolutionarily stable strategy we inferred can make all game participants reach a stable state of existence, and it will not change the existing stable state even if the sudden change of a certain population.

3. Overview of Game Theory

3.1. Game Theory

Game theory is a method to study decision-making when the behavior of the decision-making subject directly interacts and the equilibrium problem of this kind of decision-making [17]. In the process of the game, players choose a strategy to maximize their own revenues based on the opponent's information they have mastered. However, players do not know the strategies of other participants, so the information of other participants can only be inferred based on analysis and calculations. Therefore, the strategies implemented by players will be the best choices for each other. And it will not be changed by unexpected circumstances.

Considering the order in which people make decisions in the game, game theory can be divided into two categories:

- Static game: The problem of simultaneous decision-making by various players or different players in a sequence of decisions, but the latter decision-making person does not know the decision made by the former.
- Dynamic game: All problems except the conditions of static game.

In addition, game theory can be divided into the following two categories based on the players' understanding of opponent information:

- Complete information game: In this game, players know all opponents and understand opponent revenue, behavior strategy, etc. On the contrary, it is an incomplete information game.
- Perfect information game: In this game, players only know all the decisions that the opponent has made and the payoffs obtained. On the contrary, it is a game of imperfect information.

3.2. Evolutionary Game Theory

3.2.1. Key Concepts

In traditional game theory, participants were assumed to get what they want, and they can grasp all the information about the opponent. However, the relationship between people is very complicated in real life. People's actions are influenced by themselves and their opponents. There is a sentence that can be well explained, 'The intention is reasonable, but it can only be achieved in a limited way. ' Therefore, the game between people is a game based on bounded rationality under the condition of incomplete information.

Evolutionary game theory [32,33] was derived from the famous biologist Darwin's theory of biological evolution [34]. It regarded game participants as individuals in the population. And it combined game theory with the dynamic evolution process under the condition of bounded rationality and incomplete information. Evolutionary game theory [35–37] emphasizes a dynamic equilibrium.

Evolutionary game is a learning process. Participants continue to modify and improve their strategies during the evolution process. After long-term evolution, all participants can find a suitable and stable strategy in the game. When reaching an evolutionary stable state, the existing state will no longer be changed.

3.2.2. Evolutionarily Stable Strategy

When participants reach the game stability after a long period of evolution, they will obtain an evolutionarily stable strategy [38]. It will not be changed by any mutant group during the continuous evolution of the population. Even if some participants change the existing stable state, the entire system will quickly return to the original stable state.

Here is an example of albino Siberian Tigers to illustrate the evolutionarily stable strategy.

Albinism is a phenomenon caused by genetic mutations. Therefore, albino species are very rare in every population, and the albino Siberian tiger is one of them. It is equivalent to the individual of population mutation in evolutionary game. Suppose that the albino Siberian tiger accounts for τ in the overall share, and the population adopts the albino strategy as s' and the normal strategy as s. Moreover, if participants with a ratio of τ in the population choose strategy s', participants with a ratio of $1-\tau$ will choose strategy s. Hence, the payoff of the albino Siberian tiger is $u(s', \tau s' + (1-\tau)s))$. For any mutation strategy $s' \neq s$, if there is a condition $\overline{\tau_{s'}} \in (0,1)$ that make Equation (1) established, s is an evolutionarily stable strategy.

$$u(s, \tau s' + (1-\tau)s)) > u(s', \tau s' + (1-\tau)s)), \forall \tau \in (0, \overline{\tau_{s'}}) \quad (1)$$

This means that the evolutionary stability strategy needs to meet two conditions under any strategy $s' \neq s$:

- Balance, $u(s,s) \geq u(s',s)$.
- Stability, $u(s,s) = u(s',s) \Rightarrow u(s,s') > u(s',s')$.

Balance ensures that the strategy s satisfies the Nash equilibrium. If the participant changes the strategy arbitrarily, its own revenues will not be maximized. In other words, when the albino Siberian tiger invades the population, it will be gradually eliminated by nature. Only the common Siberian tiger can survive, which has evolved for tens of thousands of years. Therefore, the entire Siberian tiger population is still normal in the ecosystem. Even if the albino Siberian tiger occasionally appears, it will be eliminated during the evolution of the species, and it cannot adapt to the life of the population.

3.2.3. Replicator Dynamic

In the process of evolutionary game, there are two evolutionary mechanisms: Selection mechanism and Mutation mechanism. The selection mechanism means that participants will choose the strategy that maximizes their own revenues as their game strategy during the game. The mutation mechanism means that participants randomly choose a strategy to evolve, which is a risky game because the payoff of participants is unknown.

The Replicator Dynamic is a selection mechanism equation, which can describe the changing trend of the group behavior of bounded rational individuals, as follows Equation (2). Among them, $\frac{dx}{dt}$ is the rate of change in the number of participants who choose strategy ω. s_ω is the proportion of participants who choose strategy ω in the overall. $u(s,\omega)$ is the payoff of the participants who play the game with pure strategy s. $u(s,s) = \sum_1^n s_\omega \cdot u(s,\omega))$ is the average payoff of all the alternative strategies of the participants.

$$\frac{dx}{dt} = s_\omega \cdot [u(s,\omega) - u(s,s)] \quad (2)$$

Considering Equation (2), when $u(s,\omega) > u(s,s)$, the payoff of choosing pure strategy s is greater than the average payoff. Therefore, the population will choose strategy s to evolve. Conversely, strategy s will be gradually eliminated by the population. Then the fluctuations caused by the mutation will gradually return to the original stable state.

3.2.4. Evolutionary Stable Equilibrium

Hirshleifer proposed the concept of evolutionary equilibrium. He believed that when the equilibrium point was progressively stabling in a dynamic system, the dynamic stable equilibrium point was evolutionary equilibrium. For the evolutionary equilibrium,

Friedman proposed that the evolutionary equilibrium must be Nash equilibrium, but an evolutionarily stable strategy was not necessarily an evolutionary equilibrium. Therefore, evolutionary game equilibrium must be a pure strategy Nash equilibrium in the multi-group evolutionary game [39].

Considering the multi-group evolutionary game theory, we use Lyapunov theorem to judge whether the deduced strategy is an evolutionary game equilibrium. We can obtain the Jacobian matrix and equilibrium points by calculating the Replicator Dynamic, and then the eigenvalue matrix can be obtained. When the eigenvalue's real parts of the equilibrium point are all negative, the strategy represented by this equilibrium point is an evolutionarily stable strategy. Specifically, the Replicator Dynamics should satisfy two conditions:

$$\begin{cases} \frac{dx}{dt} = 0 \\ \frac{dx}{dt}' < 0 \end{cases} \tag{3}$$

Among them, $\frac{dx}{dt} = s_\omega \cdot [u(s,\omega) - u(s,s)]$ is the change rate of the strategy ω selected by the participants at time t.

4. System Model and Evolutionary Game Analysis

In this paper, we regard the network system as an ecological environment in an evolutionary game, and all visitors to the network are regarded as different populations in this environment. Hence, array honeypot system is a three-group ecosystem, which is composed of defenders, attackers, and legitimate users. This section discusses the equilibrium stability of the complex three-party evolutionary game in this ecosystem.

4.1. Evolutionary Game Model of Array Honeypot

We know that the evolutionary game model is established based on bound rationality, which is consistent with the offensive and defensive process of actual array honeypot. Therefore, this subsection builds an array honeypot evolutionary game model.

4.1.1. Model Assumptions

It is known that array honeypot is composed of multiple task hosts that work in coordination, as shown in Figure 1. We assume that there are N hosts in the system, and each host can provide users with M types of services, such as Mysql, IIS, etc. Since array honeypot system is composed of real systems, these M types of real services also serve as M types of honeypot services. Therefore, array honeypot can provide honeypot services and real services at the same time. Furthermore, array honeypot uses real services as honeypot services to reduce the disadvantages of honeypot services that are relatively large and easy to detect by attackers.

For individuals in each population of the array honeypot ecosystem, the goal of the defender is to attack the real server to obtain confidential information, and the goal of the user is to access the real server for making a purchase, and the goal of the defender is to resist various types of attacks to make the server run in a safe and stable environment.

On this basis, we make a few assumptions about the model:

1. It is assumed that the defender, the attacker, and the legitimate user are bounded rationally. That is, these three parties do not have 'omnipotent and omniscience', and they cannot obtain the optimal results instantly.
2. We suppose that the information among the defender, the attacker, and the legitimate user are not completely public. In other words, the three parties cannot accurately understand the status, payment functions and game strategies of other players.
3. We presume that attackers and legitimate users can only access one service at a time. Since array honeypot can provide honeypot services and real services at the same time, visitors may access the honeypot service or the real service when they access the server.

4. Assume that the success rate of hackers accessing the honeypot system is 100%, legitimate users will access the honeypot system and the real system, and defenders can predict real-time attacks.

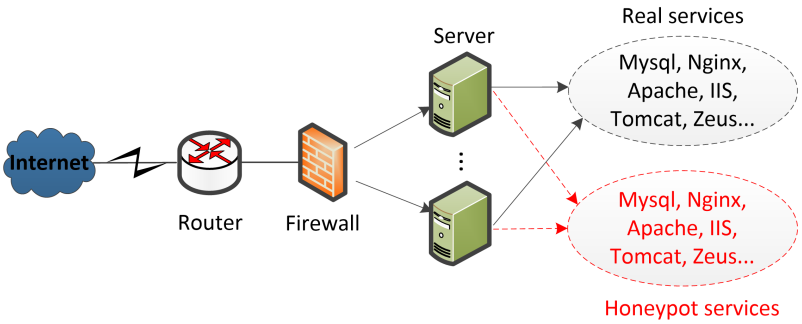

Figure 1. Array honeypots model.

4.1.2. Model Description

In this paper, the three parties of attackers, defenders, and legitimate users use 'Attacker', 'Defender', and 'User' to represent their individual collections. We assume that all three parties have two pure strategies, which are opposite to each other. That is, the executable strategies of 'Attacker', 'Defender', and 'User' are $\{S_{Attacker1}, S_{Attacker2}\}$, $\{S_{Defender1}, S_{Defender2}\}$, $\{S_{User1}, S_{User2}\}$ respectively. Among them, $\{S_{Attacker1}, S_{Attacker2}\}$ respectively indicate that the attacker chooses to attack the system or not to attack the system. And $\{S_{Defender1}, S_{Defender2}\}$ respectively represent that the server opens the service and closes the service. Then $\{S_{User1}, S_{User2}\}$ respectively indicate that the user accesses the system and does not access the system.

In particular, defenders are known to provide two types of services in array honeypot. One is the real service and the other is the honeypot service. From the essence of array honeypot, we know that these two services can be provided at the same time without interfering with each other. And visitors can only access one service at a time. Furthermore, the strategy of the defender is to turn it on or off for each type of service. Therefore, every executable strategy of the defender can be $\{S_{Defender1}, S_{Defender2}\}$. In addition, we will consider the situation of providing two kinds of services respectively in the following analysis.

On this basis, we suppose the probability of each strategy, as follows:

1. For the defender, we suppose that the array honeypot system executes the strategy $S_{Defender1}$ with probability x. Correspondingly, the strategy $S_{Defender2}$ is executed with probability $1 - x$.
2. For the attacker, it is assumed that the array honeypot system executes the strategy $S_{Attacker1}$ with probability y. And the strategy $S_{Attacker2}$ is executed with probability $1 - y$.
3. For legitimate users, we presume that the array honeypot system executes the strategy S_{User1} with probability z. In turn, the system executes the strategy S_{User2} with probability $1 - z$.

According to the assumptions above, we construct a three-party game relationship diagram between defenders, attackers and legitimate users, as shown in Figure 2. From this game relationship diagram, we can easily get the payoff distribution of the three populations in Table 1. In addition, Table 1 shows the total payoff distribution of defenders, attackers and legitimate users when the server provides two types of services, respectively. The setting of payoff parameters is shown in Table 2.

We assume that the success rate of the attacker's access to the honeypot system is 100%, and the revenue gained by the attacker from attacking the ordinary system is equal to the loss after the system is attacked. According to the service types, the revenue analysis of the three-party game can be divided into the following situations:

- Opening honeypot services. If an attacker attacks the server and a legitimate user accesses the server, the revenues are $(\frac{\eta c}{N}, -\frac{\eta c}{N} - b, -a)$ for $\{Defender, Attacker, User\}$. Since the role of deploying honeypot services is to identify illegal intrusions, attackers who access the services will suffer losses as $\frac{\eta c}{N} + b$ and legitimate users who incorrectly access the service will be damaged as a. Conversely, when the attacker does not attack the service, no matter whether the user accesses it or not, the payoff is 0 for the defender.
- Closing honeypot services. When the attacker attacks the server and the legitimate user accesses the server, the payoffs are $(0, -b, -a)$ for $\{Defender, Attacker, User\}$. It means that when they cannot access the service, they will suffer losses.
- Opening real services. If an attacker attacks the server and a legitimate user accesses the server, the payoffs are $(a - \frac{\gamma a}{N}, \frac{\gamma a}{N} - b, a)$ for $\{Defender, Attacker, User\}$. Attackers who access the real services will gain revenue as $\frac{\gamma a}{N} - b$ and legitimate users who access the real service will gain revenue as a. However, defenders will suffer losses as $a - \frac{\gamma a}{N}$. Conversely, when the attacker does not attack the service and the user accesses the service, the benefits are $(a, 0, a)$ for $\{Defender, Attacker, User\}$.
- Closing real services. Same as closing honeypot services, the attacker and the user will suffer losses, and the defender has no revenue.

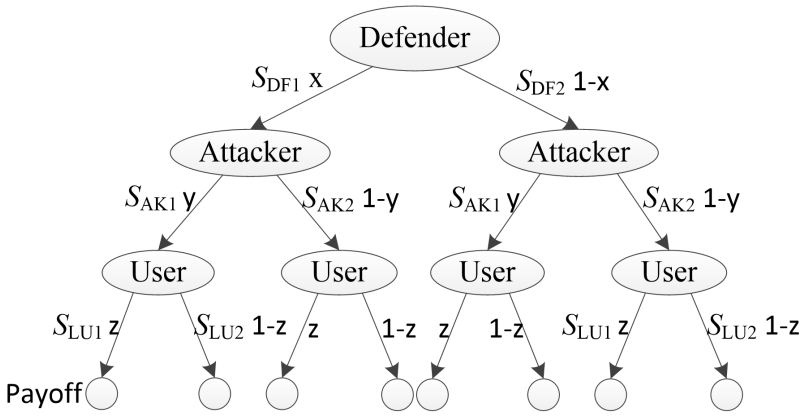

Figure 2. Three-party game relationship.

Table 1. Payoff distribution of attacker, defender, and user.

		Honeypot Services		**Real Services**	
		$S_{Defender1}$	$S_{Defender2}$	$S_{Defender1}$	$S_{Defender2}$
$S_{Attacker1}$	S_{User1}	$(-\frac{\eta c}{N} - b, -a, \frac{\eta c}{N})$	$(-b, -a, 0)$	$(\frac{\gamma a}{N} - b, a, a - \frac{\gamma a}{N})$	$(-b, -a, 0)$
	S_{User2}	$(-\frac{\eta c}{N} - b, 0, \frac{\eta c}{N})$	$(-b, 0, 0)$	$(\frac{\gamma a}{N} - b, 0, -\frac{\gamma a}{N})$	$(-b, 0, 0)$
$S_{Attacker2}$	S_{User1}	$(0, -a, 0)$	$(0, -a, 0)$	$(0, a, a)$	$(0, -a, 0)$
	S_{User2}	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$

Table 2. List of revenue parameters.

Parameters	**Conditions of Establishment**	**Characterizations of Establishment**
a	$a > 0$	Basic benefit of users or servers.
b	$a \geq b > 0$	Attack costs.
c	$c > 0$	Basic benefit of honeypots.
γ	$\gamma > 0$	The break factor of attackers.
η	$a\eta > 0$	The bait factor of honeypots.

4.2. Evolutionary Game Analysis of Array Honeypot

From the evolutionary game model , we can divide the evolutionary game analysis into two situations through the types of service, which are provided by the server.

4.2.1. Honeypot Services

According to Table 1, the payoff matrix can be obtained when the server provides the honeypot service, as shown in Table 3.

Table 3. Payoff matrix when providing honeypot services.

		$S_{Defender1}$	$S_{Defender2}$
$S_{Attacker1}$	S_{User1}	$(-\frac{\eta c}{N} - b, -a, \frac{\eta c}{N})$	$(-b, -a, 0)$
	S_{User2}	$(-\frac{\eta c}{N} - b, 0, \frac{\eta c}{N})$	$(-b, 0, 0)$
$S_{Attacker2}$	S_{User1}	$(0, -a, 0)$	$(0, -a, 0)$
	S_{User2}	$(0, 0, 0)$	$(0, 0, 0)$

Based on Table 3, we can solve the expected revenue of each strategy in the three parties respectively [40,41]. It is assumed that when the defender executes the strategies $S_{Defender1}$ and $S_{Defender2}$ respectively, the expected revenues are obtained $U_{Defender1}$ and $U_{Defender2}$, and the average expected revenue is $\overline{U_{Defender}}$. Simultaneously, the expected revenues of the attacker execute the strategies $S_{Attacker1}$ and $S_{Attacker2}$ are $U_{Attacker1}$ and $U_{Attacker2}$, and the average expected revenue is $\overline{U_{Attacker}}$. Moreover, when the user executes the strategies S_{User1} and S_{User2} respectively, the expected revenues are U_{User1} and U_{User2}, and the average expected revenue is $\overline{U_{User}}$. Therefore, we can obtain these revenues of the assumptions above as shown in Equations (4) and (5).

$$\begin{cases} U_{Defender1} = y[z \times \frac{\eta c}{N} + (1-z) \times \frac{\eta c}{N}] + (1-y)[z \times 0 + (1-z) \times 0] = y\frac{\eta c}{N} \\ U_{Defender2} = y[z \times 0 + (1-z) \times 0] + (1-y)[z \times 0 + (1-z) \times 0] = 0 \\ U_{Attacker1} = z[x \times (-\frac{\eta c}{N} - b) + (1-x) \times (-b)] + (1-z)[x \times (-\frac{\eta c}{N} - b) + (1-x) \times (-b)] = -x\frac{\eta c}{N} - b \\ U_{Attacker2} = z[x \times 0 + (1-x) \times 0] + (1-z)[x \times 0 + (1-x) \times 0] = 0 \\ U_{User1} = x[y \times (-a) + (1-y) \times (-a)] + (1-x)[y \times (-a) + (1-y) \times (-a)] = -a \\ U_{User2} = x[y \times 0 + (1-y) \times 0] + (1-x)[y \times 0 + (1-y) \times 0] = 0 \end{cases} \tag{4}$$

$$\begin{cases} \overline{U_{Defender}} = x \times U_{Defender1} + (1-x) \times U_{Defender2} = xy\frac{\eta c}{N} \\ \overline{U_{Attacker}} = y \times U_{Attacker1} + (1-y) \times U_{Attacker2} = y(-x\frac{\eta c}{N} - b) \\ \overline{U_{User}} = z \times U_{User1} + (1-z) \times U_{User2} = z(-a) \end{cases} \tag{5}$$

According to the revenues of Equation (4) and the expected revenues of Equation (5), when the three parties of defenders, attackers and legitimate users choose pure strategies $S_{Defender1}$, $S_{Attacker1}$, S_{User1} with the probability of x, y, and z respectively, the rate of change of the individual who chooses the pure strategy at time t can be obtained. These rates of change are respectively defined as $h_{Defender}(x)$, $h_{Attacker}(y)$, $h_{User}(z)$, where $h_{Defender}(x) = \frac{dx}{dt}$, $h_{Attacker}(y) = \frac{dy}{dt}$, $h_{User}(z) = \frac{dz}{dt}$. Based on the stability theory of the Replicator Dynamics system and the descriptions of Equation (2), the Replicator Dynamics [42,43] of the three-party evolutionary game can be obtained as shown in Equation (6).

$$\begin{cases} h_{Defender}(x) = \frac{dx}{dt} = x \times (U_{Defender1} - \overline{U_{Defender}}) = x \cdot (1-x) \cdot y \cdot \frac{\eta c}{N} \\ h_{Attacker}(y) = \frac{dy}{dt} = y \times (U_{Attacker1} - \overline{U_{Attacker}}) = y \cdot (1-y) \cdot y \cdot (-x \cdot \frac{\eta c}{N} - b) \\ h_{User}(z) = \frac{dz}{dt} = z \times (U_{User1} - \overline{U_{User}}) = z \cdot (1-z) \cdot (-a) \end{cases} \tag{6}$$

From Equation (6), we know that $h_{Attacker}(y) < 0$. That is, $U_{Attacker1} < \overline{U_{Attacker}}$ (the revenue of strategy $S_{Attacker1}$ is less than the average payoff of the attacker). Hence,

if evolution continues, the subgroup size of the selected strategy $S_{Attacker1}$ will gradually shrink until it is eliminated. Similarly, it can be calculated that $h_{User}(z) < 0$. In other words, $U_{User1} < \overline{U_{User}}$, which means that the revenue of strategy S_{User1} is less than the average payoff of the user. Therefore, the subgroup size of the selected strategy S_{User1} will gradually shrink until extinction.

Moreover, when $h_{Defender}(x) = 0$, we can obtain three values $x = 0$, $x = 1$ and $y = 0$. According to theoretical knowledge, the evolutionarily stable strategy is the point where the Replicator Dynamic curve intersects the horizontal axis and the slope of the tangent at the intersection is negative. Because of $y \in (0, 1)$, when $y > 0$, the slope of the tangent at the intersection $x = 0$ is positive. On the contrary, it is negative at the intersection $x = 1$. Therefore, only $x = 1$ is the evolutionarily stable strategy, which means that the defense strategy will evolve in the direction of $x = 1$ and the defense system will eventually deploy honeypots. Since y cannot be less than 0, enabling the honeypot service is the dominant strategy of the defender. Figure 3 briefly shows the trend of the function $h_{Defender}(x)$, which can reflect the evolution of the defense strategy over time when the server provides the honeypot service.

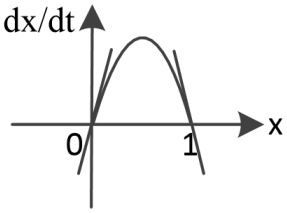

Figure 3. Sketch map of the defender's dynamic equation.

4.2.2. Real Services

When the server provides real services, the payoff matrix is shown in Table 4.

Table 4. Payoff matrix when providing real services.

		$S_{Defender1}$	$S_{Defender2}$
$S_{Attacker1}$	S_{User1}	$(\frac{\gamma a}{N} - b, a, a - \frac{\gamma a}{N})$	$(-b, -a, 0)$
	S_{User2}	$(\frac{\gamma a}{N} - b, 0, -\frac{\gamma a}{N})$	$(-b, 0, 0)$
$S_{Attacker2}$	S_{User1}	$(0, a, a)$	$(0, -a, 0)$
	S_{User2}	$(0, 0, 0)$	$(0, 0, 0)$

Same as the previous subsection, we can obtain the revenues of participants under each strategy when the server provides real services. Similarly, we assume that when the defender executes the strategies $S_{Defender1}$ and $S_{Defender2}$ respectively, the expected revenues are obtained $U_{Defender1}$ and $U_{Defender2}$, and the average expected revenue is $\overline{U_{Defender}}$. Simultaneously, the expected revenues of the attacker execute the strategies $S_{Attacker1}$ and $S_{Attacker2}$ are $U_{Attacker1}$ and $U_{Attacker2}$, and the average expected revenue is $\overline{U_{Attacker}}$. Moreover, when the user executes the strategies S_{User1} and S_{User2} respectively, the expected revenues are U_{User1} and U_{User2}, and the average expected revenue is $\overline{U_{User}}$. Then, we obtain these revenues of the assumptions for the game parties as shown in Equations (7) and (8).

$$\begin{cases} U_{Defender1} = y[z \times (a - \frac{\gamma a}{N}) + (1 - z) \times (-\frac{\gamma a}{N})] + (1 - y)[z \times a + (1 - z) \times 0] = za - y\frac{\gamma a}{N} \\ U_{Defender2} = y[z \times 0 + (1 - z) \times 0] + (1 - y)[z \times 0 + (1 - z) \times 0] = 0 \\ U_{Attacker1} = z[x \times (\frac{\gamma a}{N} - b) + (1 - x) \times (-b)] + (1 - z)[x \times (\frac{\gamma a}{N} - b) + (1 - x) \times (-b)] = x\frac{\gamma a}{N} - b \\ U_{Attacker2} = z[x \times 0 + (1 - x) \times 0] + (1 - z)[x \times 0 + (1 - x) \times 0] = 0 \\ U_{User1} = x[y \times a + (1 - y) \times a] + (1 - x)[y \times (-a) + (1 - y) \times (-a)] = 2xa - a \\ U_{User2} = x[y \times 0 + (1 - y) \times 0] + (1 - x)[y \times 0 + (1 - y) \times 0] = 0 \end{cases} \quad (7)$$

$$\begin{cases} \overline{U_{Defender}} = x \times U_{Defender1} + (1-x) \times U_{Defender2} = x \cdot (za - y\frac{\gamma a}{N}) \\ \overline{U_{Attacker}} = y \times U_{Attacker1} + (1-y) \times U_{Attacker2} = y \cdot (x\frac{\gamma a}{N} - b) \\ \overline{U_{User}} = z \times U_{User1} + (1-z) \times U_{User2} = z \cdot (2xa - a) \end{cases} \tag{8}$$

Furthermore, we define the change rate of pure strategies $S_{Defender1}$, $S_{Attacker1}$, S_{User1} of defenders, attackers and legitimate users at time t as $h_{Defender}(x)$, $h_{Attacker}(y)$, $h_{User}(z)$. Hence, the Replicator Dynamics for the three-party evolutionary game of defenders, attackers and legitimate users are shown in Equation (9).

$$\begin{cases} h_{Defender}(x) = \frac{dx}{dt} = x \times (U_{Defender1} - \overline{U_{Defender}}) = x \cdot (1-x) \cdot (za - y\frac{\gamma a}{N}) \\ h_{Attacker}(y) = \frac{dy}{dt} = y \times (U_{Attacker1} - \overline{U_{Attacker}}) = y \cdot (1-y) \cdot (x\frac{\gamma a}{N} - b) \\ h_{User}(z) = \frac{dz}{dt} = z \times (U_{User1} - \overline{U_{User}}) = z \cdot (1-z) \cdot (2xa - a) \end{cases} \tag{9}$$

Considering the idea of the Replicator Dynamics, we can judge whether the three-group game has an evolutionarily stable strategy by analyzing the asymptotic stability of the equilibrium point. From the Lyapunov theorem, we know that the asymptotic stability of the equilibrium point can be obtained by analyzing the positive and negative of the real part of the eigenvalue of Jacobian matrix. In this paper, the Jacobian matrix is called J_{DAU}, which has three eigenvalues at most. And the three eigenvalues are derived from the partial derivatives of $h_{Defender}(x)$, $h_{Attacker}(y)$ and $h_{User}(z)$ with respect to x, y and z, which are called $\lambda_k, k = 1, 2, 3$. The Jacobian matrix J_{DAU} is shown in Formula (10).

$$\begin{aligned} J_{DAU} &= \begin{bmatrix} \frac{\partial h_{Defender}(x)}{\partial x} & \frac{\partial h_{Defender}(x)}{\partial y} & \frac{\partial h_{Defender}(x)}{\partial z} \\ \frac{\partial h_{Attacker}(y)}{\partial x} & \frac{\partial h_{Attacker}(y)}{\partial y} & \frac{\partial h_{Attacker}(y)}{\partial z} \\ \frac{\partial h_{User}(z)}{\partial x} & \frac{\partial h_{User}(z)}{\partial y} & \frac{\partial h_{User}(z)}{\partial z} \end{bmatrix} \\ &= \begin{bmatrix} (1-2x) \cdot (za - y\frac{\gamma a}{N}) & x \cdot (1-x) \cdot (-\frac{\gamma a}{N}) & x \cdot (1-x) \cdot a \\ y \cdot (1-y) \cdot \frac{\gamma a}{N} & (1-2y) \cdot (x\frac{\gamma a}{N} - b) & 0 \\ z \cdot (1-z) \cdot 2a & 0 & (1-2z) \cdot (2xa - a) \end{bmatrix} \end{aligned} \tag{10}$$

We find that there are at least 8 system equilibrium points (x, y, z) in the three-party array honeypot evolutionary game by calculating the Replicator Dynamics, which are $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, $(1,1,0)$, $(0,0,1)$, $(1,0,1)$, $(0,1,1)$, $(1,1,1)$. Then, we bring these 8 points into the Jacobian matrix J_{DAU} to obtain the eigenvalue matrix J_{tz} (The three rows of elements in each column correspond in turn to a set of eigenvalues of one of the balanced nodes) of the J_{DAU}, as shown in Formula (11).

$$J_{tz} = \begin{bmatrix} (0,0,0) & (1,0,0) & (0,1,0) & (1,1,0) & (0,0,1) & (1,0,1) & (0,1,1) & (1,1,1) \\ 0 & 0 & -\frac{\gamma a}{N} & \frac{\gamma a}{N} & a & -a & a - \frac{\gamma a}{N} & -(a - \frac{\gamma a}{N}) \\ -b & \frac{\gamma a}{N} - b & b & -(\frac{\gamma a}{N} - b) & -b & \frac{\gamma a}{N} - b & b & -(\frac{\gamma a}{N} - b) \\ -a & a & -a & a & a & -a & a & -a \end{bmatrix} \tag{11}$$

From the previous understanding of eigenvalue matrix, we know that when the real part of a certain column of eigenvalues in J_{tz} is all negative, the equilibrium node corresponding to this column is an evolutionary stable equilibrium point. Otherwise, it is an unbalanced point. Moreover, the three-party evolutionary game can get evolutionarily stable strategy at the evolutionary stable node. Then the system can reach an evolutionary stable state after long-term evolution and development, and form a refined Nash equilibrium. The stability analysis of the equilibrium point is shown in Table 5.

Table 5. Balance point stability analysis.

Balance Node	Conditions for Progressive Stability			Stability Conclusion
	$\lambda_1 < 0$	$\lambda_2 < 0$	$\lambda_3 < 0$	
$(0,0,0)$	0	$-b < 0$	$-a < 0$	Unstable
$(1,0,0)$	0	$\frac{\gamma a}{N} - b$	$a > 0$	Unstable
$(0,1,0)$	$-\frac{\gamma a}{N} < 0$	$b > 0$	$-a < 0$	Unstable
$(1,1,0)$	$\frac{\gamma a}{N} > 0$	$-(\frac{\gamma a}{N} - b)$	$a > 0$	Unstable
$(0,0,1)$	$a > 0$	$-b < 0$	$a > 0$	Unstable
$(1,0,1)$	$-a < 0$	$\frac{\gamma a}{N} - b$	$-a < 0$	Indeterminate
$(0,1,1)$	$a - \frac{\gamma a}{N}$	$b > 0$	$a > 0$	Unstable
$(1,1,1)$	$-(a - \frac{\gamma a}{N})$	$-(\frac{\gamma a}{N} - b)$	$-a < 0$	Indeterminate

From Table 5, when $-(a - \frac{\gamma a}{N}) < 0$ and $-(\frac{\gamma a}{N} - b) < 0$, $(1,1,1)$ is the equilibrium point. And when $\frac{\gamma a}{N} - b < 0$, $(1,0,1)$ is the equilibrium point.

For $(1,1,1)$, it indicates that the server provides real services, the attacker attacks the server, and the user accesses the server. In this strategy, the loss of the server is less than the overall revenue of the server when the attacker attacks the server. That is, the attack probability of the attacker is small. Therefore, the server opens the real service.

$(1,0,1)$ means that the server provides real services, the attacker does not attack the server and the user accesses the server. It is known that the attack cost of attackers doubles with the increase of N. Then it brings more losses for the attacker. If the attacker wants to maintain the attack goal, he needs to increase the attack's destructive power. However, the attack difficulty will increase. However, the stable condition of this strategy is $N > 2$. Therefore, defenders can adjust the number of N to achieve system defense purposes.

In summary, since the honeypot service of the server-side is an absolute dominant strategy, the server-side strategy sets are $\{OpenHoneypotService, OpenRealService\}$ and $\{OpenHoneypotService, CloseRealService\}$. Furthermore, an attacker attacks the server in two ways. One is the attacker accessing the honeypot service, and the other is the attacker accessing the real service. Therefore, the server-side strategy set has taken into account any situation when the attacker attacks, to protect the server-side from being compromised.

5. Simulation

In this section, we use Gambit 15 and MATLAB R2020b to evaluate the effectiveness of the defense strategy obtained in Section 4.

5.1. Gambit Simulation Results

In this subsection, Gambit is used to verify the effectiveness of our inferred three-party evolutionarily stable strategies based on array honeypot. Considering that the different types of services provided by the server, we still carry out experimental analysis in two situations, which are the server providing real services and honeypot services, respectively. In addition, we know that the defense strategy will change as N increases, so the simulation experiment parameters are shown in Table 6.

Table 6. Experiment parameters.

Parameter	Values
a	100
b	80
c	80
γ	2
η	1
N	1, 2, 3, 10, 100

5.1.1. Honeypot Services

The Gambit experimental data on honeypot services are shown in Table 7. From this table, we can see that when $N = 1$, the evolutionarily stable strategy is $(1, 0, 0)$ after evolutionary game reasoning. In addition, it is obvious that even if N increases gradually, the evolutionarily stable strategy remains unchanged at $(1, 0, 0)$.

The evolutionarily stable strategy $(1, 0, 0)$ represents that when the server opens the honeypot service, neither attackers nor legitimate users will access this service. If the attacker accesses the honeypot service, there is a risk of being identified by the defender. Hence, the attacker does not access the honeypot service. Moreover, since legitimate users will not easily stray into honeypot traps, users do not access honeypot services. In this case, the stabilized Gambit experiment is shown in Figure 4.

Table 7. Experiment results when provides honeypot services.

Payoff	**N**				
	1	**2**	**3**	**10**	**100**
(a_1, b_1, c_1)	$(-160, -100, 80)$	$(-120, -100, 40)$	$(-320/3, -100, 80/3)$	$(-88, -100, 8)$	$(-80.8, -100, 0.8)$
(a_2, b_2, c_2)	$(-80, -100, 0)$	$(-80, -100, 0)$	$(-80, -100, 0)$	$(-80, -100, 0)$	$(-80, -100, 0)$
(a_3, b_3, c_3)	$(-160, 0, 80)$	$(-120, 0, 40)$	$(-320/3, 0, 80/3)$	$(-88, 0, 8)$	$(-80.8, 0, 0.8)$
(a_4, b_4, c_4)	$(-80, 0, 0)$	$(-80, 0, 0)$	$(-80, 0, 0)$	$(-80, 0, 0)$	$(-80, 0, 0)$
(a_5, b_5, c_5)	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$
(a_6, b_6, c_6)	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$
(a_7, b_7, c_7)	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
(a_8, b_8, c_8)	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
Stable Strategy (x,y,z)	$(1, 0, 0)$	$(1, 0, 0)$	$(1, 0, 0)$	$(1, 0, 0)$	$(1, 0, 0)$

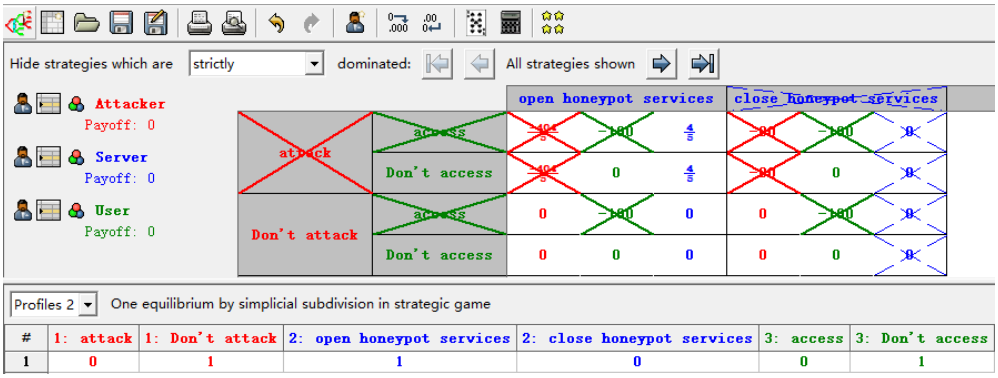

Figure 4. The stabilized Gambit experiment result of honeypot services.

5.1.2. Real Services

Table 8 shows the experimental data of Gambit when the server provides real services. We can see the table that when $N = 1$, the optimal strategy of three parties in evolutionary game is $(0, 0, 0)$. In other words, $N = 1$ is equivalent to one server providing ordinary services and not providing honeypot services. Then the server is vulnerable to be attacked. Therefore, the best strategy for defenders is to shut down the real service, and attackers and users no longer access the server.

When $N = 2$, the optimal strategy of the three-party evolutionary game is $(1, 1, 1)$. It means that the server opens the real service, the attacker attacks the service, and the user accesses the service. At this time, the attack probability of the attacker is small, and the loss of the defender is less than its revenue. However, when $N = 3$, the stability strategy is $(1, 0, 1)$ after evolutionary game reasoning, which means that the server opens the real service, the attacker does not attack the real service, and the user accesses the service. In

this strategy, the attack cost of the attacker is high, so only the user accesses the server. In addition, it is obvious that the evolutionarily stable strategy is still $(1,0,1)$ with N increases. Therefore, the results we analyzed in Section 4 are valid. The stabilized Gambit experiment is shown in Figure 5.

From the experiment above, we know that when the server starts the real service, the attacker is likely to attack the service and damage the defense side. However, attackers must pay a greater price to attack the real service as the number of N increases. When N reaches a certain value, the attacker will pay too much, and he will not attack the real service. Therefore, the defender can achieve the goal of system security defense by adjusting the number of N and the defense strategy deduced is effective in this paper.

Table 8. Experiment results when provides real services.

Payoff	N				
	1	**2**	**3**	**10**	**100**
(a_1, b_1, c_1)	$(120, 100, -100)$	$(20, -100, 0)$	$(-40/3, 100, 100/3)$	$(-60, 100, 80)$	$(-78, 100, 98)$
(a_2, b_2, c_2)	$(-80, -100, 0)$	$(-80, -100, 0)$	$(-80, -100, 0)$	$(-80, -100, 0)$	$(-80, -100, 0)$
(a_3, b_3, c_3)	$(120, 0, -200)$	$(20, 0, -100)$	$(-40/3, 0, -200/3)$	$(-60, 0, -20)$	$(-78, 0, -2)$
(a_4, b_4, c_4)	$(-80, 0, 0)$	$(-80, 0, 0)$	$(-80, 0, 0)$	$(-80, 0, 0)$	$(-80, 0, 0)$
(a_5, b_5, c_5)	$(0, 100, 100)$	$(0, 100, 100)$	$(0, 100, 100)$	$(0, 100, 100)$	$(0, 100, 100)$
(a_6, b_6, c_6)	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$	$(0, -100, 0)$
(a_7, b_7, c_7)	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
(a_8, b_8, c_8)	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
Stable Strategy (x,y,z)	$(0, 0, 0)$	$(1, 1, 1)$	$(1, 0, 1)$	$(1, 0, 1)$	$(1, 0, 1)$

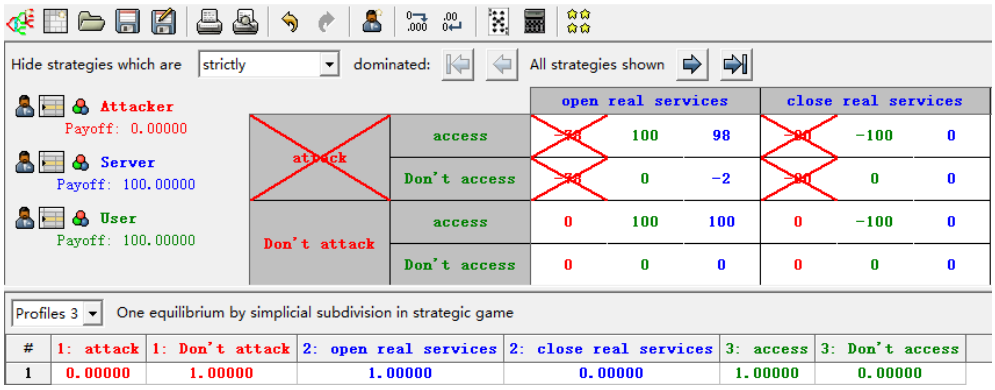

Figure 5. The stabilized Gambit experiment result of real services.

5.2. MATLAB Simulation Results

In this subsection, we use MATLAB curve to describe the evolution trend of three-party evolutionary game. Among them, the curve of each party is divided into two situations. One is the revenue when the server provides real services, and the other is the revenue when the server provides honeypot services. The revenue curves of the three parties all change with the number of servers N increases.

Figure 6 shows the defender's revenue curve, it varies greatly as N increases. For the revenue curve of the server providing real services, when $N < 2$, the minimum revenue of the defender is about -100, which means that the server will suffer a great loss. Then, the revenue gradually increases as N increases. When $N = 2$, the revenue of the server increases to a positive level. And the curve continues to grow until it stabilizes. At this time, the server revenue will stabilize at around 100. For the revenue curve of the server providing honeypot services, the revenue of the server is close to 80 at the beginning. As

N increases, the cost of deploying honeypot will gradually increase, and the payoff will decrease as the deployment cost increases.

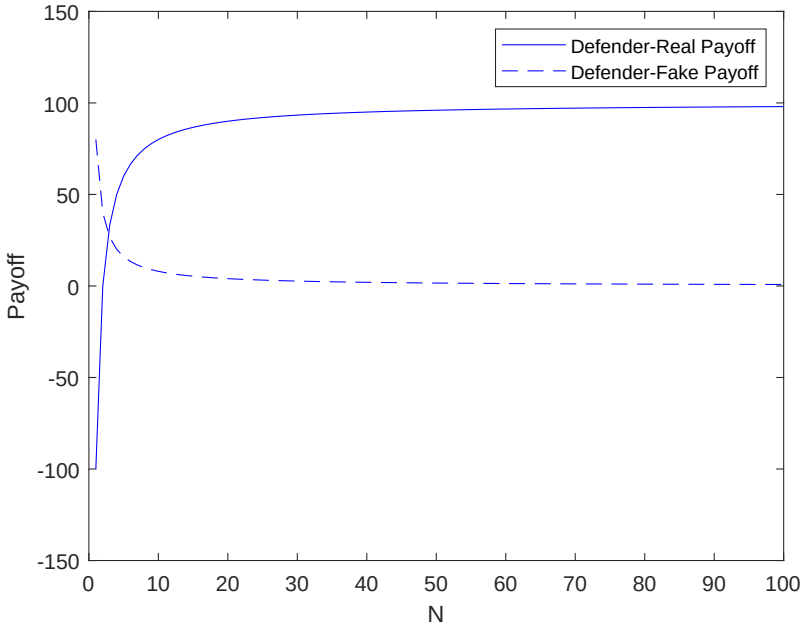

Figure 6. Payoff curves of the server.

The attacker's payoff curve is shown in Figure 7. Similarly, the revenue curve changes greatly as N increases. For the revenue curve of the attacker accesses real services, the attacker's initial revenue is about 120 , which is a great benefit. Then, the payoff decreases sharply as N increases. When $N < 2$, the payoff of the attacker decreases to a negative level. And the payoff curve continues to decrease until it tends to -80. For the revenue curve of the attacker accesses honeypot services, the attacker's initial payoff is close to -180, which means that the attacker loses a lot. However, the cost of deploying honeypot will increase with N increases. At the same time, the attacker may attack the real service, but the attack will definitely be recognized by honeypots. Therefore, although the revenues of attackers accessing honeypot services show an upward trend with the increase of N, the attacker's payoff stabilizes at around -80. That is, an attacker who attacks the honeypot service will surely suffer losses.

The legitimate user's revenue curve is shown in Figure 8. From this figure, we can see that when the legitimate user accesses real services, no matter how N changes, the revenue will always be 100. Conversely, when the legitimate user accesses honeypot services, the revenue is always -100. This means that as long as legitimate users access the honeypot service, they will certainly suffer losses. Therefore, we know that legitimate users will definitely access real services.

Figure 9 shows that the three-party revenue curves. From this figure, we can clearly see that when the server provides real services and honeypot services, the payoff trend of the three-party evolutionary game with the change of N. For the server to provide real services, when $N < 2$, the attacker obviously has an advantage over the defender. In other words, the attacker will gain a lot of revenues from attacking the real service, and the defender will suffer huge losses when providing the real service. However, the attacker's attack cost increases with N increases, and the attacker's payoff will gradually decrease. When it reaches a certain level, the attacker does not attack the real service. In this case, the defender's revenue will gradually rise. For the server to provide honeypot services, it is obvious that the revenues of the defender far exceed the payoffs of the attacker. This means that once the attacker attacks the honeypot service, he will suffer losses, while the defender will gain. Since the cost of honeypot deployment will grow with N increases,

the defense's revenues will gradually decrease. And the attacker's revenue will always be negative.

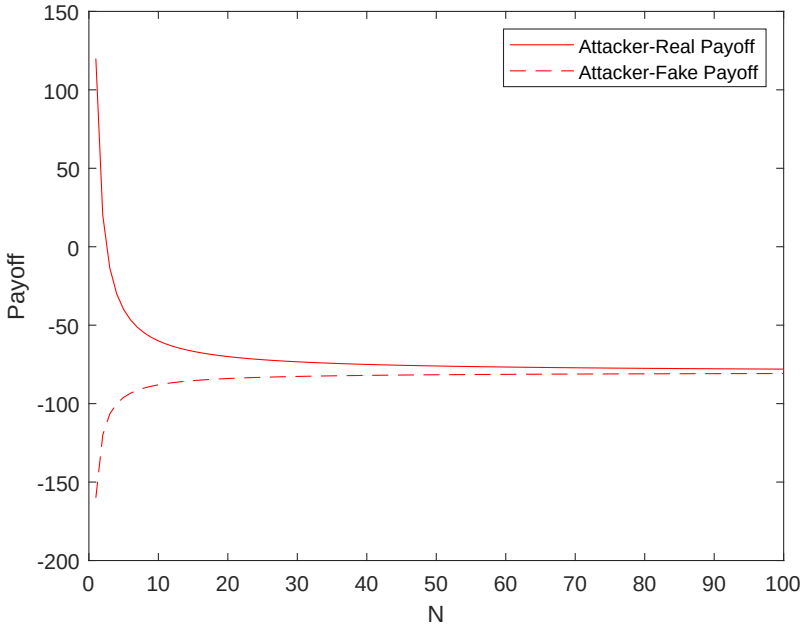

Figure 7. Payoff curves of the attacker.

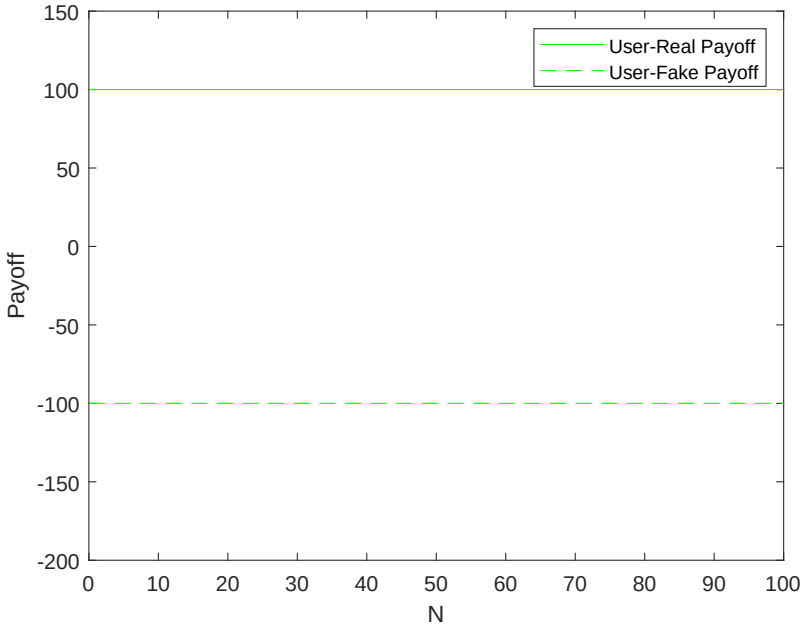

Figure 8. Payoff curves of the user.

Through the experimental data analysis above, we can prove that the evolutionarily stable strategy analyzed in Section 4 is effective. Figure 10 is the total revenue curve of the three-party evolutionary game. It can be clearly seen from this figure that as N gradually increases, the defender revenue curve shows an upward trend, and the final total revenue stabilizes at 100. The attacker's revenue curve has always been negative. It shows a downward trend, and finally stabilizes at −150. Since legitimate users will always access the real service, the payoff is always 100. Therefore, array honeypot can effectively resist attacks from attackers and ensure the safe operation of network systems.

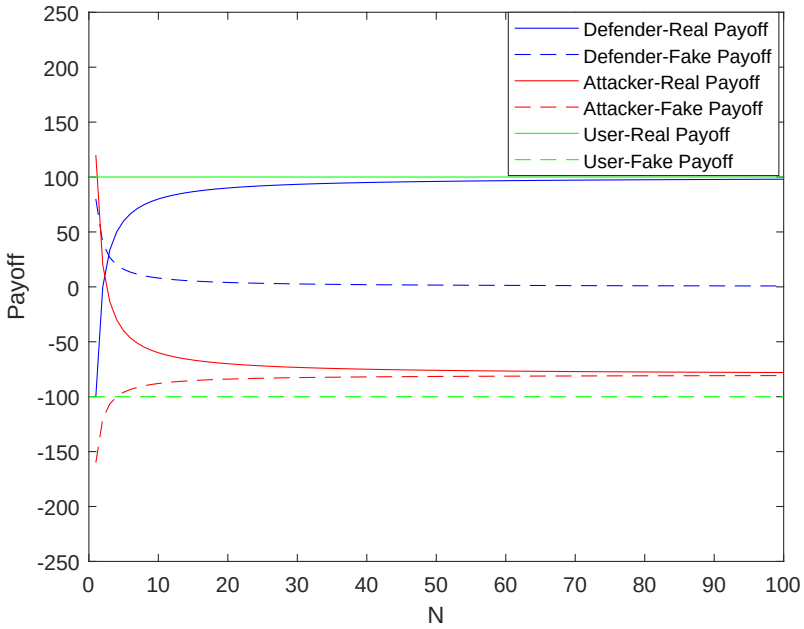

Figure 9. Payoff curves of the three-party.

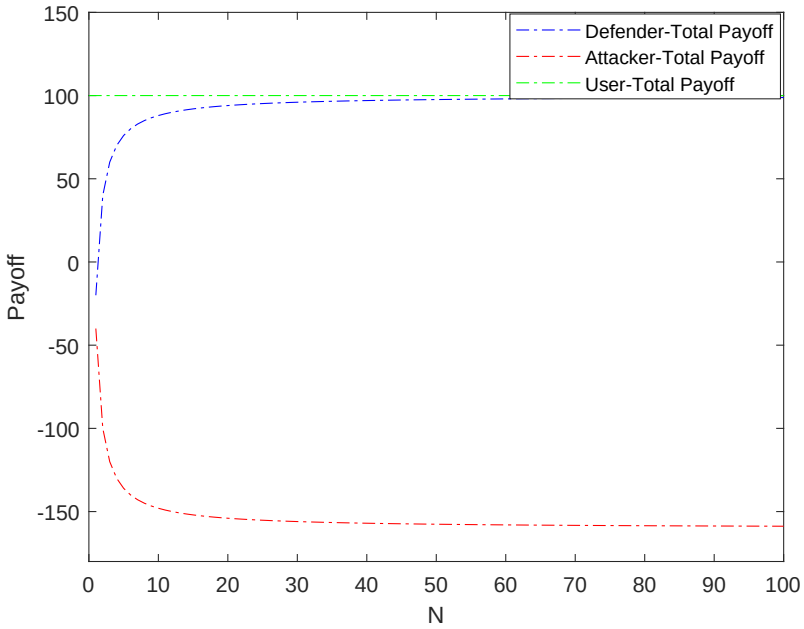

Figure 10. The three-party total revenue curves.

6. Conclusions and Outlook

With the rapid development of the Internet, network security issues have become increasingly prominent. As an active defense technology, the defense capability of honeypot is stronger than traditional passive defense. However, once honeypot is discovered by hackers, it will immediately invalid and leave the system in a passive state. Fortunately, array honeypot solves the shortcomings of honeypot technology. It uses real system to form a dynamic deception method, and it can cooperate with multiple machines and change dynamically. However, we must improve the ability of defenders to make correct defense decisions and accurate analysis in the face of attacks to enhance the security defense capabilities of the network system.

According to the evolutionary game theory, this paper studies the defense strategy of array honeypot. Because the types of services provided by servers are different, we use the same method to analyze the evolutionary game process in the two cases. First, we

construct a three-party evolutionary game model composed of defenders, attackers and legitimate users, and formally describe the payoff and strategy of game participants. Then, the payoff matrix of the three-party evolutionary game is designed. Moreover, we build the Replicator Dynamic of the three-party evolutionary game, and deduce the evolutionarily stable strategy. Finally, MATLAB and Gambit simulation experiments are used to verify the rationality of the reasoning strategies.

In this paper, we can draw the following conclusions through evolutionary game reasoning:

1. In array honeypot systems, real services and honeypot services are randomly assigned. Hence, the attacker has a high probability of accessing the honeypot service. At this time, it will bring great losses to the attacker. When attackers access a real service, the attack cost of the attacker will multiply rapidly and the revenue will decrease to a negative level with the increase of N. Therefore, the defender can adjust the number of servers N to make the evolution stable equilibrium change in favor of the defender, to resist malicious attacks and protect the network system security.
2. In the array honeypot defense system, the server will always open the honeypot service. For real services, when $N < 2$, the optimal strategy for the three-party is $(0, 0, 0)$. When $N = 2$, the optimal strategy for the three-party is $(1, 1, 1)$. When $N > 2$, the optimal strategy for the three-party is $(1, 0, 1)$.
3. When a legitimate user accesses a real service, the revenue is always 100. On the contrary, if the user accesses the honeypot service, the revenue is -100. Therefore, legitimate users will definitely access the real service and only access the real service.

This paper proposes an optimization scheme for array honeypot defense strategies based on evolutionary game theory, which can theoretically improve the defense capability of the array honeypot system. However, there are still some areas that need to be further improved:

1. In the array honeypot system, identifying hackers and legitimate users is a very important part. If the system recognizes incorrectly, irreversible losses will occur. This paper is based on the ideal situation that the system must be able to identify visitors. Therefore, we can consider using algorithms to identify system visitors in future research.
2. This paper simply states that defenders will be able to resist various types of attacks. However, we do not consider how the defender predicts the risk level of the attack. Therefore, it is necessary to establish the risk prediction mechanism to keep the risk level at an acceptable level in the future work.
3. The conclusion of our scheme is that the purpose of defending against hackers can be achieved by increasing the number of servers. However, the number of servers cannot be increased indefinitely, and the defense effect of the system will certainly be affected by the cost. Moreover, hackers may only attack the system in a flash, and the system cannot have unlimited response time. Therefore, future research needs to consider the impact of budget and response time on system defense.

Author Contributions: Conceptualization, L.S. and X.W.; methodology, L.S. and X.W.; software, L.S., X.W. and H.H.; validation, L.S., X.W. and H.H.; formal analysis, L.S. and X.W.; investigation, X.W.; resources, X.W. and H.H.; data curation, X.W.; writing—original draft preparation, L.S. and X.W.; writing—review and editing, L.S. and X.W.; visualization, L.S. and X.W.; supervision, L.S.; project administration, L.S.; funding acquisition, L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Natural Science Foundation of Shandong Province under Project ZR2019MF034, in part by the Research Funds from Guangxi Key Laboratory of Cryptography and Information Security under Grant GCIS201811, in part by the National Natural Science Foundation of China (NSFC) under grant No. 61772551.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Y.; Miao, Y.; Yang, Y.; Chen, Z.; Hu, J. The Construction and Application of Network Attack Graph. *China Commun.* **2009**, *6*, 71–74.
2. Shi, L.; Jiang, L.; Jia, C.; Wang, X. A Game Theoretic Analysis for the Honeypot Deceptive Mechanism. *J. Electron. Inf. Technol.* **2012**, *34*, 1420–1424. [CrossRef]
3. Spitzner, L. Honeypots: Catching the insider threat. In Proceedings of the 19th Annual Computer Security Conference, Las Vegas, NV, USA, 8–12 December 2003; pp. 170–179.
4. Provos, N. A virtual honeypot framework. In Proceedings of the 13th USENIX Security Symposium, San Diego, CA, USA, 9–13 August 2004; pp. 1–14.
5. Spitzner, L. *Honeypots: Tracking Hackers*; Addison-Wesley: Reading, UK, 2003.
6. Kuwatly, I.; Sraj, M.; Al Masri, Z.; Artail, H. A dynamic honeypot design for intrusion detection. In Proceedings of the IEEE/ACS International Conference on Pervasive Services, Beirut, Lebanon, 19–23 July 2004; pp. 95–104.
7. Krawetz, N. Anti-honeypot technology. *IEEE Secur. Priv.* **2004**, *2*, 76–79. [CrossRef]
8. Shi, L.; Zhao, J.; Jiang, L.; Xing, W.; Gong, J.; Liu, X. Game Theoretic Simulation on the Mimicry Honeypot. *Wuhan Univ. J. Nat. Sci.* **2016**, *21*, 69–74. [CrossRef]
9. Shi, L.; Li, J.; Han, X.; Jia, C. Design and Implementation of Distributed Self-Election Dynamic Array Honeypot System. *China Commun.* **2011**, *8*, 109–115.
10. Shi, L.; Li, J.; Liu, X.; Jia, C. Research on dynamic array honeypot for collaborative network defense strategy. *J. Commun.* **2012**, *33*, 159–164.
11. Shi, L.; Li, Y.; Liu, T.; Liu, J.; Shan, B.; Chen, H. Dynamic Distributed Honeypot Based on Blockchain. *IEEE Access* **2019**, *7*, 72234–72246. [CrossRef]
12. Herbert, G. *Game Theory Evolving*; Priceton University Press: Boston, MA, USA, 2015.
13. Edwards.; Anthony, W. The genetical theory of natural selection. *Genetics* **2000**, *154*, 1419–1426.
14. Cincotti, A. Three-player partizan games. *Theor. Comput. Sci.* **2005**, *332*, 367–389. [CrossRef]
15. Cincotti, A. N-player partizan games. *Theor. Comput. Sci.* **2010**, *411*, 3224–3234. [CrossRef]
16. Manshaei, M.; Zhu, Q.; Alpcan, T.; Basar, T.; Hubaux, J. Game Theory Meets Network Security and Privacy. *ACM Comput. Surv.* **2013**, *45*. [CrossRef]
17. Zhang, C.; Bin, N. *Game Theory and Information Economics*; Posts And Telecom Press: Beijing, China, 2015.
18. La, Q.; Quek, T.; Lee, J.; Jin, S.; Zhu, H. Deceptive Attack and Defense Game in Honeypot-Enabled Networks for the Internet of Things. *IEEE Internet Things J.* **2016**, *3*, 1025–1035. [CrossRef]
19. Liu, J.; Zhang, H.; Liu, Y. Research on Optimal Selection of Moving Target Defense Policy Based on Dynamic Game with Incomplete Information. *Acta Electron. Sin.* **2018**, *46*, 82–89.
20. Ge, X.; Zhou, T.; Zang, Y. Defense Strategy Selection Method for Stackelberg Security Game Based on Incomplete Information. In Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science, Shanghai, China, 12 November 2019.
21. Guan, R.; Li, L.; Wang, T. A Bayesian Improved Defense Model for Deceptive Attack in Honeypot-Enabled Networks. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China , 10–12 August 2019; pp. 208–214.
22. Boumkheld, N.; Panda, S.; Rass, S.; Panaousis, E. Honeypot type selection games for smart grid networks. In Proceedings of the International Conference on Decision and Game Theory for Security, Stockholm, Sweden, 30 October–1 November 2019; pp. 85–96.
23. Zhang, H.; Li, T. Optimal Active Defense Based on Multi-stage Attack-Defense Signaling Game. *Acta Electron. Sin.* **2017**, *45*, 431–439.
24. Shandilya, V. On a Generic Security Game Model. *arXiv* **2018**, arXiv:1801.05958.
25. Du, M.; Wang, K. An SDN-Enabled Pseudo-Honeypot Strategy for Distributed Denial of Service Attacks in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 648–657. [CrossRef]
26. Shi, L.; Jiang, L.; Liu, X.; Jia, C. Game Theoretic Analysis for the Feature of Mimicry Honeypot. *J. Electron. Inf. Technol.* **2013**, *35*, 1063–1068. [CrossRef]
27. Tian, W.; Ji, X.; Liu, W.; Liu, G.; Zhai, J. Prospect Theoretic Study of Honeypot Defense Against Advanced Persistent Threats in Power Grid. *IEEE Access* **2020**, *8*, 64075–64085. [CrossRef]
28. Cheng, D.; He, F.; Qi, H.; Xu, T. Modeling, Analysis and Control of Networked Evolutionary Games. *IEEE Trans. Autom. Control* **2015**, *60*, 2402–2415. [CrossRef]

29. Zhu, J.; Song, B.; Huang, Q. Evolution game model of offense-defense for network security based on system dynamics. *J. Commun.* **2014**, *35*, 54–61.
30. Huang, J.; Zhang, H.; Wang, J.; Huang, S. Defense strategies selection based on attack-defense evolutionary game model. *J. Commun.* **2017**, *38*, 168–176.
31. Li, Y.; Shi, L.; Feng, H. A Game-Theoretic Analysis for Distributed Honeypots. *Future Internet* **2019**, *11*, 65. [CrossRef]
32. Smith, J. Evolution and the theory of games. *Am. Sci.* **1976**, *64*, 41–45. [PubMed]
33. Smith, J. Game theory and the evolution of behaviour. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **1979**, *205*, 475. [CrossRef]
34. Wang, X.; Gu, C.; Zhao, J.; Quan, J. A Review of Stochastic Evolution Dynamics and Its Cooperative Mechanism. *J. Syst. Sci. Math. Sci.* **2019**, *39*, 1533–1552.
35. Huang, J.; Zhang, H.; Wang, J. Markov Evolutionary Games for Network Defense Strategy Selection. *IEEE Access* **2017**, *5*, 19505–19516. [CrossRef]
36. Huang, J.; Wang, J.; Zhang, H.; Wang, N. Network Defense Strategy Selection Based on Best-response Dynamic Evolutionary Game Model. In Proceedings of the 2nd IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 25–26 March 2017; pp. 2611–2615.
37. Zhang, H.; Huang, J. Defense Strategies Selection Method Using Non-cooperative Game. In Proceedings of the 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October 2016; pp. 1325–1330.
38. Taylor.; Peter, D.; Jonker.; Leo, B. Evolutionary stable strategies and game dynamics. *Math. Biosci.* **1978**, *40*, 145–156. [CrossRef]
39. Selten, R. A note on evolutionarily stable strategies in asymmetric animal conflicts. *J. Theor. Biol.* **1980**, *84*, 93–101. [CrossRef]
40. Deng, C. Three-Party Evolutionary Game Analysis of P2P Network Lending Based on Nonlinear System Stability Theory. *Chin. J. Manag. Sci.* **2020**, doi:10.16381/j.cnki.issn1003-207x.2019.1007. [CrossRef]
41. Zhou, T.; Zhou, S.; Liu, L. Dynamic Evolution and Stability Strategy Analysis of Game AmongGovernment, Bicycle Sharing Enterprise and Consumer. *J. Manag.* **2020**, *33*, 82–94.
42. Cheng, L.; Yang, R.; Liu, G.; Wang, J. Multi-population Asymmetric Evolutionary Game Dynamics and Its Applications in Power Demand-side Response in Smart Grid. *Proc. CSEE* **2021**. [CrossRef]
43. Cheng, L.; Liu, G.; Huang, H.; Wang, X. Equilibrium analysis of general N-population multi-strategy games for generation-side long-term bidding: An evolutionary game perspective. *J. Clean. Prod.* **2020**, *276*, 124123. [CrossRef]