Assessing the Effectiveness of Deception-Based Cyber Defense with CyberBattleSim

Quan Hong[1,2], Jiaqi Li[1,2], Xizhong Guo[1], Pan Xie[3], and Lidong Zhai[2(✉)]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
guoxizhong23@mails.ucas.ac.cn
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{hongquan,lijiaqi,zhailidong}@iie.ac.cn
[3] China United Network Communications Group Co., Ltd., Beijing, China
xiepan@chinaunicom.cn

Abstract. Deception-Based Cyber Defense technology involves deploying various elements within a network to deliberately mislead and deceive potential attackers, enabling the early detection and warning of cyber-attacks in their nascent stages. However, there is a lack of systematic research on defensive effectiveness, applicability in different scenarios, and potential synergies with other defense mechanisms of various deception technologies. To address this research gap, this study incorporates negative rewards within the CyberBattleSim platform to simulate the consequences imposed on adversaries when encountering deception techniques. We then assess the efficacy of diverse cyber deception strategies through the cumulative reward trend of attackers. Furthermore, we simulated the combined deployment of different deception technologies and the deployment of deception technology in distinct network scenarios, to evaluate the synergistic impact of deception technologies when coupled with other defensive measures and explore the suitable application scenarios of deception technology. The outcomes of multiple experiments conducted on the CyberBattleSim platform demonstrate that deception technology can impact attackers by delaying or preventing penetration and the combination of distinct deception techniques can yield varying enhancements in defense effectiveness. Additionally, the combination of Shock Trap and honeypot technology can maximize the defense effect.

Keywords: CyberBattleSim · Deception-Based Defense · Cybersecurity · Defense Effect Evaluation · Simulation

1 Introduction

The essence of cyber security lies in offensive and defensive confrontation. However, the current state of affairs indicates an imbalance in terms of time,

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 571, pp. 224–243, 2024.
https://doi.org/10.1007/978-3-031-56583-0_15

resources, information, and roles between the offensive and defensive sides. Moreover, the attack method determines the direction of cyber-defense technology development, resulting in a persistent lag for defenders. To reverse this imbalance, defenders are constantly exploring novel active defense strategies and technologies, such as moving target defense(MTD) and the honeypot. Among these technologies, we believe that deception-based defense technology holds immense potential to shift the offense-defense balance and enable proactive defense because it can not only enhance the effectiveness of defense methods but also analyze the behavior and techniques of attackers. Cyber deception technology, as denoted by previous research [12], leverages deceptive tactics within the domain of cyber security defense to interfere and mislead attackers by creating false information. This defense strategy can not only delay and consume the attacker's time but also provide early detection and warning of potential cyber threats.

Currently, deception technology has become a critical defense mechanism within the cyber security field, with widely employed technologies including the honeypot and honeynet. However, despite the existence of a variety of classic deception defense techniques, their practical application in real-world environments still faces several intricate evaluation challenges. Firstly, it is difficult to effectively quantify and evaluate the actual effectiveness of deception defense technologies in a network, making it difficult for security teams to make informed choices among divergent deception defense technologies. Secondly, the effects of distinct deception defense technologies in different network scenarios and against different attack methods have not been fully studied, which makes it difficult for defenders to maximize their defense effects when formulating network security defense strategies. Finally, the comprehensive defense effect of the combination of different deception defense technologies lacks sufficient digital evidence support, hampering the provision of clear guidance to defenders when implementing complex multi-layered network security strategies. In summary, due to the lack of comprehensive evaluation of deception defense technologies, it becomes challenging for security teams to select appropriate deception defense technologies in different scenarios. Therefore, this paper aims to explore the defensive effects of employing various deception techniques either individually or in combination across diverse scenarios. Furthermore, the acquired insights regarding defense effectiveness will be utilized to identify suitable application scenarios for the employed defense technologies, thereby providing valuable guidance to cybersecurity professionals in the design and implementation of robust defense systems. The principal contributions of this paper are as follows:

- We leverage the CyberBattleSim platform [19] for our experimentation and research, enabling us to validate and compare the defense effects of three distinct deception technologies: honeypot, decoy, and Shock Trap. We will elaborate on our reasons for choosing these three deception techniques in Sect. 2.
- We pioneer the application of the CyberBattleSim platform as a means to guide defenders by assessing the synergistic effect achieved through the combination of various deception techniques.

- Our study introduces an innovative approach by utilizing the CyberBattleSim platform to identify suitable application scenarios for the newly proposed defense technology. To exemplify this, we employ the novel deception technology called "Shock Trap" [13] as a case study to validate its application scenarios.
- This research provides a solution that enables researchers to quickly verify the defense effects of different deception strategies under a custom network architecture with less resource investment. This provides guidance for enterprises to select and deploy a combination of different deception techniques in a network environment to maximize the effectiveness of their defenses.

The rest of the paper is structured as follows. In Sect. 2, we elaborate on the process of incorporating deception techniques into the CyberBattleSim platform and the reward parameters configured for each specific deception technique. In Sect. 3, we propose three research questions that serve as the focal points of investigation in this paper, and conduct experimental evaluations for each research question, thus providing strong support for our research. Section 4 presents a brief review of related research work. In Sect. 5, we outline and discuss the limitations of our approach, while also delineating potential major research works in the future. Finally, we conclude this paper in Sect. 6.

2 Methodology

In this section, we provide a comprehensive overview of the CyberBattleSim platform [19] and its characteristics. Additionally, we describe the simulation environment we have constructed on this platform to evaluate the defense effects of various deception techniques. While the CyberBattleSim platform does not inherently support the simulation and testing of deception technology, we refer to the approach of introducing negative reward values proposed by Walter et al. [20]. This mechanism enables the attacker to acquire negative rewards upon encountering deception elements, thereby simulating the penalty of deception defense techniques on the attacker. To enhance the precision of defense effect evaluation, we configure the negative reward parameters according to the principles and properties of deception techniques. This study simulates honeypot, decoy, and Shock Trap in CyberBattleSim. The main reasons for choosing these technologies are as follows. Firstly, these technologies are all active defense technologies based on deception strategies. Evaluating the defense effect of active defense technologies can provide substantial help and guidance for enterprises when deploying and applying state-of-the-art defense technologies. Secondly, the honeypot and decoy are well-established and widely used deception defense technologies. In addition, these two technologies exhibit divergent levels of interactivity enabling us to conduct a useful comparative analysis that yields valuable insights. Thirdly, Shock Trap is an emergent deception defense technology in recent years. The simulation of Shock Trap can not only demonstrate the ability of our study to evaluate the efficacy of nascent technologies but also reflect the ability of this research to select applicable scenarios for emerging technologies. Finally, simulations of mature and emerging deception active defense techniques

underscore the comprehensiveness and generality of our study. This means that our research methodology is not only applicable to technologies that have been widely used but also to technologies that are emerging, thus providing a comprehensive evaluation and guidance for different types of defense technologies. In Sect. 2.2, we will provide a comprehensive introduction to the fundamental principles of these technologies and explain the procedure of their integration into the CyberBattleSim platform.

2.1 Environment Details

CyberBattleSim. CyberBattleSim is an open-source cyber attack and defense simulation tool released by Microsoft in April 2021. It is an experimental research platform based on reinforcement learning. It can abstract computer network and cyber security concepts at a high level, thereby abstracting the enterprise network environment into a simulation environment in which researchers can study the behavior and strategies of automated attack agents. The simulated environment consists of a network topology and a parameterized set of vulnerabilities, which can be exploited by potential attackers to move laterally in the network provided by the simulated environment. The evaluation of offensive and defensive effectiveness relies on two primary metrics: the number of simulated steps taken to accomplish a specific objective, and the cumulative reward of simulated steps across training epochs [19]. Additionally, CyberBattleSim allows researchers to design and modify simulation environments based on specific research scenarios and requirements. CyberBattleSim uses reinforcement learning algorithms to train automatic agents that are deemed attackers or defenders. These agents interact with the simulation environment to explore the optimal behavior strategy in a simulated cyber scenario. Various algorithms are employed by CyberBattleSim to train automatic agents, including Credential Lookup, Tabular Q-Learning, Deep Q-Learning (DQL), Random Search, epsilon-greedy [19]. These algorithms possess unique advantages and disadvantages, as well as distinct application scenarios, which can be selected according to network topologies and goals.

Simulation Environment. Based on CyberBattleSim, we built an abstract network simulation environment called TinyCTF. It is a modified version of the Toy Capture the Flag (ToyCTF) simulation environment provided by CyberBattleSim. The state space of TinyCTF is shown in Fig. 1. TinyCTF is an abstract representation of an enterprise's internal network topology that simulates various network assets with different security levels and attacker agents.

TinyCTF consists of 10 nodes, with each node representing either a machine or a service running on a machine. The client node serves as the initial node for the attacker to initiate lateral movement within the intranet. The edges within the graph symbolize the various attack methods that an attacker can employ. The attacker can employ three attack methods: local attack, remote attack, and authenticated connection to discover and gain control of nodes in the network. Specifically, performing a local attack can obtain credentials or other sensitive information on the controlled machine, while performing a remote attack can

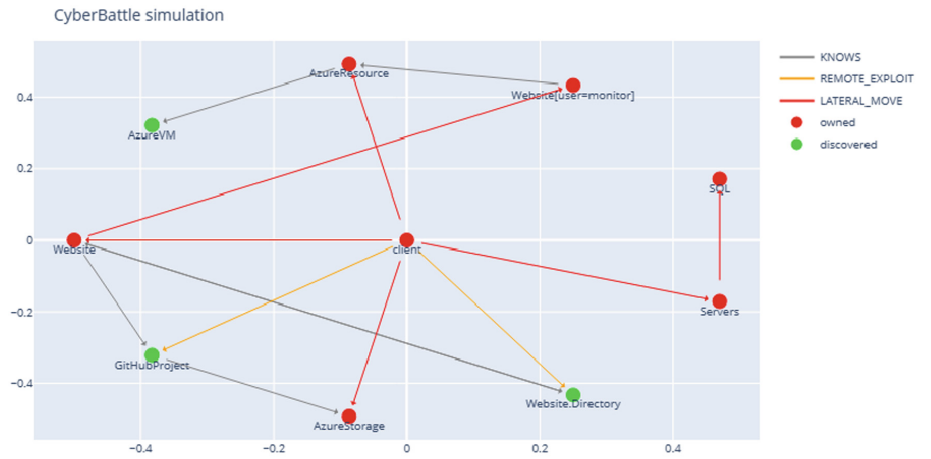


Fig. 1. The state space of TinyCTF.

discover nodes or exploit vulnerabilities on a known machine to acquire credentials. Authenticated connection means using already obtained credentials to log in and control other systems, but certain conditions must be met such as the machine being discovered, valid credentials, and enabled authentication service. The simulation environment is only partially observable for the attacker. As a result, the attacker must take action to incrementally explore the network starting from the nodes it currently possesses.

2.2 Integrating Deception Technology into CyberBattleSim

In recent years, numerous representative deception defense technologies have emerged, including Honeypot, Decoy, Honeytoken, Shock Trap, and Chaff bug [5,12,14,16]. In this section, we first introduce the basic concepts of the three deception-based active defense techniques chosen for simulation and then elaborate on their instantiation within the CyberBattleSim simulation environment. Furthermore, this section will introduce and explain the setting of the corresponding negative reward values.

Deception Defense Techniques to Evaluate. The honeypot is a cybersecurity mechanism that diverts adversaries' attention away from the real target through strategies such as luring and deception, and at the same time detects attacks early and collects attack-related data [21]. The concept of the honeypot was not initially developed for cybersecurity purposes. In 2004, Provos [17] introduced a virtual honeypot framework and demonstrated its application in the security field, which subsequently led to the widespread adoption of the honeypot in the security field. Although the basic concepts of decoy and honeypot are very similar, there exist several differences between them. Firstly, the decoy

takes the form of low-fidelity, low-interaction fake systems and is commonly used to detect early attack stages like port scanning and information gathering. The honeypot is generally deployed in the form of high-fidelity, high-interaction fake systems, which can interact more with attackers, thereby revealing more attack behaviors and methods. Secondly, the decoy is easier for administrators to create and manage than the honeypot. Recent research has demonstrated that the decoy can delay and consume an attacker's time even if they are discovered [8]. Shock Trap [13] is a deception defense technology that leverages vulnerabilities as defense resources. Different from defense technologies focused on vulnerability identification and patching, its core idea is to deploy traps based on vulnerabilities in the system and embed corresponding security mechanisms to prevent vulnerabilities from being successfully exploited. When an attacker exploits a vulnerability that builds a trap, the trap will be triggered, and then security mechanisms will detect, deny, and track the attack to deter the attacker.

Based on the principles and deception strategy of the honeypot, decoy, and Shock Trap, we configure varying negative reward values to integrate them into the simulated environment. As the attacker progresses through the exploration and control of nodes, they will inevitably encounter the deception elements established within the simulated environment. Each of the three deception techniques is implemented and characterized differently in the simulated environment.

Decoy. Within the simulated environment, the decoy is represented as virtual nodes that are generated by cloning real nodes. These decoy nodes cannot be connected or controlled by attackers. On other nodes, the attacker will find credentials that connect to the decoy nodes, but any attempt to utilize such credentials will fail. In our experimental setup, the attacker will get a reward value of -100 for the initial connection to the decoy node, and a reward value of -1 for subsequent connecting to the same decoy node.

Honeypot. The honeypot, like the decoy, is also represented as virtual nodes that are generated by cloning real nodes, but they can be connected or controlled by attackers. In our experimental configuration, the initial connection to a honeypot node by the attacker will yield a reward value of -100, and subsequent connecting to the same honeypot node shall incur a diminished penalty, amounting to a reward value of -1. Moreover, the honeypot node also contains a large number of bogus credentials, and when the attacker uses these fake credentials to connect to other nodes, an alert is also generated, which is reflected in our experiment as giving the attacker a reward value of -10.

Shock Trap. Shock Trap is strategically deployed on real nodes in the simulated environment. Since Shock Trap can deploy traps to prevent attackers from exploiting vulnerabilities, we regard the nodes where Shock Trap is deployed as nodes that cannot be controlled. In our experimental configuration, an attacker who exploits a vulnerability on a shock trap deployment node for the first time will receive a reward value of -150, and subsequent exploits of the same vulnerability will result in a reward value of -1. Moreover, as Shock Trap is a new technology, we will assess its defense effect in various scenarios and identify suit-

able application scenarios for its deployment. Finally, we will explain why we set different negative reward parameters like this in Sect. 2.3.

2.3 Reward Parameters

CyberBattleSim employs two metrics to evaluate the attacking agents: the number of simulation steps required to control the network and the cumulative reward earned by the agent during the attack. A reward value is a floating-point number that reflects the value of the attacker-controlled nodes. In our experimentation, we employed a standardized value of 1000 for all nodes that did not deploy the deception elements. However, it is acknowledged that this approach may not accurately represent real-world scenarios, where systems possess varying levels of value, such as critical infrastructure being more valuable than common systems in practical contexts. Nonetheless, this decision was influenced by several factors. Firstly, it aligns with the default configuration of the CyberBattleSim platform and the node value settings introduced by Walter et al. [20] in their integration of deception elements into the CyberBattleSim. Secondly, the objective of our experiments was to evaluate the defensive effectiveness of deception techniques by examining their impact on cumulative rewards. Therefore, unifying the values of nodes can provide a more general scenario to comprehensively evaluate the defense effect. The cumulative reward serves as an indicator of the attacker's degree of control over the network, with higher values indicating greater occupation. Moreover, it is noteworthy that CyberBattleSim itself only employs positive rewards within its framework. To assess the defense effect of the deception defense technology, we introduced various negative reward mechanisms based on the properties of deception technology, and then we compared and analyzed the impact of various deception defense technologies on cumulative rewards.

We define different negative reward values as illustrated in Table 1. The initial negative reward value assigned to an attacker upon triggering a deception technique for the first time depends on the impact the deception technique has on the attacker. For instance, when an attacker first connects to a honeypot or decoy, they will be penalized with a -100 reward value. Likewise, when an attacker first connects to a node deployed with Shock Trap, they will be penalized with a -150 reward value. The reasons why we chose to configure negative rewards in this way are as follows:

1. At present, regarding the evaluation of the defense effect of deception technology, research mainly focuses on the delay of attack time and the reduction of attack scans after the introduction of deception technologies such as honeypots/decoys. For example, studies by Aggarwal et al. [1] explored the proportion of attacks targeting conventional systems versus honeypot systems after the introduction of the honeypot, while Kocaogullar et al. [15] studied the capture of attack packets in the presence of honeypots. Balogh et al. [6] assessed the delay effect on attackers upon the introduction of honeypots. However, these studies are only limited to the evaluation of some aspects of

the defense effect of deception technology and lack a comprehensive evaluation of the impact of technologies such as honeypots and decoys on attackers. Taking the honeypot as an example, its impact on attackers encompasses three dimensions. Firstly, it delays attacks and provides timely alerts, affording defenders the time required for identifying and responding to attacks. Secondly, it can confuse attackers and redirect attack traffic toward false targets, thereby reducing attacks on real systems. Lastly, it facilitates the collection of data on attacker methods and attack characteristics, supporting the development of more advanced artificial intelligence-based defense technologies. Nonetheless, comprehensive assessments that consider the collective influence of these three aspects on attacks are lacking, and there are no standardized evaluation criteria for deception defense technologies. Consequently, we opt to customize negative reward values based on the distinct defense attributes of deception technologies. While the honeypot and decoy can divert and delay attacks, they cannot entirely prevent attackers from intruding. Their countermeasures against attackers mainly arise from defenders potentially leveraging the data they gather to profile attackers. Consequently, we set the negative reward value for the honeypot and decoy at one-tenth of the real node value, i.e., -100. Compared with technologies such as the honeypot that only detects and delays attacks, Shock Trap not only detects and denies the attack but also traces the attack, potentially exposing the identity and location of the attacker, thereby acting as a deterrent. To better compare and analyze the effect of deception defense technology, we slightly elevate the negative reward value for Shock Trap to reflect its distinct defensive characteristics.

2. Currently, there are relatively few studies evaluating the effectiveness of deception defense technologies using CyberBattleSim. We refer to the parameter settings proposed by Walter et al. [20], whose research has demonstrated that using these values can yield favorable evaluation outcomes.
3. The initial negative reward value after the deception element is triggered depends on the proactive nature of the defense. However, its value fluctuations have minimal impact on the attacker's cumulative reward trend. This is because, in our experimental scenario, only a single deception element is deployed. For automated agents simulating thousands of attacks, the cumulative reward is affected by the initial negative reward value only after the initial triggering of the deception technique. Subsequent cumulative reward trends are primarily influenced by the frequency of repeated triggers of the deception element, which, in turn, is influenced by the specific attributes of the deception element and the algorithm driving the attacker agent. We have also conducted corresponding experimental verification in Sect. 3.1.

In this study, the reward value assigned to the attacker for repetitively triggering deception elements within the simulated environment is set to -1. This determination is not only influenced by the parameter settings established by Walter et al. [20] but also rooted in several considerations. Firstly, during the process of an attacker's attempts to intrude on a network, they often connect with deceptive elements multiple times. If the negative reward value is set too

Table 1. The reward value for the agent's actions

Behavior of the attacking agent	reward value
Wrong credentials/password	−10
Repeated Mistake	−1
Trigger Shock Trap	−150
The exploit works	+50
Connect honeypot or decoy	−100
Control of node	+1000
Control of honeypot	0

high, the attacker's cumulative rewards will drop rapidly, which does not accurately reflect the practical scenario. In our experiments, establishing a reward value of −1 for the attacker's repetitive triggering of deception elements serves to provide a realistic reflection of the current attack and defense situation. Importantly, this configuration yields results that are consistent with the outcomes observed in numerous studies evaluating deception techniques. Further elaboration on this point can be found in Sect. 3. Secondly, the assessment of defense technology's effectiveness primarily focuses on the trend of the attacker's cumulative reward, and the specific size of this negative reward value mainly affects the speed rather than the fundamental direction of the trend change. The key lies in the number of triggers of this negative reward value, which is the core factor in evaluating defense effects. This study also proves this point in Sect. 3.1.

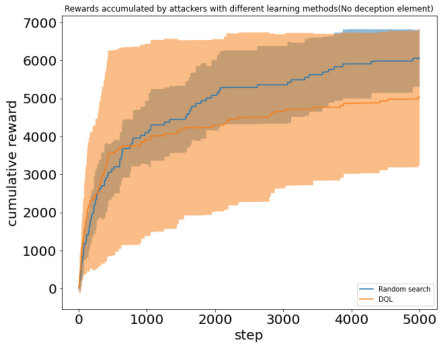
We set the value of the honeypot node to 0 because attackers will not gain any benefit from controlling the honeypot and will only consume their time. Furthermore, within the source code of CyberBattleSim, a negative reward value configuration exists explicitly designed to address situations involving incorrect password entries during the authentication process, with a default reward value set at −10. While this configuration is deactivated by default in CyberBattleSim, it is important to note that the honeypot environment contains a substantial number of fake credentials. Hence, when an attacker attempts to authenticate using these fake credentials and subsequently fails, we treat this behavior as analogous to an incorrect password entry. To account for this, we have enabled the configuration in CyberBattleSim for handling incorrect passwords, resulting in a penalty of −10 reward points for attackers, regardless of whether they employ counterfeit credentials or enter incorrect passwords.

3 Evaluation

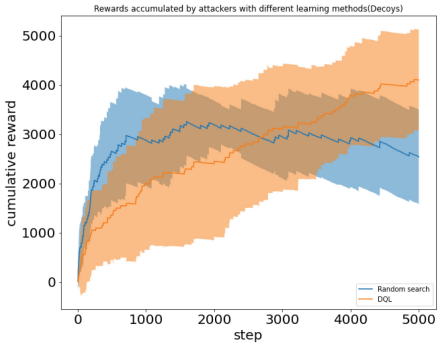
Our experiments utilize two distinct algorithms to train the attacking agent, namely Random Search and Deep Q-Learning (DQL). The Random Search algorithm is implemented by having the attacking agent randomly select an action at each step, while the Deep Q-Learning algorithm leverages the Deep Q-Network

(DQN) to learn from previous experiences to achieve the highest possible reward in the fewest steps. The experiment aims to explore and answer the following three research questions:

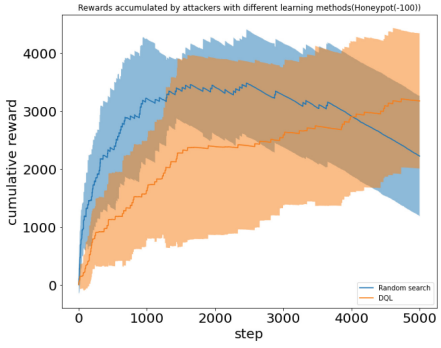
RQ1. How effective are deception defense technologies such as the Shock Trap, Honeypot, and Decoy in defending against cyber-attacks?
RQ2. Can the combination of defense technologies such as the Shock Trap, Honeypot, and Decoy, effectively enhance the security of information systems? Furthermore, which pairing of two deception techniques yields the highest degree of defense effectiveness?
RQ3. How does the defense effect of the Shock Trap vary across different scenarios? Specifically, which network topology scenarios are most suitable for its deployment? Additionally, what level of protection can the Shock Trap provide to the system within its optimal deployment scenario?



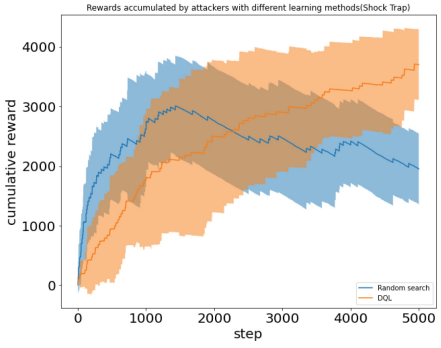
(a) Cumulative rewards plot of the agent without deception elements.



(b) Cumulative reward plot of the agent under decoy deployment.



(c) Cumulative reward plot of the agent under honeypot deployment.



(d) Cumulative reward plot of the agent under Shock Trap deployment.

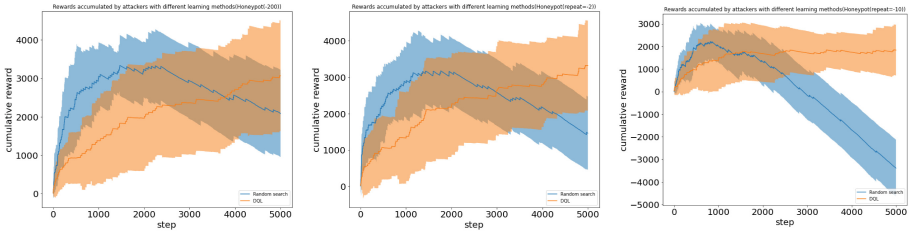
Fig. 2. The cumulative reward plot of the agent under the non-deception defense technology, decoy, honeypot, and Shock Trap.

3.1 RQ1: Evaluate Different Deception Defense Techniques

To minimize the interference of deceptive element position in the network on defense effectiveness, we deployed the deceptive elements on the same node (GitHubProject) in the TinyCTF environment. Figure 2 illustrates a comparison of the defensive effects of various strategies (no deception elements and decoy, honeypot, and Shock Trap respectively used on the same node) in the TinyCTF scenario. The colored shading in the figures shows the standard deviation from each line.

Based on the experimental results, we observed that all three deception techniques, namely the decoy, honeypot, and Shock Trap, can impede the attacker to some extent in the TinyCTF scenario. This observation aligns with the outcomes of other research endeavors that have assessed deception techniques via real-world deployments [8] and Capture The Flag (CTF) [6]. A comparison of Figs. 2b, 2c, and 2d, revealed the following conclusions. Firstly, in comparison to the decoy and Shock Trap, the honeypot exhibits higher effectiveness in terms of delaying attacks, which can be seen by the increase in the number of steps required for the attacking agent to reach the peak of the average cumulative reward. This finding is consistent with the results of Balogh et al. [6] who deployed some honeypots in a Capture The Flag (CTF) environment and subsequently collected and analyzed data. Secondly, the defense effects of the honeypot and Shock Trap are almost equivalent in this scenario, but both are superior to the decoy. Thirdly, these three deception techniques show a better performance in the later stages of the attack, as the nodes deploying the deception elements are located in the deep layers of the network. The attacking agent is unaffected until it encounters these deception elements. Finally, none of the three deception techniques significantly affect the DQL algorithm-driven attacking agent, particularly Shock Trap. The role of the Shock Trap is to prevent attackers from entering deep nodes for further attacks, but the DQL algorithm can adaptively adjust the attack strategy. When it finds that a path is blocked, it will attack more from other paths. It is worth noting that the DQL algorithm-based attacking agent outperforms the random search algorithm-based attacking agent after incorporating the deceptive elements , which implies that DQL can learn and adapt better in the presence of defensive mechanisms.

In this section, we validate our opinions regarding reward parameter configuration in Sect. 2.3. First of all, the fluctuation of the negative reward value obtained when the attacker triggers the deception technology for the first time has a minimal impact on the attacker's cumulative reward trend. The cumulative reward trend is mainly affected by the frequency with which the deception element is repeatedly triggered. Secondly, the oscillation in negative reward values resulting from repeated activation of the deception technology by the attacker influences the rate of change in cumulative rewards, yet it does not significantly alter the fundamental trend. Furthermore, a high setting of this negative reward value leads to cumulative reward trends that deviate from the authentic attack and defense scenarios.



(a) Cumulative reward plot when the negative reward for the first trigger is -200.

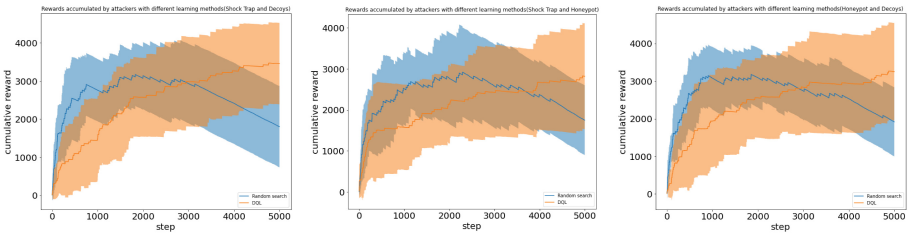
(b) Cumulative reward plot when the negative reward for repeated triggers is -2.

(c) Cumulative reward plot when the negative reward for repeated triggers is -10

Fig. 3. The cumulative reward plot of the agent after the honeypot's reward parameters are modified.

We employed the prior assessment of the honeypot defense efficacy as a reference point and modified the negative reward value assigned when the attacker first triggers the deception technology, setting it to -200. All other control conditions remained unaltered, and we subsequently conducted the controlled experiment. Through a comparative analysis between Fig. 2c, and Fig. 3a, we can deduce that the fluctuation in the negative reward value upon the initial activation of the deception technology has a minimal impact on the attacker's cumulative reward trend. It will only slightly affect the cumulative reward obtained by the attacker throughout the intrusion process. To illustrate, following 5000 steps taken by the attacker, the average cumulative reward for the attack agent driven by the random search algorithm decreased from approximately 2150 to about 2050. Similarly, the average cumulative reward for the attack agent driven by the DQL algorithm also decreased by approximately 100.

In addition, we also conducted another set of controlled experiments, adjusting the negative reward values for the attacker to repeatedly trigger the deception technique to -2 and -10 respectively. During this process, all other control conditions remained constant, and subsequent experimental tests were conducted. By comparing the results in Figs. 2c, and 3b, we can draw the conclusion that for negative reward values that repeatedly trigger deception elements, moderate changes will have a certain impact on the rate of change of the cumulative reward, but will not change its basic trend. In comparing Fig. 2c, 3b, and 3c, it becomes evident that when the negative reward value for the repetitive triggering of deception elements is set at a high level, the attacker's cumulative reward will become negative. This outcome is inconsistent with the realities of cyber security. After conducting numerous experimental iterations, we set the negative reward value for repeated triggering of deception elements to -1. Under such a reward parameter configuration, we evaluate and test the honeypot's defense effect. By comparing Fig. 2a, and 2c, it becomes evident that the average cumulative reward of the attack agent driven by the random search algorithm decreased by approximately 50%, while the average cumulative reward of the attack agent



(a) The cumulative reward plot of the agent under the combination of decoy and Shock Trap.

(b) The cumulative reward plot of the agent under the combination of honeypot and Shock Trap.

(c) The cumulative reward plot of the agent under the combination of honeypot and decoy.

Fig. 4. The cumulative reward plot of the agent under the combination of different deception techniques.

driven by the DQL algorithm decreased by roughly one-third. This observation confirms the efficacy of honeypots in diverting and attracting cyber attacks, aligning with the results of other honeypot evaluation studies [1,2,7,18].

3.2 RQ2: The Effect of Combining Different Deception Defense Technologies

A study conducted in 2021 [20] showed that deploying the same defense technology at multiple nodes in CyberBattleSim would affect defense effectiveness. To better evaluate the effect of different combinations of deception defense techniques, we conducted an experiment in the TinyCTF scenario. In the experiment, we combined Shock Trap, honeypot, and decoy deception defense technologies in pairs, and deployed them on the same node (GitHubProject) to eliminate interference from the number of deception elements in the experimental results. Based on the experimental results illustrated in Fig. 4, it can be concluded that employing a combination of deception elements within nodes enhances the overall defense effectiveness compared to their individual deployment. However, it is crucial to note that the degree of improved defensive outcomes varies across different combinations of deception elements. Specifically, the combination of Shock Trap and honeypot exhibits greater efficacy in comparison to the pairing of decoy and honeypot.

The analysis of Figs. 2b,2c, and 4c reveals significant conclusion. When defending against agents driven by a random search algorithm, whether the decoy and honeypot are deployed individually or combined in the same node, they all exhibit similar defensive effects. This similar defensive effect is mainly manifested in the ability to delay the action of the attacking agent and the effect of suppressing the cumulative reward of the attacking agent. By comparing Fig. 2, 4a, and 4b, we can draw the following conclusions. The defense effect can be significantly improved by combining Shock Trap with either the honeypot or decoy, compared to their individual deployment. Particularly, the combined deployment

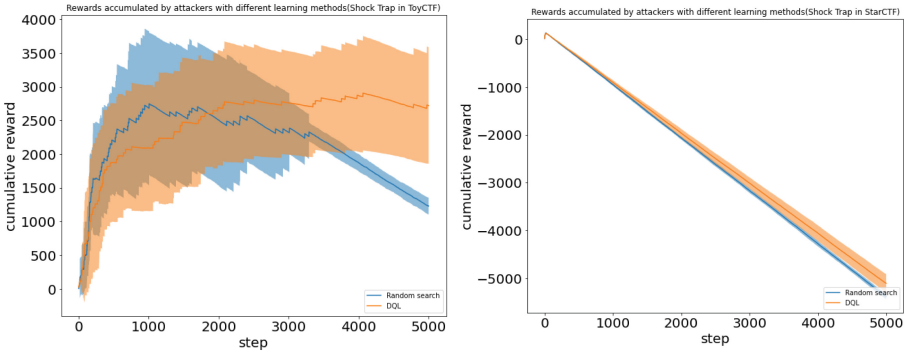
of the Shock Trap and honeypot yields the best effect. When combating agents driven by random search algorithms, the combined deployment of Shock Trap and honeypot can effectively delay their actions and significantly decrease their cumulative reward value. Meanwhile, when dealing with the attacking agent driven by the DQL algorithm, it can restrain the upward trend of the attacking agent's cumulative reward and lower its peak value. The effectiveness of the honeypot in countering the adaptive strategies of DQL complements the limitations of the Shock Trap. Specifically, the honeypot serves as a magnet for attracting attack agents toward the node where Shock Trap is deployed. The Shock Trap, in turn, can prevent attackers from further breaching the node, thereby preventing subsequent attacks. When the attacking agent realizes the futility of launching attacks, it will adaptively adjust its strategy and choose to control the network from alternative paths. However, the coexistence of the honeypot within the Shock Trap-deployed node continues to allure attack agents, leading to a repetitive cycle of attack attempts until the agent eventually ceases its attack. Hence, we conclude that the combination of Shock Trap and honeypot can maximize its defensive effect. This finding also demonstrates the compatibility of the Shock Trap, which can be effectively combined with other deception defense technologies to enhance defense capability. The analysis and comparison of defense effects achieved through different combinations of defense technologies can offer valuable references for cyber defense practitioners in devising effective defense systems.

3.3 RQ3: Shock Trap Applicable Scenarios

To investigate the effectiveness of Shock Trap in different network scenarios and identify the optimal application scenarios, we conducted experiments using three network topologies with varying complexity: TinyCTF, ToyCTF, and StarCTF. The TinyCTF topology, which was introduced in Sect. 2.1, includes two attack paths for potential attackers. ToyCTF is a simulated environment provided by CyberBattleSim, which is very similar to TinyCTF but only contains one attack path. We also introduced a simple scenario, StarCTF, in which a client node can only access subsequent data nodes through a key service node. The three network topologies used are representative. After moderate abstraction and combing, the architecture of most enterprises can be represented by different variants of these network topologies. The effectiveness of Shock Trap in different network scenarios was evaluated by deploying it at the same position in ToyCTF and TinyCTF scenarios and comparing the results. The experimental results are shown in Fig. 2d, and Fig. 5a. It can be seen from the results that Shock Trap was able to effectively prevent attackers from attacking nodes in ToyCTF, where only one attack path was present. However, in TinyCTF, where multiple attack paths existed, Shock Trap had limited defense capabilities as attackers could bypass it through other paths.

To validate our conclusions, we conducted an experiment in which Shock Trap was deployed at the key service nodes of the StarCTF scenario. As shown

in Fig. 5b, the results demonstrate that Shock Trap was highly effective in preventing the agent's attack, leaving it unable to obtain any cumulative rewards and only subjected to constant punishment. This outcome differs from the results obtained in the ToyCTF scenario, where the agent was able to gain some accumulated rewards by exploiting other vulnerable nodes before encountering Shock Trap. From these experiments, it was concluded that Shock Trap is suitable for deployment at key facilities such as routers or switches that can connect multiple systems and play a crucial role in the network topology. Moreover, the approach employed in this study provides references for cyber defense personnel in identifying the applicable scope of the technology and expeditiously identifying the optimal application scenario, all while minimizing costs and streamlining environment configuration.



(a) The cumulative reward graph of the agent after deploying Shock Trap in the ToyCTF.

(b) The cumulative reward plot of the agent under Shock Trap deployment in the StarCTF.

Fig. 5. The cumulative reward plot of the agent after deploying Shock Trap in different scenarios.

4 Related Work

With the continuous evolution of offensive and defensive confrontations, novel deception defense technologies are constantly emerging, such as Chaff Bug and Shock Trap. To enhance the effective application of these defense technologies, it becomes imperative to undertake an evaluation of their practical efficacy. This necessitates the utilization of methodologies encompassing game theory, simulation techniques, and real-world scenario deployments. Game theory-based approaches provide a theoretical foundation for comprehending the role of deception technology, but the insights provided are relatively abstract, and the evaluation results may deviate from the actual situation. Aggarwal et al. [2,3] proposed a non-cooperative dynamic deception game. Through experiments and game theory modeling, they deeply studied the proportional impact of the timing and

quantity of deception on attack behavior and non-attack behavior. However, this method is still limited to the theoretical level, and difficult to directly map to practical applications. The deployment of deception techniques in real-world scenarios, coupled with data collection, is currently regarded as the most precise evaluation approach. Nonetheless, this strategy presents several drawbacks, including substantial time and resource demands and the difficulty in controlling factors within the assessment environment. Ferguson-Walter et al. [10] analyzed four separate incidents roughly six months apart by tracking a three-person red team on a live operational network to assess the impact on attackers of the presence of decoys and being informed of the decoys. This research can provide insights into actual adversarial behavior but has limitations such as the small number of participants and the absence of control conditions. HAN et al. [11] assessed deception techniques in web applications by quantifying false positives in a production environment and evaluating detection accuracy in a controlled red team experiment involving 150 participants. This approach strikes a balance between real-world environments and controlled experiments but entails substantial manpower and time resources. In 2022, Kocaogullar et al. [15] deployed multiple honeypots within an actual network environment, collecting data over 14 days. Through an analysis of the gathered data, they explored the impact of high-interaction honeypots and low-interaction honeypots on attacks, including the types and proportions of attacks attracted, IP addresses captured, etc. Ferguson-Walter et al. [8,9] designed a network penetration testing experiment involving over 130 professional red team members. Their study analyzed the impact of cyber deception and psychological deception on attackers and evaluated the effectiveness of the decoy. This experimental approach takes into account the complexity of real-world scenarios and provides more specific insights into the evaluation and practical application of deception techniques.

Simulation techniques are an assessment method between theoretical and practical for evaluating deception techniques. This method can obtain relatively accurate assessment results in a short time while conserving substantial time and resources. Aggarwal et al. [1] proposed a simulation tool called "HackIt" to evaluate the effectiveness of honeypot deception timing in mitigating cyber attacks. The results obtained through this simulation approach are consistent with the conclusions obtained through dynamic deception games [2]. Balogh et al. [6] used the Capture The Flag (CTF) game to analyze and evaluate the effectiveness of honeypots in affording defenders additional time for detection and response, as well as their impact on delaying attackers. Compared with real-world scenarios, the utilization of CTF games for evaluating defense effects offers heightened controllability and facilitates more straightforward data collection. Compared to simulation tools such as "HackIt", CTF games provide more realistic datasets and simulations of attacker and defender behavior. In this study, we employ Microsoft's open-source cyber attack and defense simulation tool, CyberBattleSim [19], for a comprehensive evaluation of multiple deception technologies' defense effects. Other research on the CyberBattleSim platform also confirmed the feasibility of using CyberBattleSim to evaluate the efficacy of deception tech-

niques. For example, Amin et al. [4] defined two types of attackers within the CyberBattleSim platform and explored the impact of deploying the decoy on these distinct attacker types.

The most closely related work to our research is a 2021 paper [20]. Their work incorporated multiple deception techniques into CyberBattleSim and investigated their impact on attacking agents driven by different algorithms. Moreover, they examined how the quantity and location of deception elements within the network affect the overall effectiveness of defensive measures. Drawing inspiration from the methods presented by Walter et al. [20], our work has innovated and expanded in many aspects. Firstly, our study goes beyond evaluating and comparing the individual defensive capabilities of different deception technologies. We conduct experiments to assess the defense effects achieved through the combination of various defense techniques. Secondly, we investigate leveraging platforms like CyberBattleSim to streamline the testing and identification of suitable application scenarios for emerging defense technologies. By utilizing such platforms, we can accelerate the evaluation process and help to identify the most promising deployment application scenarios for new defense technologies. Finally, we integrate a novel defense technology known as Shock Trap into the CyberBattleSim platform. Through a comparative analysis, we examine the distinguishing characteristics of the Shock Trap in contrast to other defense technologies like the honeypot and decoy.

5 Discussion

We incorporated deception techniques such as honeypot, decoy, and Shock Trap into the CyberBattleSim platform, and simulated the punishment of deception techniques on attacker behavior by introducing different negative reward mechanisms. We assess the defense effectiveness by comparing the impact of deception defense techniques on the attacker's cumulative reward. The results demonstrate that platforms such as CyberBattleSim can not only provide a rapid verification and testing environment for new defense techniques but also facilitate the identification of optimal defense technique combinations.

This study has made progress in assessing the effectiveness of deception technologies and identifying appropriate scenarios and combinations for defense technologies. However, some limitations and inadequacies need to be acknowledged. Firstly, the study only involves three deception defense technologies and has a small sample size, which limits its representativeness for the broader range of deception technologies. Secondly, there are certain differences between simulated scenarios and actual scenarios. For instance, the efficacy of defense technologies in simulated scenarios may align with theoretical expectations, but in the actual environment, managers' configuration errors may affect the defense effect, leading to differences. Additionally, system values vary in real-world scenarios, whereas simulated environments employ uniform node values, thereby deviating from the complexities of actual situations. Finally, the research is conducted within the CyberBattleSim platform, subject to some inherent constraints and limitations, mainly reflected in the following aspects:

- After turning on negative rewards in CyberBattleSim, the cumulative rewards of attacking agents are difficult to converge, and the introduction of deception elements further exacerbates this problem.
- The evaluation of defense effects primarily relies on a single index, namely the change in cumulative reward for attacking agents, which may not fully reflect the impact of deception elements on attackers.
- CyberBattleSim primarily analyzes the lateral movement cyber attack technology within the intranet, which limits its application scenarios.
- The limitations of the CBS platform prevent it from accurately simulating complex attack methods and vulnerability modes, resulting in a restricted range of simulated attack agents and vulnerability types. Consequently, the platform fails to fully reflect real-world attackers and vulnerabilities.

Given the above problems, future research work will focus on improving and exploring the following aspects. Firstly, evaluating and researching more deception defense technologies and other cyber defense technologies on the CyberBattleSim platform, to better understand their defensive effects, strengths and weaknesses, and synergies. Secondly, adopt more evaluation methods and standards. Besides using the cumulative reward as the evaluation index of the attack effect, the number of simulation steps required for an agent to control a network will also be employed. The fewer steps an agent takes, the better its attack strategy. Thirdly, an open challenge currently facing simulation technology is the discrepancy between simulation evaluation results and actual scenario evaluation results. In subsequent work, we will assign different values to different nodes in the simulation environment to reduce this difference, to provide more accurate and comprehensive guidance for the deployment of defense technology in real scenarios. Finally, enhancing the CyberBattleSim platform by abstracting and modeling the real network traffic, will enable simulations of more diverse network scenarios.

6 Conclusions

Based on CyberBattleSim, an open-source simulation tool from Microsoft, this paper introduces and investigates deception technologies such as decoy, honeypot, and Shock Trap in an abstract enterprise network topology environment. The objective is to explore the impact of these deception technologies individually or in combination on attacker agents. In addition, we also designed three different network topologies within the CyberBattleSim platform and deployed Shock Trap in each of them for experiments. The experimental results show the effectiveness of deception defense technologies, including the Shock Trap, honeypot, and decoy, in effectively delaying and preventing attackers. In particular, deploying the Shock Trap in combination with other deception defense technologies such as decoy or honeypot can significantly improve the defense effect. Furthermore, the experiments conducted within the CyberBattleSim platform, involving the deployment of Shock Trap in various network topologies, reveal

that it is suitable for deployment in facilities that can connect multiple systems and play an important role in the network topology.

To sum up, enterprises need to invest a lot of manpower and time resources to evaluate and verify the effectiveness of defense technologies, and there are many uncontrollable factors during the evaluation process, that may pose threats to network security. This research presents a method for identifying suitable application scenarios and optimal combinations of defense techniques while minimizing deployment costs and simplifying environment configurations. The research results provide a valuable reference for security teams to design and implement multi-layered defense systems within real-world environments. In addition, based on the results of this study, security teams can systematically deploy and verify defense technologies in actual scenarios, thus yielding substantial time and workload savings. This strategic approach also facilitates the deployment of a defense system harmonious with the current network architecture, thereby maximizing its defense efficacy.

References

1. Aggarwal, P., Gautam, A., Agarwal, V., Gonzalez, C., Dutt, V.: *HackIT*: a human-in-the-loop simulation tool for realistic cyber deception experiments. In: Ahram, T., Karwowski, W. (eds.) AHFE 2019. AISC, vol. 960, pp. 109–121. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20488-4_11
2. Aggarwal, P., Gonzalez, C., Dutt, V.: Cyber-security: role of deception in cyber-attack detection. In: Nicholson, D. (ed.) AHFE 2016. AISC, vol. 501, pp. 85–96. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-41932-9_8
3. Aggarwal, P., Gonzalez, C., Dutt, V.: Modeling the effects of amount and timing of deception in simulated network scenarios. In: 2017 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA), pp. 1–7. IEEE (2017)
4. Al Amin, M.A.R., Shetty, S., Kamhoua, C.: Cyber deception metrics for interconnected complex systems. In: 2022 Winter Simulation Conference (WSC), pp. 473–483. IEEE (2022)
5. Almeshekah, M.H., Spafford, E.H., Atallah, M.J.: Improving security using deception. Technical report, CERIAS technical report 13. Center for Education and Research Information Assurance and Security, Purdue University (2013)
6. Balogh, Á., Érsok, M., Erdődi, L., Szarvák, A., Kail, E., Bánáti, A.: Honeypot optimization based on CTF game. In: 2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI), pp. 000153–000158. IEEE (2022)
7. Crouse, M., Prosser, B., Fulp, E.W.: Probabilistic performance analysis of moving target and deception reconnaissance defenses. In: Proceedings of the Second ACM Workshop on Moving Target Defense, pp. 21–29 (2015)
8. Ferguson-Walter, K., Major, M., Johnson, C.K., Muhleman, D.H.: Examining the efficacy of decoy-based and psychological cyber deception. In: USENIX Security Symposium, pp. 1127–1144 (2021)
9. Ferguson-Walter, K., et al.: The Tularosa study: an experimental design and implementation to quantify the effectiveness of cyber deception. Technical report, Sandia National Lab. (SNL-NM), Albuquerque, NM, United States (2018)

10. Ferguson-Walter, K., LaFon, D., Shade, T.: Friend or faux: deception for cyber defense. J. Inf. Warfare **16**(2), 28–42 (2017)
11. Han, X., Kheir, N., Balzarotti, D.: Evaluation of deception-based web attacks detection. In: Proceedings of the 2017 Workshop on Moving Target Defense, pp. 65–73 (2017)
12. Han, X., Kheir, N., Balzarotti, D.: Deception techniques in computer security: a research perspective. ACM Comput. Surv. (CSUR) **51**(4), 1–36 (2018)
13. Hong, Q., Zhao, Y., Chang, J., Du, Y., Li, J., Zhai, L.: Shock trap: an active defense architecture based on trap vulnerabilities. In: 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC), pp. 24–31. IEEE (2022)
14. Hu, Z., Hu, Y., Dolan-Gavitt, B.: Towards deceptive defense in software security with chaff bugs. In: Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses, pp. 43–55 (2022)
15. Kocaogullar, Y., Cetin, O., Arief, B., Brierley, C., Pont, J., Hernandez-Castro, J.C.: Hunting high or low: evaluating the effectiveness of high-interaction and low-interaction honeypots (2022)
16. Lu, Z., Wang, C., Zhao, S.: Cyber deception for computer and network security: survey and challenges. arXiv preprint arXiv:2007.14497 (2020)
17. Provos, N., et al.: A virtual honeypot framework. In: USENIX Security Symposium, vol. 173, pp. 1–14 (2004)
18. Robertson, W.: Using web honeypots to study the attackers behavior. Ph.D. thesis, TELECOM ParisTech (2017)
19. Microsoft Defender Research Team: Cyberbattlesim. https://github.com/microsoft/cyberbattlesim (2021). Created by Christian Seifert, Michael Betser, William Blum, James Bono, Kate Farris, Emily Goren, Justin Grana, Kristian Holsheimer, Brandon Marken, Joshua Neil, Nicole Nichols, Jugal Parikh, Haoran Wei
20. Walter, E., Ferguson-Walter, K., Ridley, A.: Incorporating deception into cyberbattlesim for autonomous defense. arXiv preprint arXiv:2108.13980 (2021)
21. Wikipedia: Honeypot (computing) (2022). https://en.wikipedia.org/wiki/Honeypot_(computing)