Honeypot-Based Cyber Deception Against Malicious Reconnaissance via Hypergame Theory

Ahmed H. Anwar†, Mu Zhu⋆, Zeilin Wan‡, Jin-Hee Cho‡, Charles A. Kamhoua†, and Munindar P. Singh⋆
†US Army Research Lab, 2800 Powder Mill Rd, Adelphi, MD 20783
⋆Department of Computer Science, North Carolina State University, Raleigh, NC 27695
‡ Department of Computer Science, Virginia Tech, Falls Church, VA
a.h.anwar@knights.ucf.edu, mzhu5@ncsu.edu, {zelin, jicho}@vt.edu, charles.a.kamhoua.civ@army.mil, mpsingh@ncsu.edu
*Ahmed H. Anwar and Mu Zhu contributed equally in this research.

***Abstract*—Malicious reconnaissance is a critical step for attackers to collect sufficient network knowledge and choose valuable targets for intrusion. Defensive deception (DD) is an essential strategy against threats by misleading attackers' observations and beliefs. Honeypots are widely used for cyber deception that aims to confuse attackers and waste their resources and efforts. Defenders may use low-interaction honeypots or high-interaction honeypots. In this paper, we consider a hybrid honeypot system that balances the use of the two levels of honeypot complexity, where high-interaction honeypots are more capable of deceiving skilled attackers than low-interaction honeypots. We present a two-player hypergame model that characterizes how a defender should deploy low and high-interaction honeypots to defend the network against malicious reconnaissance activities. We model the tradeoff of each player and characterize their best strategies within a hypergame framework that considers the imperfect knowledge of each player toward their opponent. Finally, our numerical results validate the effectiveness of the proposed honeypot system.**

I. Introduction

Reconnaissance is a critical step that attackers perform to identify vulnerable and valuable targets. We consider *passive monitoring* and *active probing* of an enterprise network [2]. The passive monitoring attacks search services by monitoring the traffic between servers and clients and aims to use the collected intelligence to perform future attacks. Thus they are often invisible (i.e., hard to be detected by the system) but consume fewer resources to launch the attacks. On the other hand, active probing attacks aim to identify services by aggressively sending packets to hosts and analyzing their responses. They finally identify vulnerable nodes and their criticalities in a target system. Although active probing is more effective in terms of actual attack benefits, it consumes more resources and can be easily detected by the system due to its active scanning.

To detect and mitigate malicious reconnaissance, we consider defensive deception using honeypots in this work. Honeypots are commonly used to detect and mislead attackers and are mainly categorized into low-interaction honeypots (LHs) and high-interaction honeypots (HHs). LHs behave in a more-or-less fixed way while HHs carry out more realistic interactions [8]. LHs are easier to build and operate than HHs, while attackers can easily detect them. Although the attacker may not know an entire network topology or system information (e.g., active nodes' IP addresses), it may be able to distinguish the LHs from actual nodes by performing active probing.

In this work, we are interested in considering how the target system (i.e., a defender) can strategically identify an optimal defensive deception strategy (e.g., LHs or HHs) against such malicious reconnaissance where it aims to best protect the system from such attacks, given limited resources. Given resource constraints, we also would like to consider intelligent attackers who can strategically perform an optimal reconnaissance attack (e.g., passive or active scanning). Further, the attacker and the defender may perceive the game and their corresponding opponent's move differently under inherent uncertainty due to their partial observability. To effectively deal with such uncertainties in a game setting, we consider the so-called *hypergame theory* which enables players to choose their best action based on hypergame expected utilities (HEUs). The HEUs estimate players' expected utilities by considering uncertainty introduced by imperfect, partial observations of the game. To be specific, we make the following **key contributions** in this work:

- We develop a defensive deception framework based on a two-player hypergame for a setting in which an attacker applies active probing and passive monitoring while the defender deploys a mix of LHs and HHs to detect an attack. No prior work has studied game-theoretic defensive deception solutions for the attacker's game-theoretic strategic scanning attacks (i.e., passive or active).
- We identify the optimal defensive deception strategies considering each party's perceived uncertainty and hypergame expected utility (HEU). The two-player hypergame deals with uncertainty where the attacker and defender take actions based on their subjective perception of the game and the opponent's move.
- Via extensive simulation experiments, we prove that our game model can provide an optimal defensive deception strategy to effectively defend against reconnaissance attacks.

II. Background & Related Work

Passive and active reconnaissance attacks. Passive monitoring focuses on searching services by observing traffic between servers and clients. Encrypted web traffic can leak

information through packet length, timing, web flow size, and response delay [14]. Hence, passive scanning can detect active nodes, their services, supported protocols, operating systems (OSs), enterprise roles, and update schedule [11]. Since passive monitoring consumes fewer network resources than active probing, it can run on a long-term basis and detect active services running on transient hosts. The passive monitoring is generally invisible to the hosts running the services and can be difficult to detect by traditional security measures as it is non-intrusive. Moreover, firewall configurations can catch services that an active probing attack misses. However, a passive scanning attack can monitor only active services and those running on well-known ports and using protocol-specific decoders.

Active probing attacks can find services by sending packets to a host and analyzing its response, from which an attacker can learn the vulnerabilities and importance of a node [6]. Probing can be specific to a protocol or customized to an application. The active probing is fast and gives a complete report of all open and unprotected ports [1]. However, its aggressive intrusiveness can be easily detected. We will consider the attackers performing a mix of passive monitoring and active probing [1, 6] as attack strategies in a cyber game setting.

Game-theoretic defensive deception. Cybersecurity research has widely discussed game theoretic defensive deception [20]. Schlenker et al. [13] proposed a deception game where a defender chooses a deceptive response to reply to an attacker's observation while the attacker is unaware or aware of the deception. Pawlick and Zhu [12] introduced a signaling game to develop a honeypot-based defense system where an attacker can detect honeypots. However, the prior works [12, 13] did not consider honeypot systems with HHs and LHs when uncertainty in expected utilities is considered using hypergame theory.

Hypergame theory [5] has been used as an extensive game model to model different subjective views between players under uncertainty. Vane and Lehner [17] explored hypergames for decision-making in adversarial settings. Ferguson-Walter et al. [4] leveraged hypergames to quantify how a defensive deception signal can manipulate an attacker's beliefs. Cho et al. [3] and Wan et al. [19] discussed hypergame-based deception against advanced persistent threat (APT) attacks performing multiple attacks performed in the stages of the cyber kill chain. Kulkarni et al. [10] developed a zero-sum hypergame of incomplete information for evaluating the effectiveness of a proposed honeypot allocation. Unlike the prior works [3, 4, 10, 19], our work mainly focuses on developing a defensive deception game framework where the defender uses two types of honeypots (i.e., HHs and LHs) while the attacker performs either passive or active reconnaissance attacks.

III. System Model

A. Network Model

We consider a software-defined network (SDN)-based enterprise network consisting of servers, routers, and connected clients with a centralized entity. The SDN environment separates the network control and the data (e.g., packet forwarding) planes for higher flexibility, robust security/performance, and programmability [15]. Considering each node's network position and worth (i.e., importance), we group all involved nodes within three subsets, including *asset layer*, *internal layer*, and *core layer*, as described in Fig. 1. Each layer is detailed as:

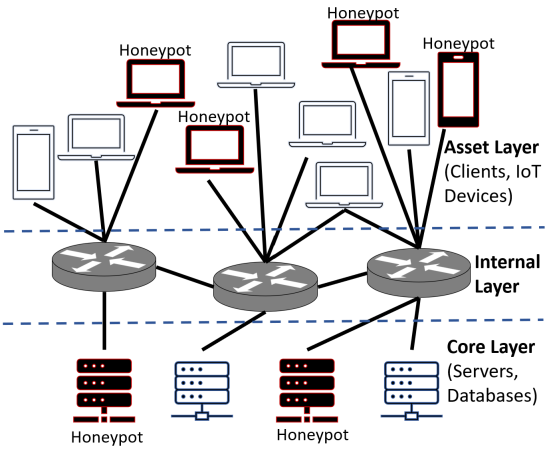


Fig. 1. Network Model: 3-layer network.

- *Asset Layer* (AL): This layer includes clients, such as Internet-of-Things (IoT) devices and laptops.
- *Internal Layer* (IL): This layer contains routers, switches, or other nodes between the asset layer and the core layers.
- *Core Layer* (CL): This layer includes high value nodes, such as database or other servers, which involve sensitive data.

Let $\mathcal{N} = N_{AL} \bigcup N_{IL} \bigcup N_{CL}$ denote the set of all nodes. We consider nodes in IL and CL. A node belonging to IL or CL is characterized by a *node worth*, denoted by $v(i)$, indicating the importance of the information node i has, which is susceptible to attacks. Depending on which layer node i belongs to, node i's importance, $v(i)$, is determined differently.

IL nodes (N_{IL}) play a role in network flow transfer. Node i is assigned with $W(i)$, representing node i's workload (e.g., the number of network flows) where higher $W(i)$ is more valuable. CL nodes (N_{CL}) are the most valuable assets in a given network. Their value is defined based on their type, $T(i)$, such as database or web server, where $T(i) = \{\text{type}_1, \text{type}_2, \text{type}_3, \ldots\}, \forall i \in N_{CL}$. Each node i's value, $v(i)$, is represented by these two values, $W(i)$ and $T(i)$, by:

$$v(i) = \begin{cases} W(i), & \text{if } i \in N_{IL} \\ T(i), & \text{if } i \in N_{CL} \end{cases} . \tag{1}$$

B. Defender Model

The defender uses various honeypots to collect the attackers' intelligence and protect tangible assets. Honeypots mimic fake vulnerable services to lure attackers into exploiting them. Virtual machines perform such a role with simple implementation; therefore, they are usually referred to as low-interaction honeypots, LHs. Moreover, adding LH nodes helps the defender hide the actual information (e.g., network topology or traffic data) that the attacker can collect using passive monitoring.

The attackers can easily detect LHs. Therefore, for a node to look like a CL node (e.g., database servers), the defender needs to deploy HHs specifically within the core layer.

The defender is aware of the network topology and the values of the connected nodes. The defender aims to balance the levels of honeypots (e.g., when to use LHs or HHs). LHs suffice to deceive passive attackers. However, they can be easily detected by active probing attackers. To prevent this, the defender can deploy HHs to interact and respond to the attacker's probes and requests. Relying solely on HHs is impractical and incurs high costs. Moreover, deploying HHs is unnecessary if the defender thwarts passive reconnaissance. Therefore, the defender should be able to strategically select an optimal defense depending on a given network situation and the attacker's strategies taken.

a) **Deploy LHs:** Low-interaction Honeypots (LH) [16] are virtual machines that reside on a host. More specifically, LHs do not represent a fully-featured operating system and usually cannot be thoroughly exploited. As a result, a LH is not well suited for capturing active probings or zero-day exploits and can be easily detected by experienced attackers.

To protect against passive monitoring, the defender decides a deception budget in terms of the number of honeypots to be allocated. Let $\mathcal{A}_d^\ell = [Low, Medium, High]$ be the strategy space for the defender, which represents three numbers of honeypots deployed in the asset layer and connected to IL. If the defender uses more LHs against passive reconnaissance it leads to a more secure internal and asset layer. The defender incurs a cost c_d associated with the implemented deception budget. However, to protect the core layer, the defender needs defense strategies that include HHs.

b) **Deploy HHs:** High-interaction Honeypots (HH) [18] represent real hosts attached to the network. Hence, we allow the defender to mimic more sophisticated systems that can interact with attackers and mislead them by false responses and fake information. Such a honeypot can monitor attackers and record their activities on the machine. The defender uses HHs to protect a particular device i. The strategy space containing the core layer nodes to be protected is denoted by $\mathcal{A}_d^h = N_{CL}$. An action $a_d \in \mathcal{A}_d^h$ defines a subset of the core layer nodes to be protected via HHs is defined as, $a_d = \{N_i | N_i \in N_{CL}\}$. HHs can successfully deceive attackers performing active probing. The associated defender cost, denoted by C_d, is much higher, i.e., $C_d > c_d$. The defender needs to balance between LHs and HH to be deployed to deceive the attacker. The defender action space is $\mathcal{A}_d = \mathcal{A}_d^\ell \times \mathcal{A}_d^h$.

C. Attacker Model

We consider an attacker in the reconnaissance stage, where the attacker targets an SDN-based enterprise network and aims to obtain the information, including the network topology and node worth, via passive monitoring and active probing.

a) **Passive Monitoring:** The attacker performs passive monitoring to collect information about the network, such as hosts' IP addresses, connectivity, and hosts' worth. Passive monitoring snoops data exchanged in a network without altering it. The such attack includes traffic monitoring to identify communication parties and functionalities, eavesdropping, and traffic analysis [7]. The attacks can help the attacker gain packet header and non-encrypted information and learn the network topology and the connection between different hosts.

TABLE I
KEY DESIGN PARAMETERS, MEANINGS, AND THEIR DEFAULT VALUES

Symbol	Meaning	Default
N_{AL}, N_{IL}, N_{CL}	The nodes in the asset layer, internal layer, and core layer, respectively	None
$W(i)$	The node worth of the node in IL	[1,5]
$T(i)$	The node worth of the node in CL	[6,10]
r	Information leaked by passive monitoring	None
R	Information leaked by active probing	None
c_a	Cost of passive monitoring	3
C_a	Cost of active probing	10
c_d	Cost of low-interaction honeypot	[2,4,6]
C_d	Cost of high-interaction honeypot	10
$v(i)$	Value of an arbitrary node i	Eq. 1
γ	Low-interaction honeypot detectability	0.5
Γ	High-interaction honeypot detectability	0.2

Assume an attacker must select one node in IL (e.g., routers or switches) to perform passive monitoring and obtain information about the hosts only if their network traffic passes through the selected node. As a result, a higher workload of the selected IL node means the attacker can obtain more information through the corresponding node. Recall $W(n)$ in Eq. 1 as the node worth of the nodes in N_{IL}. Because the attacker is unaware of the entire network topology at the beginning of the game, the attacker initially selects a random node as the target. After the attacker performs passive monitoring with a node, the attacker obtains the information corresponding to the node worth to the attacked node. We use $\mathcal{A}_a^p = N_{IL}$ to denote the action space when the attacker performs passive monitoring via node N_i and incurs cost c_a.

b) **Active Probing:** The attacker performs active probing by sending packets to a particular host to collect information regarding a specific host, such as finding open ports and OS types. Recall that the attacker's objective is to gather information from the core layer (e.g., database servers). Therefore, the active probing targets are the nodes in CL denoted by $\mathcal{A}_a^a = N_{CL}$, which is the attacker's action space under active probing with associated cost C_a where $C_a > c_a$. The attacker action space is $\mathcal{A}_a = \mathcal{A}_a^p \times \mathcal{A}_a^a$. A pure action $a_a \in \mathcal{A}_a$ is the set of nodes to be probed actively and passively. The attacker balances the two reconnaissance levels to avoid excessive attack costs and gain necessary network information.

In practice, an attacker may discover honeypots. Let γ and Γ denote the attacker's probability of discovering LHs and HHs, respectively. Table I summarizes game parameters. The attacker's reward depends on the deception strategy implemented by the defender, as explained next in Section III-D.

D. Utility Functions

Let u_d and u_a denote the utility functions for the defender and attacker, respectively. Consider a zero-sum game (i.e., u_d+

$u_a = 0$) where the players action profile is $(a_d, a_a) \in \mathcal{A}$.

If the attacker passively monitors a node in the IL, $a_a \in \mathcal{A}^p$, and the defender deploys LHs, $a_d \in \mathcal{A}_d^\ell$. **The defender's utility when taking LHs** is expressed by:

$$u_d(a_d, a_a) = \Big(\sum_{i \in a_a} -[r \cdot \hat{v}(i)] \Big) - c_d \cdot a_d + c_a, \quad (2)$$

where r is the information leaked via passive monitoring. Recall that $v(i) = W(i)$ is the value the node in IL. Let $W^{a_d}(i)$ denote the amount of fake traffic generated by LHs and passing through node i and $W(i)$ is the number of all traffic passing through node i. We assign $\hat{v}(i) = v(i) - W^{a_d}(i)$ as the value of node i under deception due to fake traffic. In other words, the node's value is decreased due to deceptive traffic that belongs to honeypots.

Similarly, if the attacker performs active probing, (i.e., $a_a \in \mathcal{A}_a^a$) , the defender needs to deploy HHs to mimic a certain type of node in CL (i.e., $a_d \in \mathcal{A}_d^h$). **The defender utility when taking HHs** is:

$$u_d(a_d, a_a) = \Big(\sum_{i \in a_a} R \cdot v(i) \mathbb{1}^{\Gamma}_{\{i \in a_d\}} \Big) - C_d \cdot |a_d| + C_a, \quad (3)$$

where R denotes the leaked information due to active probing and $v(i) = T(i)$ is the value of a node $i \in a_a$ targeted via active probing. $\mathbb{1}^{\Gamma}_{\{\cdot\}}$ is a special indicator function that equals Γ, if $i \in a_d$, where Γ is the probability of successful deception, and returns -1, otherwise. That is, if the attacked node is a honeypot, the defender receives a reward with the probability that the attacker is deceived successfully; the attacker gains a reward R, otherwise.

In some cases, the defender may implement a low level of deception against active probes, or high-level deception against passive monitoring. Thus, the utility will be a combination of Eqs. 2 and 3 as follows. In the first scenario, we have $a_d \in \mathcal{A}_d^\ell$ and $a_a \in \mathcal{A}_a^a$. As such, the defender utility is similar to Eq. 3, replacing Γ by γ as the probability of successful deception and replacing C_d by c_d as the cost of deception as given by:

$$u_d(a_d, a_a) = \Big(\sum_{i \in a_a} R \cdot v(i) \mathbb{1}^{\gamma}_{\{i \in a_d\}} \Big) - c_d \cdot a_d + C_a \quad (4)$$

In the second scenario, we have $a_d \in \mathcal{A}_d^h$ and $a_a \in \mathcal{A}_a^p$. The defender utility is:

$$u_d(a_d, a_a) = \Big(\sum_{i \in a_a} -[r \cdot \hat{v}(i)] \Big) - C_d \cdot |a_d| + c_a. \quad (5)$$

Both cases represent improper deception scenarios. However, the defender may fall into such scenarios due to a lack of information, which motivates our hypergame formulation as presented in the next section.

IV. Hypergame Formulation

Considering the attacker and defender models above and the uncertainty associated with each player, we present our hypergame model. The game is played repeatedly between two players, each facing an interesting tradeoff. The attacker decides whether to use active or passive probes. The defender balances the use of LHs and HHs to reduce the cost and provide effective deception. The cost for running a HH is greater than that of an LH. If the defender is certain about the type of reconnaissance the attacker uses, it can better optimize its deception strategies.

A. Subgames

We develop a hypergame formulation that captures the possible subgames played by the attacker. We refer to the defender as the row player. We consider three subgames (i.e., subgame 1, 2, and 3) where each subgame specifies the set of strategies to be played by both players defined below:

1) *Subgame 1*: The attacker solely relies on passive monitoring to perform reconnaissance. The defender conducts deception via LHs or HHs. The action space of this subgame is $\mathcal{A}_d^l \times \mathcal{A}_a^p$. The defender decides on the deception budget to invest in deception. The attacker selects a switch/router to monitor for a specific duration to gather information about the nodes' connectivity, and traffic flows passing through the targeted switch. Deception via LH is considered dominant for the defender in this subgame as HH incurs unnecessary costs.
2) *Subgame 2*: The attacker uses active probing to gather information about a node in the CL. The defender uses a HH to protect a certain node in the CL. The action space of this subgame is $\mathcal{A}_d^h \times \mathcal{A}_a^a$.
3) *Subgame 3*: This is a full game. Therefore, the action space of this subgame is $\mathcal{A}_d \times \mathcal{A}_a$.

B. Players' Beliefs

The belief vector is $\mathbf{P} = [P_1, P_2, P_3]$, where $P_3 = 1 - P_1 - P_2$. P_k is the true probability that subgame k will take place. These probabilities will be given based on the attacker's preference in performing different reconnaissance approaches (i.e., passive, active, or both). In practice, we assume that these probabilities are initially unknown to the defender. The defender improves his expected utility by learning the attacker's subgame preference P_k as in [9]. In this work, we assume that the defender knows the attacker's subgame preferences P_ks with uncertainty g_d as detailed next.

C. Players' Uncertainty

The defender is uncertain about which subgame the attacker will play and thus does not know true P_k's. However, as the game is repeated over time, the defender can learn the actual belief probability P_k for each subgame. One way is to assume that the defender observes the actual probability distribution of P_k's with probability $1 - g_d$ where g_d is the defender's perceived uncertainty toward what subgame the attacker will play. Hence, the defender is assumed to know the actual belief vector with $1 - g_d$. g_d is represented by a decay function in terms of the number of play rounds j as follows:

$$g_d(j) = \exp(-\frac{j}{\mu}), \quad (6)$$

where μ is a decay rate.

Let g_a denote the attacker perceived uncertainty about the cyber deception implemented by the defender. g_a exponentially decays in terms of Γ (i.e., HH detectability by the attacker), and γ (i.e., LH detectability by the attacker) by:

$$g_a(\gamma, \Gamma) = \exp(-\frac{\gamma + \Gamma}{2}). \quad (7)$$

D. Players' Mixed Strategies

If the attacker is playing *subgame 1* (i.e., performing passive monitoring), it is more efficient for the defender to implement deception using LHs. In fact, implementing HHs will come at an excessive cost for the defender (i.e., dominated strategies by $\mathcal{A}_d^\ell$ strategies). However, if the attacker performs active probing (i.e., *subgame 2*), the defender needs to use HHs and thus $\mathcal{A}_d^h$ will be the dominant strategies. Finally, if the attacker plays a combined reconnaissance (i.e., *subgame 3*), which is a full game), the defender needs to use both levels of deception. We consider a rational attacker that plays Nash equilibrium strategies within each subgame.

For any subgame $k \in \{1, 2, 3\}$, the defender (i.e., row player) mixed strategies, $DEF_k = [d_{k1}, \ldots, d_{km}]$, where $m = |\mathcal{A}_d|$ for the full game, such that $\sum_{i=1}^{m} d_{ki} = 1$. Similarly, the attacker (i.e., column player) mixed strategies at the k^{th} subgame is believed by the defender to be Att_k such that, $Att_k = [\mathrm{a}_{k1}, \ldots, \mathrm{a}_{kn}]$, where $n = |\mathcal{A}_a|$ for the full game, such that $\sum_{j=1}^{n} \mathrm{a}_{kj} = 1$.

According to the belief vector $\mathbf{P}$, the defender redefines its beliefs regarding the mixed strategies played by the attacker at different subgames. Specifically, let $\bar{\mathbf{a}} = [\bar{\mathrm{a}}_1, \ldots, \bar{\mathrm{a}}_n]$, where $\bar{\mathrm{a}}_j = \sum_{k=0}^{4} P_k \mathrm{a}_{kj}$ for $\forall j = 1, \ldots, n$.

E. Hypergame Expected Utility

Now we can calculate the hypergame expected utility (HEU) for the defender. Recall that the defender is uncertain about its beliefs regarding the subgame. Hence, the defender's HEU (DHEU) is a weighted combination of the EUs obtained under its belief and the EU under uncertainty, g_d. When the defender takes a_d, DHEU is obtained by:

$$DHEU(a_d, g_d) = (1 - g_d) \cdot EU\,(a_d; \bar{\mathbf{a}}) + g_D \cdot EU(a_d; \mathrm{a}_w), \quad (8)$$

where $EU\,(a_d; \bar{\mathbf{a}}) = \sum_{a_a \in \mathcal{A}_a} \bar{\mathrm{a}}(a_a) u_d(a_d, a_a)$. The second term represents the worst case expected utility received by the defender due to uncertainty when playing a strategy a_d against a damaging attack strategy w, such that $EU(a_d; \mathrm{a}_w) = n \cdot \mathrm{a}(a_w) \cdot u(a_d, a_w);\ a_w \in \mathcal{A}_a$.

The defender selects the defense strategy that maximizes his hypergame expected utility, $DHEU$. The attacker's mixed strategies, a_{kj}'s, are considered to represent Nash equilibrium mixed strategies within each subgame.

Similarly, the attacker's HEU (AHEU) for any strategy a_a is given by:

$$AHEU(a_a, g_a) = (1 - g_a) \cdot EU\left(a_a; \bar{\mathbf{d}}\right) + g_a \cdot EU(a_a; \mathrm{a}_w). \quad (9)$$

V. Evaluation by Simulation and Results

Simulation Settings. Recall that each node belongs to one of the three layers. In this simulation, we consider 100 nodes belonging to the asset layer, 70 nodes in the internal layer, and the core layer containing ten nodes. Nodes in the asset layer *AL* are assigned a value between $[1, 5]$ and between $[5, 10]$ for the node in the core layer *CL*. The IL nodes' values are calculated via Eq. 1. When the defender deploys low-interaction honeypots, each honeypot obtains an importance value of $[1, 5]$. If the defender deploys a high-interaction honeypot strategy to protect a particular node in CL, we create a new node as the honeypot and assign it the same importance value as the protected node. For attacker, we assign LH and HH detectabilities with $\gamma = 0.5$ and $\Gamma = 0.2$, respectively. Table I summarizes the notations of key design parameters, the corresponding meaning, and default values.

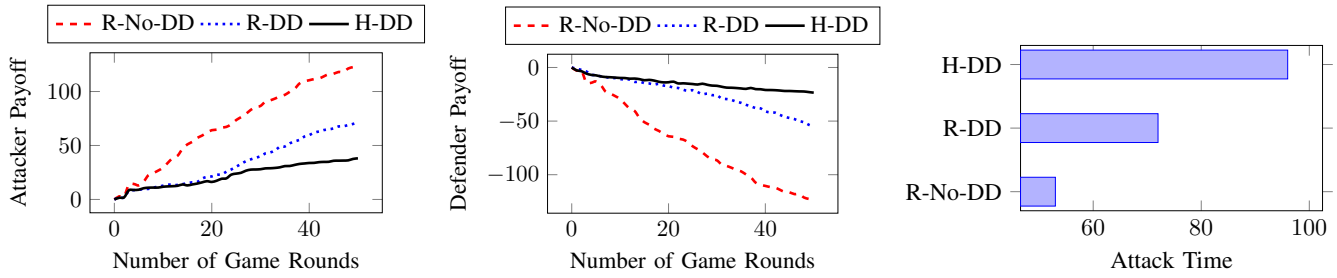
We use the following performance metrics:

- **Accumulated Payoff:** This is the accumulated values of an attacker's and defender's payoffs with respect to the number of game rounds, respectively.
- **Total Attack Time**: The attacker aims to collect all the values of the nodes in the CL. We use the number of game rounds for the attacker to complete its objective successfully to estimate this metric.

We compare the following policies for the defender and attacker to choose their best strategies:

- **Random with No DD (R-No-DD)**: Given no knowledge of the network, an attacker randomly selects its action when the defender does not use any defensive deception (DD).
- **Random with DD (R-DD)**: The defender and attacker randomly select their strategies when the defender uses DD.
- **Hypergame with DD (H-DD)**: The attacker and defender play a hypergame when the defender uses DD.

Fig. 2 (a) shows the attacker's accumulated payoffs when the attacker and defender play under the three policies (i.e., R-No-DD, R-DD, H-DD). Similarly, Fig. 2 (b) shows the defender's accumulated payoffs. Under R-No-DD (i.e., red dashed line), the attacker's payoff is significantly higher than under R-DD and H-DD (i.e., blue dotted and black solid lines). When the defender uses DD, the attacker's payoff decreases. Even if the defender randomly selects its strategies, the honeypots provide false information (e.g., deceptive network flow and fake nodes) and disturb the attacker's reconnaissance, as described in Eqs. 2 and 3). Moreover, playing H-DD decreases the attacker's payoff and increases the defender's payoff compared to playing R-DD. Recall that the uncertainty creates a discrepancy between the players' beliefs about the game. The defender uncertainty ($g_d(j)$) decreases over game rounds, which implies that the defender knows better about its opponents and chooses more suitable strategies against the attacker. As a result, the defender obtains the highest payoff under H-DD. Fig. 2 (c) shows the number of game rounds (i.e., total attack time) exhausted by the attacker to obtain all the values of the core layer nodes successfully. Without DD, the attacker needs 53 game rounds to harvest the



(a) Attacker's payoffs under various schemes with respect to the number of game rounds.

(b) Defender's payoffs under various schemes with respect to the number of game rounds.

(c) Total attack time with different schemes.

Fig. 2. Comparative performance analysis of defensive deception vs. non-defensive deception under R-No-DD, R-DD, and H-DD.

network information and the core layer nodes' worth. LHs and HHs provide false information to delay the attacker's passive monitoring and active probing (i.e., R-DD and H-DD). In addition, hypergame allows the defender to identify the best response to the attacker's reconnaissance actions (i.e., H-DD) where real-world uncertainties are better reflected on H-DD.

VI. Conclusion

This research proposed a hypergame-based hybrid honeypot system to defend against malicious reconnaissance. We proved that defensive deception could significantly delay the attacker's processes, leading to attack failures. We also observed that the hypergame considering uncertainty in practice allows the defender to select optimal strategies to mislead the attacker's reconnaissance by leveraging the inherent uncertainty that can mislead the attacker's perception.

ACKNOWLEDGMENT

This research was partly sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-19-2-0150. In addition, this research is also partly supported by the Army Research Office under Grant Contract Numbers W91NF-20-2-0140 and W911NF-17-1-0370. 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.

References

[1] G. Bansal, N. Kumar, S. Nandi, and S. Biswas, "Detection of NDP based attacks using MLD," in *Proceedings of the Fifth International Conference on Security of Information and Networks*, 2012, pp. 163–167.

[2] G. Bartlett, J. Heidemann, and C. Papadopoulos, "Understanding passive and active service discovery," in *Proc. 7th ACM SIGCOMM Conf. on Internet measurement*, 2007, pp. 57–70.

[3] J.-H. Cho, M. Zhu, and M. P. Singh, "Modeling and analysis of deception games based on hypergame theory," in *Autonomous Cyber Deception*. Springer, 2019, pp. 49–74.

[4] 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*. ACM, 2019, p. 4.

[5] N. M. Fraser and K. W. Hipel, *Conflict Analysis: Models and Resolutions*. North-Holland, 1984.

[6] X. Fu, B. Graham, D. Xuan, R. Bettati, and W. Zhao, "Empirical and theoretical evaluation of active probing attacks and their countermeasures," in *International Workshop on Information Hiding*. Springer, 2004, pp. 266–281.

[7] P. Goyal, S. Batra, and A. Singh, "A literature review of security attack in mobile ad-hoc networks," *International Journal of Computer Applications*, vol. 9, no. 12, pp. 11–15, 2010.

[8] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security: A research perspective," *ACM Computing Surveys*, vol. 51, no. 4, Jul. 2018.

[9] J. T. House and G. Cybenko, "Hypergame theory applied to cyber attack and defense," in *Proc. SPIE Conf. on Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IX*, vol. 766604, May. 2010.

[10] A. N. Kulkarni, J. F. H. Luo, C. A. Kamhoua, and N. N. Leslie, "Decoy allocation games on graphs with temporal logic objectives," in *Proc. Int'l Conf. on Decision and Game Theory for Security*. Springer, 2020.

[11] D. Montigny-Leboeuf and F. Massicotte, "Passive network discovery for real time situation awareness," Communications Research Centre Ottawa (Ontario), Tech. Rep., 2004.

[12] J. Pawlick and Q. Zhu, "Deception by design: Evidence-based signaling games for network defense," *arXiv preprint arXiv:1503.05458*, 2015.

[13] A. Schlenker, O. Thakoor, H. Xu, F. Fang, M. Tambe, and P. Vayanos, "Game theoretic cyber deception to foil adversarial network reconnaissance," in *Adaptive Autonomous Secure Cyber Systems*. Springer, 2020, pp. 183–204.

[14] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: Remote identification of encrypted video streams," in *26th USENIX Security Symposium (USENIX) Security 17)*, 2017, pp. 1357–1374.

[15] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in *IEEE SDN for Future Networks and Services (SDN4FNS)*. IEEE, 2013, pp. 1–7.

[16] C. Seifert, I. Welch, P. Komisarczuk *et al.*, "Honeyc-the low-interaction client honeypot," *Proc. 2007 NZCSRCS, Waikato University, Hamilton, New Zealand*, vol. 6, 2007.

[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*, 1999, pp. 103–111.

[18] G. Wagener, A. Dulaunoy, T. Engel *et al.*, "Self adaptive high interaction honeypots driven by game theory," in *Symp. on Self-Stabilizing Systems*. Springer, 2009, pp. 741–755.

[19] Z. Wan, J.-H. Cho, M. Zhu, A. H. Anwar, C. Kamhoua, and M. P. Singh, "Foureye: Defensive deception against advanced persistent threats via hypergame theory," *IEEE Transactions on Network and Service Management*, 2021.

[20] M. Zhu, A. H. Anwar, Z. Wan, J.-H. Cho, C. A. Kamhoua, and M. P. Singh, "A survey of defensive deception: Approaches using game theory and machine learning," *IEEE Communications Surveys Tutorials*, vol. 23, no. 4, pp. 2460–2493, 2021.