

# Cyber Deception against Zero-day Attacks: A Game Theoretic Approach<sup>\*</sup>

Md Abu Sayed<sup>1</sup>[0000–0002–5560–9150], Ahmed H. Anwar<sup>2</sup>[0000–0001–8907–3043],  
 Christopher Kiekintveld<sup>1</sup>[0000–0003–0615–9584], Branislav  
 Bosansky<sup>3</sup>[0000–0002–3841–9515], and Charles Kamhoua<sup>2</sup>[0000–0003–2169–5975]

<sup>1</sup> University of Texas at El Paso, TX 79968, USA  
 msayed@miners.utep.edu, cdkiekintveld@utep.edu

<sup>2</sup> US Army Research Laboratory, MD 20783, USA  
 a.h.anwar@knights.ucf.edu, charles.a.kamhoua.civ@mail.mil

<sup>3</sup> Department of Computer Science, Faculty of Electrical Engineering,  
 Czech Technical University in Prague  
 branislav.bosansky@fel.cvut.cz

**Abstract.** Reconnaissance activities precedent other attack steps in the cyber kill chain. Zero-day attacks exploit unknown vulnerabilities and give attackers the upper hand against conventional defenses. Honeypots have been used to deceive attackers by misrepresenting the true state of the network. Existing work on cyber deception does not model zero-day attacks. In this paper, we address the question of "How to allocate honeypots over the network?" to protect its most valuable assets. To this end, we develop a two-player zero-sum game theoretic approach to study the potential reconnaissance tracks and attack paths that attackers may use. However, zero-day attacks allow attackers to avoid placed honeypots by creating new attack paths. Therefore, we introduce a sensitivity analysis to investigate the impact of different zero-day vulnerabilities on the performance of the proposed deception technique. Next, we propose several mitigating strategies to defend the network against zero-day attacks based on this analysis. Finally, our numerical results validate our findings and illustrate the effectiveness of the proposed defense approach.

**Keywords:** Cyber deception · Game theory · Zero-day attacks.

---

<sup>\*</sup> Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Numbers W911NF-19-2-0150 and W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Branislav Bosansky was also supported by the Czech Science Foundation (no. 19-24384Y).

## 1 Introduction

The cyber kill chain defines seven stages of cyber attack that end with gaining control of a system/network and infiltrating its data [1]. The first stage is the reconnaissance stage in which an adversary collects valuable information regarding the network topologies, structures, node features, and the important assets of the system. To achieve this goal, an attacker may use active sensing techniques and/or passive sensing techniques. The latter can observe traffic between servers and clients to infer information from packet length, packet timing, web flow size, and response delay [2], it is difficult to detect and is invisible to the hosts running the services and can be difficult to be detected by conventional IDS. Active probing attacks send packets to a host and analyze its response. Hence, the attacker learns the system information and vulnerabilities [3]. The active reconnaissance is faster and identifies open and unprotected ports [4]. On the other side, it is more aggressive and intrusive, and hence can be detected. Also, attackers may mix between active and passive attacks during the reconnaissance stage. Game theory provides a suitable framework for modeling attacks and defense against several attacks [5,6,7].

**Cyber deception:** Cyber deception is an emerging proactive cyber defense technology it provides credible yet misleading information to deceive attackers. Deception techniques have been used in the physical space as a classical war technique. However, deception has recently been adopted into cyberspace for intrusion detection and as a defense mechanism [8]. Cyber deception shares some characteristics of non-cyber deception and follows similar philosophical and psychological traits. Successful deception relies on understanding the attacker's intent, tools and techniques, and mental biases. The first step to achieve this deep level of understanding is to act proactively aiming to capture the attacker and exploit the opportunity to monitor her behavior. For that purpose, honeypots are effectively used as fake units in the system/network that deceive the attacker and allow the defender to study her attack strategy, and intent in order to design a better deception scheme.

**Honeypots:** Among many other techniques, honeypots are widely used for cyber deception. Honeypots are fake nodes that can mislead attackers and waste their resources. They are categorized into two levels, namely low-interaction honeypots and high-interaction honeypots. Low interaction honeypots can mimic specific services and are virtualized, and hence they are easier to build and operate than high interaction honeypots. However, they can be detected by adversaries more easily [9].

**Attack Graph:** An attack graph (AG) is a graph-based technique for attack modeling. Attack modeling techniques model and visualize a cyber-attack on a computer network as a sequence of actions or a combination of events [10]. Attack graphs (trees) are a popular method of representing cyber-attacks. There exist no unique way to instantiate attack graphs, authors in [10] surveyed more than 75 attack graph visual syntax and around 20 attack tree configurations. A key challenge of generating attack graphs is the scalability [11]. None of the existing works has shown that the graph generation tool can scale to the size of an

enterprise network. In this work, we consider a simplified attack graph where each node represents a vulnerable host in the network (i.e., it suffers one or more vulnerabilities), and edges on the attack graph represent specific exploits that provide reachability to the attacker from one host to another. In this sense, the graph scale is in the order of the size of the original network. Although this model does not explicitly model each vulnerability in the network, however, it sufficiently illustrates the attack paths that can be exploited by adversaries which are an essential input to generating optimal honeypot allocation policy. However, attack graphs can not directly model zero-day attacks since it remain unknown to the graph generating tool. Therefore, we propose parallel

**Zero-day attack:** The challenge with defending against zero-day attacks is that these attacks exploit a vulnerability that has not been disclosed. There is almost no defense against exploiting such unknown vulnerability [12]. In this work, we leverage attack graphs to model potential zero-day attacks. If the considered network suffers a zero-day vulnerability then the corresponding attack graph will have some edges and hence attack paths that are unknown to the defender. Moreover, zero-day attacks are used for carrying out targeted attacks. To the best of our knowledge, this represents a new framework to proactively defend against zero-day attacks via strategic honeypot allocation based on game theory and attack graphs.

**Contributions:** In this paper, we propose a cyber deception technique using strategic honeypot allocation under limited deception budget. We consider a game theoretic approach to characterize the honeypot allocation policy over the network attack graph. We then evaluate the deception allocation policy under zero-day attacks by introducing several vulnerabilities to the attack graph and study the sensitivity of the different potential vulnerabilities on the attacker and defender game reward. In our analysis, the defender has no information regarding the zero-day vulnerability. We identify the most impactful vulnerability location and introduce several mitigating strategies to address the possible zero-day attack. The developed game model accounts for the network topology and different importance to each node. We summarize our main contribution below:

- We formulate a deception game between defender and attacker to study the effectiveness of cyber deception against lateral move attacks. The game is played on an attack graph to capture relation between node vulnerabilities, node importance, and network connectivity. We characterize a honeypot allocation policy over the attack graph to place honeypots at strategic locations.
- We evaluate the proposed deception approach against zero-day attack under asymmetric information where the attack graph is not fully known by the defender. We conduct sensitivity analysis to identify critical locations that have major impact on the deception policy in place.
- We present three mitigating strategies against zero-day attacks to readjust the existing honeypot allocation policy based on the conducted analysis.
- Finally, we present numerical results for the developed game model to show the effectiveness of cyber deception as well as the zero-day attack mitigating strategies.

The rest of the paper is organized as follows. We discuss related work in section 2. In section 3, we present the system model, define the game model, and propose our deception approach. In section, 5 we present zero-day attack mitigating strategies. Our numerical results are presented in section 6. Finally, we conclude our work and discuss future work in section 7.

## 2 Related work

Our research builds upon existing work on cyber deception and games on attack graphs to model zero-day attacks and characterize game-theoretic mitigation strategies.

### 2.1 Cyber Deception GT:

Game theoretic defensive deception [13] has been widely discussed in cybersecurity research. Authors in [14] presented a deception game for a defender who chooses a deception in response to the attacker's observation, while the attacker is unaware or aware of the deception. Authors in [15,5] proposed a signaling game based model to develop a honeypot defense system against DoS attacks. Hypergame theory [16] has been used as an extensive game model to model different subjective views between players under uncertainty. [17] explored hypergames for decision-making in adversarial settings. Authors in [18],[19] leveraged hypergames to quantify how a defensive deception signal can manipulate an attacker's beliefs.

### 2.2 Games on AG:

Game Theory (GT) provides a suitable framework to study security problems including cyber deception [20]. Modeling the attacker behavior allows the network admin to better analyze and understand the possible interactions that may take place over cyberspace. Security games are defender-attacker games, the defender allocates a limited set of resources over a set of targets. On other hand, the attacker goal is to attack and gain control over these targets [21]. Resource allocation problems are usually modeled as Stackelberg game where the defender leads the course of play. We consider two-player zero-sum games acting simultaneously, hence the Nash equilibrium of the game coincides with that of the Stackelberg game. Moreover, most of the resource allocation problems considered had no underlying network structure. For the cyber deception problem considered we play a security game on an attack graph which imposes a structure on the players reward function and defines the action space for both players as will be discussed in Section 3.

### 2.3 Zero-day

A zero-day attack is a cyber attack exploiting a vulnerability that has not been disclosed publicly [12]. Due to the challenges associated with zero-day attacks,

Eder-Neuhauser et al. [22] introduced three novel classes of malware that are suitable for smart grid attacks. Their model provides a basis for the detection of malware communication to predict malware types from existing data. They suggest proper update and segmentation policies for anomaly detection. However, such an approach does not capture the dynamics of zero-day attacks or model its usage in lateral movement attacks.

### 3 System Model

```

graph TD
    1((1)) -- yellow --> 2((2))
    1((1)) -- yellow --> 3((3))
    2((2)) -- yellow --> 4((4))
    2((2)) -- red --> 3((3))
    2((2)) -- red --> 5((5))
    3((3)) -- red --> 2((2))
    3((3)) -- red --> 5((5))
    3((3)) -- yellow --> 6((6))
    3((3)) -- red --> 7((7))

```

Fig. 1: 7-node tree network topology with single point of entry and two target nodes (5,7) and zero-day vulnerability(2,3) and (3,5).

An edge connecting node  $u$  and node  $v$ , represents an exploitable vulnerability that allows an adversary to launch an attack to reach node  $v$  from a compromised node  $u$ . In this setting, we adopt a slightly different version of the attack graph introduced in [24]. In other words, in this graph, each node represents a host that suffers one or more vulnerabilities that could be exploited to reach a neighboring node. Hence, the edge models the connecting link that could be used by malicious users to reach the next victim node. A legitimate user at node  $v$ , will have the right credentials to reach node  $u$ . However, an adversary will only reach  $u$  through an exploitable vulnerability. For each node  $i \in \mathcal{N}$  we assign a value  $v(i)$ . Hence,  $G_1(\mathcal{N}, \mathcal{E}_1)$  is an attack graph assumed to be known to the defender and the attacker.

A zero-day attack vulnerability is exclusively known to the attacker. The effect of a single zero-day vulnerability is an additional edge. This generates a different attack graph perceived by the attacker solely. Let  $G_2(\mathcal{N}, \mathcal{E}_2)$  denote the attack graph induced via zero-day vulnerability  $e$ , such that,  $\mathcal{E}_2 = \mathcal{E}_1 + e$ . The attacker plays a game on the graph with an additional edge(s) representing the zero-day vulnerability. In other words, considering a single zero-day vulnerability at a time,  $G_2 = G_1 + \{e\}$ , where  $\{e\}$  is the new edge due to zero-day vulnerability.

Fig. 1 denotes 7-node tree attack graph consists of one entry node(1), 4 intermediate nodes(2,3,4,6) and 2 target nodes(5,6). In this network, one available path for reaching every target nodes. However, with two zero-day vulnerabilities(2,3) and (3,5) increases the available attack path to every target node from attacker perspective.

### 3.1 Defender model

The defender allocates one or more honeypots along the network edges as fake vulnerabilities to capture malicious traffic and illegitimate users. Let  $H$  denote the honeypot budget. The defender's action is to allocate up to  $H$  honeypots over the different edges. Therefore, the defender action space  $\mathcal{A}_d = \{\mathbf{e} \in 2^{\mathcal{E}} \mid \mathbf{e}^T \mathbf{1} \leq H\}$ . Where,  $\mathbf{e}$  is a binary vector of length  $|\mathcal{E}|$ , such that the  $i^{th}$  entry  $\mathbf{e}(i) = 1$  indicates a honeypot is allocated along the  $i^{th}$  edge, and is set to zero otherwise. The defender incurs a cost associated with each action that takes into account the number of allocated honeypots. Otherwise, the defender will always max out the number of allocated honeypots. Let  $C_d$  denote the cost per honeypot. Hence, the total cost is  $C_d \times \|a_d\|_1$ , where  $\|a_d\|_1$  is the number of honeypots allocated by  $a_d$ . The defender tries to reduce the attacker reward via placing more honeypots at the edges of high potential that are attractive to attacker, while reducing the total deception cost.

### 3.2 Attacker model

The attacker is assumed to launch a targeted attack. Therefore, she selects one of the possible attack paths to reach a target node to maximize his/her expected reward. Hence, the attacker's action space,  $\mathcal{A}_a$ , is the set of all the possible attack paths starting at an entry node  $u \in E$  to a target node  $v \in T$ . The attacker pays

an attack cost that depends on the selected attack path. We consider a cost due to traversing a node in the attack graph denoted by  $C_a$ . The attacker faces a tradeoff between traversing through important nodes while reducing his overall attack cost.

### 3.3 Reward function

We define the reward function to capture the tradeoff that faces each player. For each action profile played by the defender and attacker  $(a_d, a_a) \in \mathcal{A}_d \times \mathcal{A}_a$ , the defender receives a reward  $R_d(a_d, a_a)$  and the attacker reward is  $R_a(a_d, a_a)$ . The defender is interested in protecting specific nodes than others. Recall that each node  $i \in \mathcal{N}$  is assigned a value  $v(i)$  that reflect its importance for the attacker, the defender gains more by protecting high valued nodes. On the other hand, the attacker reward increases when attacking nodes of high values along the selected attack path.

The defender reward is expressed as:

$$R_d(a_d, a_a) = \sum_{i \in a_a} Cap \cdot v(i) \cdot \mathbf{1}_{\{i \in a_d\}} - Esc \cdot v(i) \cdot \mathbf{1}_{\{i \notin a_d\}} - C_d \cdot \|a_d\|_1 + C_a(a_a) \quad (1)$$

where  $Cap$  denotes the capture reward received by the defender when the attacker hits a honeypot along the selected attack path  $a_a$ .  $Esc$  is the attacker gain upon successful attack from one node to another in the way toward the target node. Finally,  $C_d$  and  $C_a(a_a)$  are the cost per honeypot, and attack cost, respectively. The attack cost is proportional to the length of the attack path as the attacker could become less stealthy due to numerous moves. We consider a zero-sum game where  $R_a + R_d = 0$ . Now we readily define a two-player zerosum game  $\Gamma(\mathcal{P}, \mathcal{A}, \mathcal{R})$ , where  $\mathcal{P}$  is the set of the two players (i.e., defender and attacker). The game action space  $\mathcal{A} = \mathcal{A}_d \times \mathcal{A}_a$  as defined above, and the reward function  $\mathcal{R} = (R_d, R_a)$ .

Due to the combinatorial nature of the action spaces in terms of the network size, characterizing a Nash equilibrium (NE) in pure strategy is challenging. However, the finite game developed above, holds a NE in mixed strategies. Let  $\mathbf{x}_1$  and  $\mathbf{y}_1$  denote the mixed strategies of defender, and attacker, when the game is played on known-to-all graph,  $G_1$ . The defender expected reward of game 1 (i.e., game on  $G_1$  with no zero-day vulnerabilities) is expressed as:

$$U_d(G_1) = \mathbf{x}_1^T R_d(G_1) \mathbf{y}_1 \quad (2)$$

where  $R_d(G_1)$  is the matrix of the game played on  $G_1$  and the attacker expected reward  $U_a(G_1) = -U_d(G_1)$ . Both defender and attacker can obtain their NE

mixed strategies  $\mathbf{x}_1^*$  and  $\mathbf{y}_1^*$  via a linear program (LP) as follows,

$$\begin{aligned}
& \underset{\mathbf{x}}{\text{maximize}} && U_d \\
& \text{subject to} && \sum_{a_d \in \mathcal{A}_d} R_d(a_d, a_a) x_{a_d} \geq U_d, \quad \forall a_a \in \mathcal{A}_a. \\
& && \sum_{a_d \in \mathcal{A}_d} x_{a_d} = 1, \quad x_{a_d} \geq 0,
\end{aligned} \tag{3}$$

where  $x_{a_d}$  is the probability of taking action  $a_d \in \mathcal{A}_d$ .

Similarly, the attacker's mixed strategy can be obtained through a minimizer LP under  $\mathbf{y}$  of  $U_d$ .

## 4 Zero-day Vulnerability Analysis

We conduct a zero-day vulnerability analysis by modifying the original graph  $G_1$  via considering one vulnerability at a time. The goal of this analysis is to identify the most critical zero-day vulnerability in terms of the impact of each vulnerability to the attacker's reward against a base deception strategy. The base deception strategy is  $\mathbf{x}_1$  that is obtained from the game played on  $G_1$ . In other words, the attacker expected reward is  $U_a(G_1) = \mathbf{x}_1^T R_a(G_1) \mathbf{y}_1$ , for any game 1 mixed strategies  $\mathbf{x}_1$  and  $\mathbf{y}_1$ , and  $U_a(G_2) = \mathbf{x}_2^T R_a(G_2) \mathbf{y}_2$  for the game played on  $G_2$ . The game played under  $G_1$  is referred to as game 1, and the game played on  $G_2$  is referred to as game 2. Where,  $\mathbf{x}_2$  and  $\mathbf{y}_2$  denote the mixed strategies of the game played on the  $G_2$  graph (i.e., under zero-day vulnerability).

However,  $x_2$  is infeasible in practice for the defender since the defender has no information about the zero-day vulnerability nor its location. However, the attacker's action space expands to contain additional attack paths induced by zero-day vulnerabilities. Each of these vulnerabilities may produce one or more new attack path leading to the target node.

Although the defender does not actually know that the network suffers a zero-day vulnerability at a specific location, he may have the knowledge that such vulnerability exists. Therefore, the attacker is not fully certain that this specific vulnerability is unknown to the defender. The attacker uncertainty regarding the defender knowledge leads to two possible game settings and hence two evaluation criteria as follows:

- The first criterion considers an attacker that uncertain whether his opponent knows about the zero-day vulnerability. In fact, the defender has no such information, yet the attacker accounts for some infeasible defender actions. We refer to this criterion as '*optimistic*'. Hence, the expected game value for the attacker,  $U_a^{opt}(G_2) = \hat{\mathbf{x}}_1^T R_a(G_2) \mathbf{y}_2$ , where  $\hat{\mathbf{x}}_1$  is a mixed strategy adopted from  $\mathbf{x}_1$  and padded with zeros to ensure proper matrix multiplication while it zero-enforce infeasible actions for the defender.



- Secondly, we consider a ‘*pessimistic*’ criterion, where the attacker is certain that the defender does not know the zero-day vulnerability. Hence, the defender action space is exactly similar to game 1. The expected reward is  $U_a^{pes}(G_2) = \mathbf{x}_1^T R_a(G_2) \mathbf{y}_2$ . This pessimistic criterion is referred to as game 3.

*Remark 1.* Considering one zero-day vulnerability at a time allows the defender to study the impact of each vulnerability separately, reduces the game complexity, and enables parallel analysis by decoupling the dependencies between different vulnerabilities.

As explained above, we augment zeros to  $\mathbf{x}_1$  to restrict the defender honeypot allocation strategy and make the defender strategy consistent with the  $R_a(G_2)$  matrix.

Let,  $\mathbf{x}^1 = [x_0, x_1, x_2, \dots, x_r]$ , and  $\hat{\mathbf{x}}_1 = [x_0, x_1, x_2, \dots, x_r, \dots, x_n]$ . Hence,  $\hat{\mathbf{x}}_1 = [\mathbf{x}^1, \dots, x_n]$ , where  $n \geq r$ , and value of all strategies from  $x_{r+1}$  to  $x_n$  will be zero after sorting the corresponding actions in  $\hat{\mathbf{x}}_1$ . For the pessimistic case (i.e., game 3), the defender is forced to play the base deception strategy  $\mathbf{x}_1$  in which he also deviates from the NE of game 3.

We solve one game corresponding to each zero-day vulnerability. Assume we have a set of possible zero-day vulnerabilities  $\mathcal{E}_0$  such that  $G_2(e) = G_1 + e$  ;  $\forall e \in \mathcal{E}_0$ . For each game we record the expected attack reward, hence we sort the vulnerabilities to find the most impactful that results in the highest increase of the attacker’s reward.

Without loss of generality, assuming the new vulnerability introduced one new pure strategy  $a_e$  for the attacker (if  $e \in \mathcal{E}_0$  induces more than one new attack path, we select  $a_e$  to be the path that has higher reward), then we can establish the following theorem.

**Theorem 1.** *For the game  $\Gamma$  defined in Section 3, given any base policy of the defender  $\mathbf{x}$ , for the new attacker pure strategy  $a_e$ :*

$$\begin{aligned} \mathbf{y}_2[a_e] &= 1 \quad ; \text{ if } U_a(\mathbf{x}, s_e) > U_a(\mathbf{x}, \neg a_e) \\ \mathbf{y}_2[a_e] &= 0 \quad ; \text{ if } U_a(\mathbf{x}, a_e) < U_a(\mathbf{x}, a_a) \quad \forall a_a \in \text{supp}\{\mathbf{y}_1\} \text{ and } U_a(\mathbf{x}, a_e) < U_a(\mathbf{x}, \mathbf{y}_1). \end{aligned}$$

*Proof.* The proof follows strong dominance definition [25].

In Theorem 1, we characterize two main conditions: (1) when the zero-day vulnerability generates a new attack path that strongly dominates every other existing attack path and (2) when it is being dominated by every path in terms of both pure and mixed strategy.

## 5 Zero-day mitigating strategies

The defender takes additional actions to mitigate the possible zero-day vulnerability exploits. The performed game-theoretic analysis identifies the impact of

each vulnerability, and the attacker's strategy for exploiting such vulnerability. The defender does not know which of the vulnerabilities will take place. However, to mitigate the zero-day attack, the defender allocates an additional honeypot. We consider four different strategies such as impact-based, capture-based, worst case mitigation, and critical point analysis to select the location of the new mitigating honeypot.

### 5.1 Impact-based mitigation (Alpha-mitigation)

First, we allocate based on the impact of each zero-day vulnerability. The impact measures the increase of the attacker reward due to each introduced vulnerability,  $e \in \mathcal{E}_0$ , where  $\mathcal{E}_0$  is the set of zero-day vulnerabilities. We allocate the new honeypot to combat the most impactful vulnerability such that,  $U_a(G_2(e))$  is the highest. The defender may allocate more honeypots following the same order of impact of each  $e \in \mathcal{E}_0$ . In this mitigating strategy, we assume no information is available to the defender about which vulnerability is introduced. In the next subsection, we consider the probability of each of these vulnerabilities. In Section 6, we shed more light on the formation of the set of zero-day vulnerabilities  $\mathcal{E}_0$  overcoming the possible explosion in its cardinality and applying several rules to exclude dominated elements that are obviously useless or infeasible to the attacker.

### 5.2 Capture-based mitigation (LP-mitigation)

In the previous strategy (i.e., Alpha-mitigation), the defender does not account for the probability that a zero-day vulnerability may occur. Let  $P(e)$  denote the probability that a vulnerability located at edge  $e \in \mathcal{E}_0$  exists. The impact of such vulnerability is denoted by  $i(e)$ , where the impact is the innovation in reward received by the attacker due to exploiting  $e$  on  $G_1 + \{e\}$  compared to the attacker's reward on  $G_1$ . Let  $J(x)$  denote the cost function for the defender as follows:

$$J(x) = \sum_{e \in \mathcal{E}_0} P(e) \cdot i(e) \cdot (1 - y(e) \cdot x(e)) \quad (4)$$

where  $y(e)$  is the probability that  $e \in \mathcal{E}_0$  is exploited during an attack, and  $x(e)$  is the unknown probability to assign honeypot at  $e$ .

The goal of the defender is to find  $x \in [0, 1]^{|\mathcal{E}_0|}$  that minimizes the cost function  $J(x)$ . This results in a linear program that can be solved efficiently. The outcome of this LP will pick the location  $e$ , of the highest impact and most likely to occur (i.e.,  $\arg\max_e P(e) \cdot i(e)$ ). However, since the defender may not know the probability of existence priorly, we consider a worst-case scenario. In other words, we assume that nature will play against the defender and try to minimize its reward. Hence, the defender mitigating strategy should be characterized in response to the selection of the nature that can be obtained by solving a max-min problem as discussed next.

**Worst-case mitigation (play against Nature):** After identifying the most impactful vulnerability location or set of vulnerabilities by doing graph analysis defender does not sure about which zero-day vulnerabilities the attacker is going to exploit. Therefore, we do game formulation to find defender mitigating strategies based on the available information of the high impactful locations.

The attacker chooses one vulnerability at a time to exploit and selects a possible attack path associated with that vulnerability. Hence, the attacker's action space,  $\mathcal{A}_n$ , is the set of all possible zero-day vulnerabilities, and defender action space,  $\mathcal{A}_{md}$ , is the set of all possible high impacted locations. This problem is formulated as an auxiliary game played between the defender and nature.

For each action profile played by both players  $(a_{md}, a_n) \in \mathcal{A}_{md} \times \mathcal{A}_n$ , the defender receives a reward  $R_{md}(a_{md}, a_n)$  after zero-day attack mitigation and the attacker reward is  $R_n(a_{md}, a_n)$ . When the attacker selects vulnerability and the defender selected high impact location, does not match the defender's mitigating reward simply comes from the defender expected reward based on which criteria we are following. When they match we just follow Eqn.(1) based on which attack path and honeypot allocation are selected by the attacker and defender respectively.

The defender reward is expressed as

$$R_d^{mitigate}(G_1) = \begin{cases} U_d(G_2) \text{ or } U_d(G_3) & i \neq j \\ R_d^{mitigate}(a_d^{mitigate}, a_n) & i = j \end{cases}$$

We consider a zero-sum game. Let  $\mathbf{x}_{mitigating}$  and  $\mathbf{y}_{nature}$  denote the mitigating strategies of defender, and nature mixed strategies, when the game is played on known-to-all graph,  $G_1$ . The defender expected reward in worst case mitigation play against nature is expressed as:

$$U_d^{mitigate}(G_1) = \mathbf{x}_{mitigating}^T R_d^{mitigate}(G_1) \mathbf{y}_{nature}$$

where  $R_d^{mitigate}(G_1)$  is the matrix of the game played on  $G_1$  and the nature(attacker) expected reward  $U_n(G_1) = -U_d^{mitigate}(G_1)$ .

This game has played over high impact locations in graph and gives defender mitigating strategies which location to focus for mitigation and nature mixed strategies what location attacker may choose to exploit. After having defender mitigating strategies and nature mixed strategies of the attacker, any mitigation approaches can run and eventually evaluate over it.

### 5.3 Critical point mitigation

Previously, we specified a honeypot to combat zero-day attacks in addition to the honeypots used via the defender's base deception policy. Now aim at modifying the base deception policy itself to combat the zero-day attack given the outcome of our analysis of the impact associated with each zero-day vulnerability. An insightful observation is that the defender tends to greedily deploy honeypots

in locations that are closer to the target nodes in the network. However, when the attacker chooses a different path to attack and reach the target node, this approach does not help. It is worth noting that, in defending against zero-day attacks, when the defender selects a location that protects high-degree nodes (this is captured in node values  $v(i)$ ) that belong to multiple attack paths while being far from target nodes, the defender successfully captures the attacker more often. Interestingly, high-impact locations align with high-degree locations to be protected more often following Nash equilibrium deception strategies.

These observations led us to conduct critical point mitigation to find overlapping locations in the graph. In critical point mitigation, we choose one of the high impacted locations which is also an overlapping location where we deploy mitigation. After having critical points we increase the cost of accessing these points, consequently re-run the game on increased cost locations to find updated defender strategies and align optimistic and pessimistic defender strategies based on the updated defender strategies.

In our critical point mitigation, we modify the base policy of the defender,  $\mathbf{x}_1$ , as we do not deploy additional honeypots. After identifying the most impactful vulnerability locations, we increase node values of the nodes most affected (i.e., neighbors) by such vulnerabilities. This shifts the defender’s attention to these locations and in turn, results in a modified base policy (which we refer to as critical point mitigation) that considers the significance of these nodes. We show that critical point mitigation increased the capture rate of attackers when tested in different settings such as increased number of honeypots, different vulnerable entry nodes, and target nodes.

## 6 Numerical Results

In this section, we present our numerical results to validate the proposed game-theoretic model. We evaluate our analysis of zero-day vulnerability and the proposed mitigating strategies. First, we present game results to identify impact of possible zero-day vulnerabilities for the optimistic and pessimistic defender. Second, we show the results of the proposed deception and mitigation strategies. Finally, we discuss our findings, limitations, and future directions of our current research.

### 6.1 Experiment:

The initial honeypot allocation strategy follows the Nash equilibrium of the game model (game 1). We formulate the problem as a zero-sum game, solve the game defined in Section 4 and find the Nash equilibrium in terms of the mixed strategies,  $x^*$  and  $y^*$ , for the defender and attacker strategies, respectively.

For the analysis of the potential impact of zero-day vulnerabilities, we consider a 20-node network topology with 22 edges shown in Fig. 2. The 20-node network topology shown in Fig. 2 represents an attack graph with multiple root

node (i.e., the entry node  $\mathcal{E} = \{0, 1, 2\}$ ). In this scenario we define the set of target nodes as three nodes,  $T = \{18, 19, 20\}$ .

To form the set of zero-day vulnerabilities  $\mathcal{E}_0$ , we analyze our 20 nodes network topology. If we consider all possible new edges in the network which is impractical as time complexity is  $n^2$  for  $n$  nodes network. Since the vulnerability analysis is independent between different elements in  $\mathcal{E}_0$ , it can run in parallel computing nodes to reduce the run-time. Moreover, some locations are practically infeasible or useless to the attacker. We implemented a set of rules that excluded useless edges and edges that do not benefit our analysis. For instance, we excluded edges leading to dead-end nodes and edges from nodes that are unreachable from any attack path. Also, direct edges between targets and entry nodes are dominant edges without further analysis.

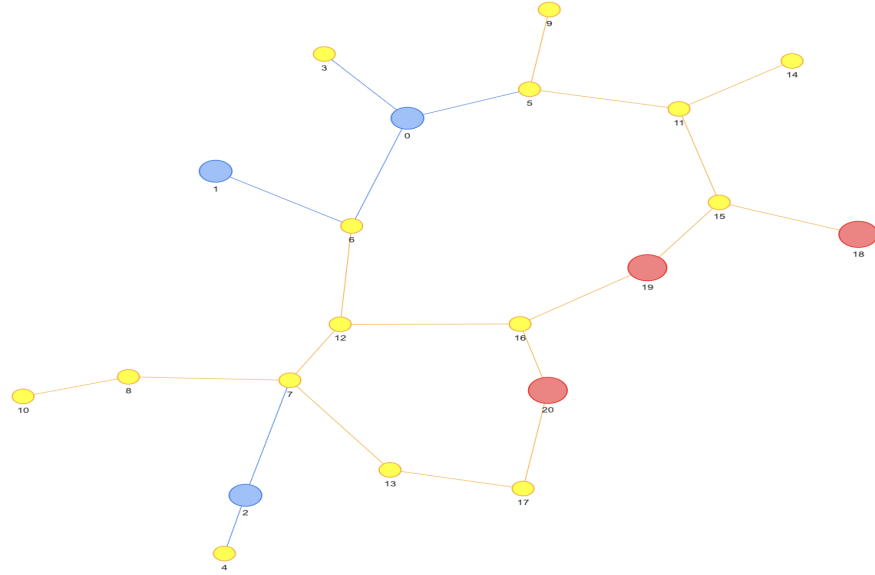


Fig. 2: Network topology of 20 nodes with blue red, and yellow color for entry, target and intermediate nodes respectively

We compare the Nash equilibrium strategy for honeypot allocation with existing attack policies to illustrate the effectiveness of our proposed cyber deception approach. For that, we observe defender and attacker gain under different conditions. Fig. 3 illustrates how defender reward change on several conditions including escape reward of honeypot and capture cost of the attacker.

Fig. 3a shows the defender reward against different cost values for escaping a single honeypot in the network over different attack policies. We also compare the Nash equilibrium reward against the greedy and random attacker. A greedy

attacker always selects nodes that have the highest values to attack regardless of their cost. A random attacker does not that informed about network that is unable to distinguish between possible attack paths and hence uniformly selects among available attack paths.

When the attacker deviates from equilibrium strategies  $\mathbf{y}^*$ , such as choosing greedy or random strategies, the defender reward tends to be higher or the same. For low Esc values, defender reward against greedy attacker higher compare to Nash equilibrium which less motivates an attacker to play rational strategies. On the other hand, high Esc values lure the attacker to take more risk in attacking valuable nodes, and as a consequence a gradual decrease in defender reward.

Fig. 3b shows a linearly increase in defender reward for different attacker policies. For high cap values, defender reward increases if the attacker deviates from rational strategies.

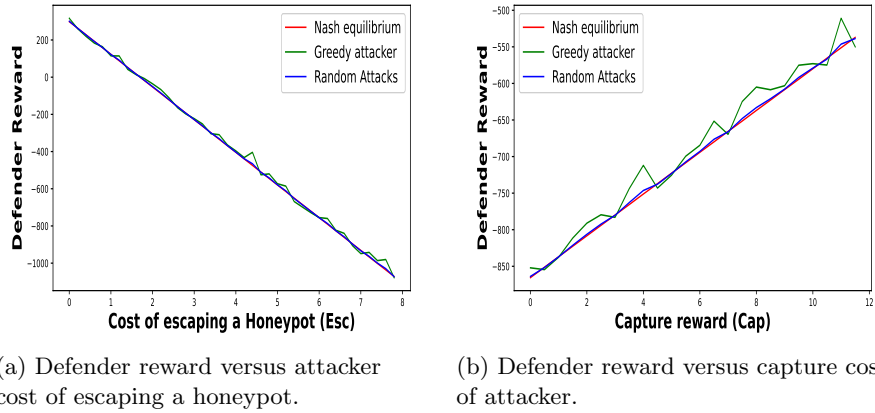
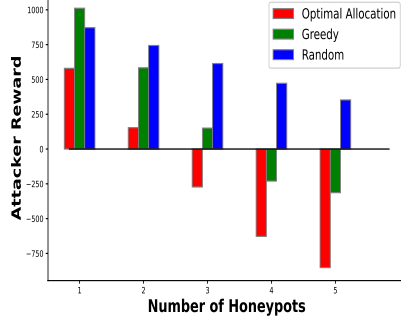


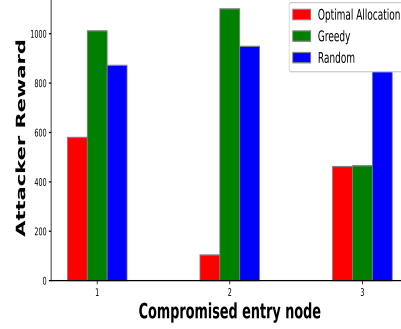
Fig. 3: Defender reward over different Cap and Esc values.

In addition, we also examine attacker's reward against different defender policies to deceive attacker and protect the network. Fig. 4 shows how attacker gain decreases as the number of honeypots increases and its dependence on the entry nodes.

In Fig. 4a we plot the average attack reward for different defender policies on honeypot budgets. We compare the performance of our optimal allocation with the greedy allocation that always allocates honeypots in the path of highest values nodes and random policies where defender uniformly select one node to protect rather than considering network topology analysis. The analysis of Fig. 4a illustrates that the optimal budget of honeypot in this network is three or more honeypots, as it dramatically reduces the effect of the attack. Also deploying 3 or more honeypots is very costly.



(a) Attacker reward versus the number of honeypots.



(b) Attacker reward versus compromised entry node.

Fig. 4: Attacker reward over different condition such as variation in honeypot number and compromised entry node.

In our 20-node network, three entry nodes are compromised at the start of the attack, so the attacker can attack using all possible existing paths in the network starting from any of the compromised entry nodes. We also plot the attacker's reward for a different number of compromised nodes in the network as shown in Fig. 4b over different defender policies. Here greedy and optimal allocation produces the same magnitude result.

Both Fig. 3 and Fig. 4 illustrate that deviating from Nash equilibrium and selecting some naive policies would not be optimal. Developing optimal mitigating strategies against a well-informed attacker critical for the defender to outperform naive deception policies such as random or greedy policies.

## 6.2 Impact of zero-days vulnerability

In our analysis, we find out high impact locations (zero-day vulnerabilities) for the 20-node network. We measure the impact of zero-day vulnerability. We consider two scenarios, first, when the attacker is certain that the defender does not know zero-day vulnerability. Second, the attacker is not sure whether the defender is aware of these zero-day vulnerabilities.

We also observe some zero-day vulnerabilities increase attacker reward massively and some remain the same compared to the naive defender. It is worth mentioning that some zero-day vulnerabilities also increase attacker reward in both cases, some increase only one scenario, not both depending on the reward function.

In Table. 1 we present attacker reward for different high-impact locations against the naive, optimistic, and pessimistic defender. Attacker reward against naive defender is the benchmark, attacker reward against pessimistic and optimistic defender defines how impactful that zero-days is.

Table 1: Attacker reward against naive defender, optimistic defender, pessimistic defender for top 10 edges

Edge	Naive defender	Optimistic defender	Pessimistic defender
(6, 7)	153.43	401.03	398.70
(5, 7)	153.43	374.65	370.26
(3, 7)	153.43	344.39	345.20
(16, 17)	153.43	326.52	326.52
(12, 13)	153.43	325.54	325.55
(15, 17)	153.43	323.60	323.60
(11, 13)	153.43	322.42	322.42
(11, 17)	153.43	315.72	315.71
(14, 17)	153.43	313.99	313.99
(12, 17)	153.43	307.24	307.23

**Attacker reward increases:** Based on our study, we highlight several reasons why certain zero-day vulnerabilities cause high damage to the defender compared to others. First, if a zero-day vulnerability creates multiple attack paths to any or all target nodes, that challenges the defender base-deception policy with limited honeypots in place and hence, causes significant damage. Second, zero-day vulnerabilities that are very close to any target nodes on the attack graph empower the attacker through a shortcut and enhance her reward. Also, a combination of the first two features leads to a significant loss for the defender.

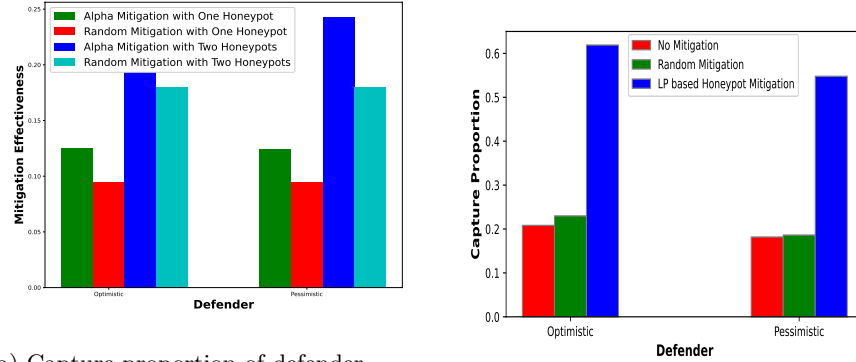
**Attacker reward remain same:** Interestingly, not all potential zero-day vulnerabilities cause significant damage to defender in terms of increasing attacker reward. Such zero-day vulnerabilities do not add useful actions to attacker action spaces that benefits the defender, consequently, the defender does not need to take mitigating measures for these types of vulnerabilities. Therefore, these observations benefit the defender to develop proactive defense focusing on most critical vulnerability locations.

### 6.3 Mitigation:

As detailed in Section 5, we proposed several approaches to develop mitigating strategies against zero-day attacks. In our approach, the defender goal is to thwart the attacker’s progress in the network by observing network information. We present numerical results to show the effectiveness of our mitigating approaches such as measuring proportion under various settings.

In Fig. 5 we show the proportion of attacker capture both for the optimistic and pessimistic defender with impact and linear programming-based mitigation. Fig. 5a presents the result of our impact-based mitigation. In our Alpha mitigation, we place honeypot based on the high-impact location whereas random strategies choose a location uniformly to place honeypot. Mitigation effectiveness





(a) Capture proportion of defender over zero-days vulnerabilities with single and multiple honeypot for both Alpha and random strategies.

(b) Capture proportion of defender with no, random, and LP-based honeypot mitigating strategies.

Fig. 5: Attacker capture proportion over different mitigating strategies of defender including Alpha or LP.

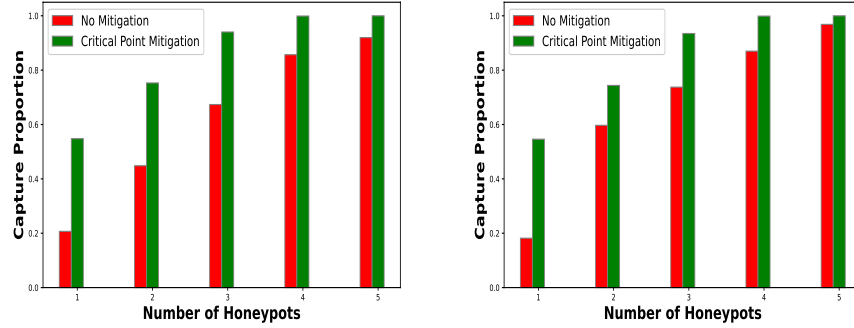
denotes the percentage of zero-day vulnerabilities defender mitigation (Alpha) prevents among all vulnerabilities. And capture proportion denotes the percentage of time an attacker is captured when exploiting a particular vulnerability.

Fig. 5a shows optimistic defender Alpha mitigation with one honeypot has higher mitigation effectiveness compared to random mitigation with one honeypot. On the other hand, the same strategies with 2 honeypots show a higher degree of deviation compared to the previous which denotes an increasing number of honeypots is useful but not a feasible solution.

Fig. 5b denotes attacker capture proportion over no, random, and LP-based honeypot mitigation both for the optimistic and pessimistic defender. No-mitigation and random mitigation are very close to each other meaning that randomly allocating honeypots will not bring any gain. After having the probability of allocating mitigating honeypot at different locations by solving a linear program explained in Section 5, we place honeypot on the corresponding location(s) and measure the proportion of capture the attacker increases.

In critical point mitigation, we modify the base policy without additional honeypot to take into account the criticality of the most impactful vulnerability. Fig. 6 denotes the capture proportion over the increased number of honeypots for the defender.

Fig. 6a shows capture rate increase for both no-mitigation and critical point mitigation strategies at different deception budgets (numbers of honeypots in the base policy). The difference between no and critical point mitigation reduces over the increased number of honeypots, which denotes that the number of honeypots more than three is not useful for mitigation. Fig. 6b shows the same result compare to Fig. 6a.



(a) Proportion of capture of attacker versus the number of honeypots by optimistic defender.

(b) Proportion of capture of attacker versus the number of honeypots by pessimistic defender.

Fig. 6: Attacker's capture proportion over different number of honeypots for defender on before and after critical node mitigation.

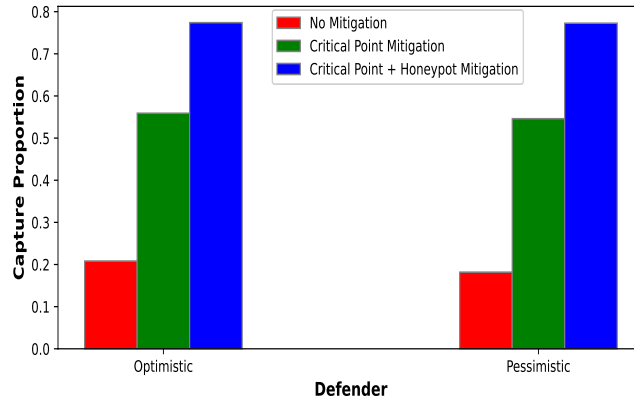


Fig. 7: Capture proportion on critical point based defender mitigation strategies

Fig. 7 demonstrates capture proportion on different defender mitigation strategies. Critical point mitigation and critical point mitigation with added honeypot outperform no mitigation. It is worth noting that adding one honeypot with critical point mitigation is useful as it increases the proportion of capturing the attacker.

## 7 Conclusion

In this paper, we proposed a security resource allocation problem for cyber deception against reconnaissance attacks. We proposed a novel framework to assess the effectiveness of cyber deception against zero-day attacks using an attack graph. We formulated this problem as a two-player game played on an attack graph with asymmetric information assuming that part of the attack graph is unknown to the defender. We identified the critical locations that may impact the defender payoff the most if specific nodes suffer a zero-day vulnerability. The proposed analysis is limited to considering a single vulnerability at a time, and focusing on the node location. Our future work will consider a set of vulnerabilities at a time which will follow the proposed analysis while significantly increasing the action space of the game model.

## References

1. Tarun Yadav and Arvind Mallari Rao. Technical aspects of cyber kill chain. In *International symposium on security in computing and communication*, pages 438–452. Springer, 2015.
2. Roei Schuster, Vitaly Shmatikov, and Eran Tromer. Beauty and the burst: Remote identification of encrypted video streams. In *26th USENIX Security Symposium (USENIX) Security 17*, pages 1357–1374, 2017.
3. Xinwen Fu, Bryan Graham, Dong Xuan, Riccardo Bettati, and Wei Zhao. Empirical and theoretical evaluation of active probing attacks and their countermeasures. In *International Workshop on Information Hiding*, pages 266–281. Springer, 2004.
4. Gunjan Bansal, Niteesh Kumar, Sukumar Nandi, and Santosh Biswas. Detection of NDP based attacks using MLD. In *Proceedings of the Fifth International Conference on Security of Information and Networks*, pages 163–167, 2012.
5. Hayreddin Çeker, Jun Zhuang, Shambhu Upadhyaya, Quang Duy La, and Boon-Hee Soong. Deception-based game theoretical approach to mitigate dos attacks. In *International conference on decision and game theory for security*, pages 18–38. Springer, 2016.
6. Quanyan Zhu and Stefan Rass. On multi-phase and multi-stage game-theoretic modeling of advanced persistent threats. *IEEE Access*, 6:13958–13971, 2018.
7. Ahmed H Anwar, Charles Kamhoua, and Nandi Leslie. A game-theoretic framework for dynamic cyber deception in Internet of Battlefield Things. In *Proc. 16th EAI Int’l Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 522–526, 2019.
8. Cliff Wang and Zhuo Lu. Cyber deception: Overview and the road ahead. *IEEE Security & Privacy*, 16(2):80–85, 2018.
9. Iyatiti Mokube and Michele Adams. Honeypots: concepts, approaches, and challenges. In *Proceedings of the 45th annual southeast regional conference*, pages 321–326, 2007.
10. Harjinder Singh Lallie, Kurt Debattista, and Jay Bal. A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review*, 35:100219, 2020.
11. Xinming Ou, Wayne F Boyer, and Miles A McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 336–345, 2006.

12. Leyla Bilge and Tudor Dumitras. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 833–844, 2012.
13. Mu Zhu, Ahmed H. Anwar, Zelin Wan, Jin-Hee Cho, Charles A. Kamhoua, and Munindar P. Singh. A survey of defensive deception: Approaches using game theory and machine learning. *IEEE Communications Surveys Tutorials*, 23(4):2460–2493, 2021.
14. Aaron Schlenker, Omkar Thakoor, Haifeng Xu, Fei Fang, Milind Tambe, and Phebe Vayanos. Game theoretic cyber deception to foil adversarial network reconnaissance. In *Adaptive Autonomous Secure Cyber Systems*, pages 183–204. Springer, 2020.
15. J. Pawlick and Q. Zhu. Deception by design: Evidence-based signaling games for network defense. *arXiv preprint arXiv:1503.05458*, 2015.
16. N. M. Fraser and K. W. Hipel. *Conflict Analysis: Models and Resolutions*. North-Holland, 1984.
17. R. Vane and P. E. Lehner. Using hypergames to select plans in adversarial environments. In *Proc. 1st Workshop on Game Theoretic and Decision Theoretic Agents*, pages 103–111, 1999.
18. K. Ferguson-Walter, S. Fugate, J. Mauger, and M. Major. Game theory for adaptive defensive cyber deception. In *Proc. 6th Annual Symp. on Hot Topics in the Science of Security*, page 4. ACM, 2019.
19. Jin-Hee Cho, Mu Zhu, and Munindar P. Singh. Modeling and analysis of deception games based on hypergame theory. In *Autonomous Cyber Deception*, pages 49–74. Springer, 2019.
20. Thanh Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 718–724, 2013.
21. Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. Stackelberg security games: Looking beyond a decade of success. *IJCAI*, 2018.
22. Peter Eder-Neuhauser, Tanja Zseby, Joachim Fabini, and Gernot Vormayr. Cyber attack models for smart grid environments. *Sustainable Energy, Grids and Networks*, 12:10–29, 2017.
23. Huthifh Al-Rushdan, Mohammad Shurman, Sharhabeel H Alnabelsi, and Qutaibah Althebyan. Zero-day attack detection and prevention in software-defined networks. In *2019 International Arab Conference on Information Technology (ACIT)*, pages 278–282. IEEE, 2019.
24. Paul Ammann, Duminda Wijesekera, and Saket Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 217–224, 2002.
25. Tamer Başar and Geert Jan Olsder. *Dynamic Noncooperative Game Theory*, volume 23. Siam, 1999.