Cyber Deception using Honeypot Allocation and Diversity: A Game Theoretic Approach

Ahmed H. Anwar, and Charles A. Kamhoua
US Army Research Laboratory
Adelphi, Maryland 20756
Email: a.h.anwar@knights.ucf.edu, charles.kamhoua.civ@mil.gov

2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC) | 978-1-6654-3161-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCNC49033.2022.9700616

***Abstract*—Cyber deception has become the core of advanced enterprise-level defense systems. It is also being used for early detection by many experts. In this paper, we propose a novel approach for cyber deception using honeypot allocation and software diversity to enhance network security. The network defender chooses where to place the honeypots given a limited budget of resources. Also, we consider an interesting tradeoff between the level of software diversity to be implemented in the network and the operational cost incurred due to using different types of honeypots. To this end, we formulate a game-theoretic approach to characterize the honeypot allocation policy that protects the most valuable resources of the network. Moreover, we develop a game model between the two players to investigate the diversity tradeoff. Our results show that careful honeypot allocation is critical to protect high-value nodes and validate the proposed software-diversity approach.**

I. Introduction

Cyber deception has been playing an important role in advanced enterprise-level network defense systems [1] as modern networks are growing in both size and heterogeneity. Today's networks accommodate a wide range of complicated services with changing demands [2]. For instance, added devices such as wireless-enabled devices, sensors, and other IoT devices led to a world of large-scale dense networks. Security challenges are growing faster than ever as a direct consequence to the growth of network size and applications which makes network more vulnerable. Moreover, there is an added cost due to the diversity to secure and maintain new devices such as patching vulnerabilities. Such security issues are even more challenging in the military environment. Attacks targeting infrastructure and government entities illustrates the great need for developing more advanced defense systems capable of preventing well-orchestrated attacks. Moreover, the need for cyber resilience is most pressing for mission-critical systems [3], [4]. As information systems become ever more complex and the interdependence of these systems increases, defending our network becomes more and more critical. Unfortunately, it is not always possible to anticipate every type of component failure and cyber attack within large information systems and attempting to predict and protect against every conceivable failure and attack soon becomes exceedingly cumbersome and costly. For example, in 2019, the published strategy of the U.S. Department of the Army highlighted the role of military deception strategies for multi-domain operations [5], and network security.

In this paper, we propose a cyber deception approach based on honeypot allocation and software diversity to protect networks against adversaries capable of launching effective attacks. As attackers gather information about the networks ahead of starting any malicious activities [6], deception is of great significance to misrepresent the network state and information that could be collected during the reconnaissance stage. Moreover, attackers can obtain information about the structure and connectivity of the targeted network via insider help [7]. An adversary can map out the network using available scanning tools such as Nmap [8] or via more advanced network inference techniques [9]. The network administrator needs to protect the network resources. Cyber deception does not only help in defending the network against malicious activities, it can also assist IDS and detect intruders. Deceiving attackers via manipulating the network state by using honeypots and decoy devices disrupt the attackers' reconnaissance outcome. To this end, we develop a game theoretic framework to find the optimal deception (i.e., honeypot allocation) policy.

Although Cyber Deception approaches based on game-theoretic approaches are fairly recent, it is a growing topic in the literature. The developed models allow for modeling realistic scenarios and capture pivotal characteristics of network security in an attack/defense scenario. In such a scenario, the defender protects the network by deceiving the attacker using decoy devices via manipulating the network state, or by using deceptive signals, [10]–[12], [12]–[17]. The network admin can also camouflage nodes to hide critical networked devices [18]–[20]. In Schlenker et al. [21], [22], a Stackelberg game is used to increase the uncertainty of the adversary regarding the system and to disrupt the outcome of scanning tools.

Cyber deception is an under-studied topic in the literature, and the proposed approach provides a novel framework to tackle the tradeoffs associated with cyber deception accounting for the network characteristics and topological features in a game theoretic framework. Also, it consider a realistic threat model, we assume a that the defender is uncertain about the node that may be compromised first. Our contributions can be summarized as follows:

- We present a two-layer cyber deception scheme to protect the networks most valuable resources via capturing the network topological. Unlike prior work, the defender feasible actions are tied to an attack graph.
- We formulate a two-player game between the defender

and the attacker played on top of the attack graph and characterize the Nash equilibrium of the game. The defender places honeypot interfaces to machines, while the attacker selects an attack path trough the attack graph topology.
- We extend the game model to enhance deception via diversity, and study the tradeoff between efficient diversity level implemented in the network and the additional operational cost introduced by diversity.
- We provide numerical results to show the validity of the developed deceptive techniques in terms of the defender reward and show that the developed game-theoretic game is yielding a meaningful honeypot allocation policy.

The rest of the paper is organized as follows: In Section II we describe the system model. In Section III we formulate the game between the two players and characterize the solution concept. Then in Section IV we present the diversity game model. Finally, we present numerical results in Section IV-A and conclude the paper in Section IV-B.

II. System Model

When dealing with deception via honeypot/honeynet generation, the first question that usually arises is *"Where should we place the deceptive resources on the network topology ?"*. Although it might be intuitive to think that deception and decoy devices should be placed near and around the most valuable nodes, in many cases experienced attackers may not follow such a greedy approach. Advanced persistent attackers have shown patience and intelligence in evading naive allocation policies [23]. The answer to this question relied heavily on the importance of each location. Therefore, the network topology and the characteristics of each node is the key to differentiate between the nodes. Given the network topology, the network admin would need to take judicial actions to protect the network valuable resources against adversaries. However, placing honeypots and decoy devices has an associated cost such as the overhead it represents and the operational cost to configure the honeypots. The operational cost also depends on the type of the honeypot. For instance, high interaction honeypots will have a higher operational cost due to their configuration complexity. High-interactive honeypots are more difficult to be detected by the attacker. On the other hand, low-interactive honeypots could be detected by an experienced attacker while its operational cost is less [24]. For sake of conciseness, we assign a value for each node in the network to capture the significance of its functionality, features, and importance. While an attacker may dwell within a sophisticated honeypot this may not be related to an attacker's available resources. An attacker with fewer resources may have less interest in diving deep into a single system versus casting a wider net for easier targets. Therefore, honeypots are portrayed in the attack graph to lure attackers through easy-to-compromise vulnerabilities.

If the adversary is able to identify a honeypot, then the attacker will be able to identify all the allocated honeypots of the same type in the network (i.e., same software and configuration) [25]. Therefore, it is essential to diversity between the several honeypot types in the network and not rely only on the same software type all over the network. However, there is a an additional cost that is paid by the network defender for each new type used.

On the attacker side, the goal of the attacker aims at compromising the network nodes. A strategic attacker will also select the path between an entry node and a target node. The adversary expects that some of the nodes along the way are traps. The tradeoff that the attacker faces is to reach the target node and at the same time remain stealthy by avoiding the paths that may contain honeypots. According to the type of the software of the honeypot, the attacker may have the capability to identify it or not. For instance, if the attacker has limited resources and tools, he may be captured in one of the allocated honeypots.

We start by separating the two problems to initially solve the problem of honeypot allocation/evasion. Then we investigate the diversification between the different types of honeypot. Next, we introduce network model, then we formulate the problem of honeypot allocation and avoidance as a two-player game model between the defender and the attacker.

A. Network Model

Assume a network of $|\mathcal{N}|$ nodes, where $\mathcal{N}$ is the set of nodes of the network. These nodes are networked according to a binary adjacency matrix, $\mathbf{H}$, where any entry of the network $[\mathbf{H}_{u,v}] = 1$ if node v is connected to node u, and is equal to 0 otherwise, for every $u, v \in \mathcal{N}$. Assuming the network is a directed graph denoted by $G(\mathcal{N}, \mathbf{H})$. Such assumption fits a hierarchical network where there is a subset of nodes that containing the potential entry nodes. Any adversary starts his attack via pivoting through one of the entry nodes. Let $\chi \subset \mathcal{N}$ denote the subset of entry nodes. Moreover, there exist another subset of target nodes denoted by $\mathcal{T} \subset \mathcal{N}$, that contains the most valuable and potential targets for any adversary. The remaining set of nodes are considered intermediate nodes. After the attacker enters the network through one of the entry nodes in χ, it also needs to compromise intermediate nodes to find a way to reach and then compromise one of the target nodes in $\mathcal{T}$. Such networks resemble networks with a chain of command. It also represents an interesting scenario where the depth of the network can be captured. Adversaries are interested in reaching targets that belong practically to deeper layers of the network as shown in Figure. 1, where $\chi = \{1\}$ and $\mathcal{T} = \{5, 7\}$. The defender needs to decide where to place the honeypot(s), while the attacker chooses the path to reach one of the target nodes from node 1. Practically, the subset of entry nodes contains IoT devices, personal computers, and client server nodes, while target nodes are the network core nodes such as database servers.

B. Defender Model

The defender's goal is to protect the target nodes, $\mathcal{T}$ via selecting the location(s) of the honeypot(s). Let B denote

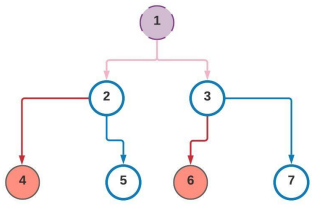

Fig. 1. A 7-node tree network topology with single point of entry and two target nodes (4,6).

the total number of allowed honeypots to be allocated (i.e., budget). In order to preserve the network topology, we assume that the honeypots will be placed as interfaces to each node. In other words, if to protect a certain node, the original node address will be hidden and its IP address will be switched to a honeypot that hijacks the location of the true node. Therefore, each honeypot will be modeled along the connecting edge to capture any attempts to reach a specific node if the defender decided to protect it via a honeypot. Therefore, to compromise any node, the attacker has to compromise its immediate parent node first. The defender needs to take into account the node values, and the total budget, and the possible paths that the attacker can follow before making his decision.

Hence, the defender action space per single honeypot contains all the set of edges. Let $\mathcal{E}$ be the set of edges and let $\mathcal{A}_d$ denote the defender actions space. Considering B honeypots, $\mathcal{A}_d = \{\mathbf{e} \in 2^{\mathcal{E}} \quad s.t \quad \mathbf{1}^T\mathbf{e} \leq B\}$. Where $\mathbf{1}^T\mathbf{e}$ is the inner product of the vectors. $\mathbf{e} \in \mathcal{E}$ is a binary vector of length $|\mathcal{E}|$, with an entry is set to 1 if a honeypot is allocated on this edge, and is set to 0, otherwise. The inner product condition between $\mathbf{e}$ and a vector of all ones, $\mathbf{1}^T$, ensures that the total number of allocated honeypots does not exceed the defender's budget. Therefore, a feasible deception action $a_d \in \mathcal{A}_d$ is binary vector of length $|E|$ such that, $||a_d||_1 \leq B$ To avoid trivial deception scenarios, we assume a limited budget that does not allow for covering the entire network via decoys, and hence strategic allocation is significant to capture the adversary and protect critical and high-value nodes.

The payoff function for the defender is designed to account for the number of successful and wasteful honeypots. If the attacker paths through an edge where a honeypot has been placed, this represents a successful allocation for the defender, otherwise it is considered a loss as the attacker is able to avoid the honeypot. Each successful event for protecting a specific node is multiplied by the node value. Similarly, a loss is penalized by the node value that was successfully compromised. Since the reward of the defender depends on the action taken by the attacker, we model the attacker model next.

C. Attacker Model

The attacker's goal is to reach a target node, which the defender cannot precisely specify. The defender tries to select the path from the network entry node, to the target node avoiding paths that are likely to have honeypots on the way. Let $\mathcal{P}$ be a set that contains all feasible paths between every pair of nodes in the network. For instance, a path between nodes 1 and 6 for the network topology shown in Fig. 1 is $P_{\{1,6\}} = \{1, 3, 6\}$, containing the 2 edges of $(1, 3)$ and $(3, 6)$. Note that, other network topologies may allow for multiple paths between the same two nodes in $\mathcal{P}$. Starting from an entry node v_0 where the attacker initially compromises the network, the attacker's action space $\mathcal{A}_a$ is a subset of $\mathcal{P}$ where v_0 is the starting node. An action played by the attacker $a_a \in \mathcal{A}_a$ is a vector containing all the nodes along the the attack path leading one of the target nodes and starting from v_0.

The attacker selects a full path between the entry node and one of the target nodes (leaf node). The attacker gains along the way per each successful move that allows him to avoid honeypots and compromise a new node. Next, we formulate the game model between the two players.

III. Honeypot Allocation Game Formulation

We formulate a two-player zerosum game between the network defender and the attacker. The end goal of his game is to provide the network admin with an allocation policy to be implemented initially for a given network topology and the node values.

A two-player game defined as a tuple $\Gamma(\mathcal{L}, \mathcal{A}, \mathcal{R})$, where:

- $\mathcal{L} = \{Defender, Attacker\}$ is the set of players. We use the subscripts 'd' and 'a' to refer to the defender and the attacker, respectively.
- $\mathcal{A} = \{\mathcal{A}_d \times \mathcal{A}_a\}$ is the game action space, which is the product of the action space of the two players. The defender action space $\mathcal{A}_d = \{\mathbf{e} \in 2^{\mathcal{E}} \quad s.t \quad \mathbf{1}^T\mathbf{e} \leq B\}$ and the attacker action space is $\mathcal{A}_a = \{\mathcal{P}(v_0)\}$ where, $v_0 \in \chi$ is the attack starting node.
- $\mathcal{R} = \{R_d, R_a\}$ denotes the game reward set such that $\mathcal{R} : \mathcal{A}_d \times \mathcal{A}_a \rightarrow R$, where the zerosum game, $R_d + R_a = 0$. The reward function is formulated below.

For the formulated game Γ, zerosum reward functions, captures the tight conflict between the network defender and the adversary. As such, the defender focus on the nodes of great interest to be protected via deception resources. The reward of the defender is strictly limited to successful allocation strategies that lure the attacker to attack the honeypot. Game-theoretic models that focus on resource allocations to prevent attacks in the literature (a.k.a interdiction-games) are largely modeled as Stackelberg games, in which the defender is the leader and the adversary follows [26]. However, in our game model, the defender sticks to deceptive actions, these actions are not observable to the attacker. Similarly, the defender is assumed to have no information regarding the exact location of the attacker in the network. Therefore, To address this incomplete information setting, we assume the two players to play simultaneously and each player is solving a minmax optimization problem.

A. Reward Function:

Let ω_v denote the value of any node v excluding the entry nodes, $v \in \mathcal{N} \setminus \chi$. Let (a_d, a_a) be an action profile taken by

the two players and $\mathcal{S}$, be the set of successfully compromised nodes. A compromised node $u \in \mathcal{S}(a_d, a_a)$ if $a_d(u) = 0$ while $u \in a_a$ (i.e., u belongs to the attack path specified by a_a). However, $\mathcal{U}$ represents the subset of nodes that were attacked through an edge that has been protected via a honeypot. In other words, a node $v \in \mathcal{U}$ if $a_d(v) = 1$ and $v \in a_a$. Let $\mathbf{u}$ be a binary vector of length $|\mathcal{N}|$, entries corresponding to nodes in $\mathcal{U}$ are set to 1, and 0 otherwise. Similarly, $\mathbf{s}$ is a binary vector of the same length as $\mathbf{u}$, with entries equal to 1 for nodes in the set $\mathcal{S}$. The defender reward can be defined as follows,

$$R_d(a_d, a_a) = \mathbf{u}^T \cdot \omega - \mathbf{s}^T \cdot \omega \tag{1}$$

Where $\omega \in R^{|\mathcal{N}|}$ is a vector of node values. The defender's goal is to maximize the number of protected nodes that the attacker is attempting to compromise at the same time while reducing the total number of successfully compromised nodes. The attacker on the other side attempts to minimize the same amount defined in equation (1). This implies that the attacker is interested in compromising nodes with high values (i.e., target nodes) while the attacker is interested in protecting valuable nodes as well. Note that to avoid trivial outcomes, both players are competing against each other on the same group of high-value nodes, otherwise the defender will protect nodes equally, and the attacker may also maximize his reward by attacking non-target nodes.

Each player has to maximize his reward function. However, the reward function of each player depends not only on his actions but on the action taken by his opponent as well. Therefore, when making his/her decision, a player needs to consider all actions that can be taken by his/her opponent. The most adopted solution concept used in game theory is the Nash equilibrium. An action profile (a_d^*, a_a^*) means that no play will have an incentive to unilaterally deviate from this equilibrium action and play another action [27].

B. Game Solution

Nash equilibrium may not exist in terms of pure actions $a_d \in \mathcal{A}_d$ and $a_a \in \mathcal{A}_a$ in each game. Interestingly, we can show that for the most simple non-trivial case such as in a network of 3 nodes, the game goes not admit a Nash equilibrium in pure strategies. To show that, assume a simple network of three nodes a, b, and c, and two edges connecting the root node a to the other two nodes, $e_{(a,b)}$ and $e_{(a,c)}$. The defender can either choose to protect b or c by allocating deceptive honeypot. The attacker, on the other hand, can either attack b or c. Assuming that b and c have the same exact value, according to the reward function defined in 1 the defender/attacker payoff matrix is as,

$$R = \begin{bmatrix} (1,-1) & (-1,1) \\ (-1,1) & (1,-1) \end{bmatrix} \tag{2}$$

and the game resembles a penny-matching game matrix. Clearly, the game for this example does not admit a Nash equilibrium in pure strategies, and the best strategy for both players is to alternate between their two actions with probability $(1/2, 1/2)$, i.e., play a mixed strategy rather than a pure one. In addition, a mixed strategy is well-fitted for deceptive games making the deceptive actions difficult to be learnt or predicted by the opponent.

For the formulated game, Γ, it is a finite, hence it admits at least one equilibrium point in mixed strategies.

A mixed strategy, $\mathbf{x}$, is a probability vector distributed over the defender action space, $\mathcal{A}_d$. Hence, $0 \leq x_i \leq 1$ for $i = 1, ..., |\mathcal{A}_d|$, is probability of playing a specific action $a_d \in \mathcal{A}_d$, On the attacker side, let $\mathbf{y}$ be a probability distribution vector over $\mathcal{A}_a$. A point point ($\mathbf{x}^*$,$\mathbf{y}^*$) is an equilibrium point if it satisfies the following condition, $\mathbf{x}^{*T} R_d \mathbf{y}^* \geq \mathbf{x}^{*T} R_d \mathbf{y}$ for any mixed strategy, $\mathbf{y}$, played by the attacker. Similarly, $\mathbf{x}^{*T} R_d \mathbf{y}^* \geq \mathbf{x}^T R_d \mathbf{y}^*$ for any mixed strategy, $\mathbf{x}$, played by the defender.

Theorem 1: For the finite game, Γ there exists at least one point ($\mathbf{x}^*$,$\mathbf{y}^*$) of mixed equilibrium.

The proof follows Nash's theory in [27], [28] directly.

For each possible joint action a_d and a_a we calculate the value of the reward function $R_d(a_d, a_a)$ formulating a matrix game A. We now readily solve the game for both players. Finding $\mathbf{x}^*$,$\mathbf{y}^*$ for finite games is requires each player to solve a linear program (LP) as shown below. We start by formulating the LP for the defender,

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}^T A \mathbf{y} \\ \text{subject to} \quad & \sum_{i=1}^{|\mathcal{A}_d|} R_d(a_d^i, a_a^j) x_i \geq \mathbf{x}^T A \mathbf{y} \ \forall j = 1, .., |\mathcal{A}_a| \\ & \sum_{i=1}^{|\mathcal{A}_d|} x(a_d^i) = 1, \\ & \mathbf{x} \geq 0. \end{aligned} \tag{3}$$

The objective is to maximize the expected reward under the defender mixed strategy, $\mathbf{x}$. The first condition ensures that the played mixed strategy x_i for each action a_d is the best response to the action played by the attacker a_a. This condition represents a set of conditions as the defender needs to consider all the $|\mathcal{A}_a|$ possible attack actions. The remaining two conditions ensure that $\mathbf{x}$ is a valid probability vector.

Similarly, we formulate the LP for the attacker that attempts to minimize the same expected payoff via ensuring that $\mathbf{y}$ is the best response to all the actions of the defender,

$$\begin{aligned} \underset{\mathbf{y}}{\text{minimize}} \quad & \mathbf{x}^T A \mathbf{y} \\ \text{subject to} \quad & \sum_{j=1}^{|\mathcal{A}_a|} R_a(a_d^i, a_a^j) y_j \geq \mathbf{x}^T A \mathbf{y} \ \forall i = 1, .., |\mathcal{A}_d| \\ & \sum_{j=1}^{|\mathcal{A}_a|} y(a_a^j) = 1, \\ & \mathbf{y} \geq 0. \end{aligned} \tag{4}$$

The two LPs defined above in (3) and (4) yield $\mathbf{x}^*$ and $\mathbf{y}^*$.

C. Defender Uncertainty about Entry Node

The game is well-defined only from the attacker's point of view since the defender does not know the node at which the game is starting. Only the attacker successfully compromises the system from a specific entry node v_0. We assume that the defender is uncertain about which entry node is eventually compromised. However, we relax this uncertainty condition assuming that the defender knows the subset of nodes that are potential entry nodes as represented by the subset χ. Moreover, we assume that the defender apriori information and records provides enough statistics about the probability that each node would be compromised for each node in χ. In other words, the defender has the estimates that node $\nu \in \chi$ will be compromised in probability. Let $\mathbf{z}$ be a probability vector distributed over χ.

The defender uses this available information, $\mathbf{z}$, to compute the optimal mixed strategy. Let $\mathbf{x}^{*\nu}$ be the Nash equilibrium mixed strategy obtained if the attack started at node $\nu \in \chi$. The defender solves a collection of $|\chi|$ games and finds the mixed strategies of each game, $\mathbf{x}^*_\nu$ and $\mathbf{y}^*_\nu$ to calculate the game *value* i.e., $\mathbf{x}^{*T}_\nu A \mathbf{y}^*_\nu$ and let Q_ν denote the value of each game as the defender constructs a value for each possible game ν. Let $V = \sum_{\nu \in \chi} \lambda_\nu Q_\nu z_\nu$ be the probability weighted sum of the different game values. The defender solves the following optimization problem to find the probability of playing each mixed strategy $\mathbf{x}^*_\nu$, where $\Lambda = \left[\lambda_1, ..., \lambda_{|\chi|}\right]$.

$$\begin{aligned} \underset{\mathbf{\Lambda}}{\text{maximize}} \quad & V \\ \text{subject to} \quad & \sum_{\nu \in \chi} \lambda_\nu = 1, \\ & \mathbf{\Lambda} \geq 0. \end{aligned} \tag{5}$$

After computing Λ^*, the defender finds the final adjusted mixed strategies x^*_ν.

In the next section, we enhance our defense approach by diversifying between the different types of honeypots. Due to its added operational cost, the defender should find the optimal degree of diversity in the network.

IV. Software Diversity

To ensure the effectiveness and sustainability of the proposed honeypot allocation in large networks, it is important to not rely on a single type of honeypot. Diversifying between different honeypot types ensures that if one type is detected or compromised by the attacker, other honeypot types remain covered. Otherwise, our deception approach may suffer a single point of failure.

On the attacker side, any attacker is equipped with a toolbox of tools to exploit vulnerabilities of different nodes in the network. Hence, diversity can be a double edged weapon in terms of security. In one hand, for an unskilled attacker with limited toolbox, diversity will enhance the system security as the attacker may have few exploits that will not fit to compromise the diversified network nodes. On the other hand, diversity may increase the set of vulnerabilities in the network which maybe in the favor of a skillful attacker. In either case, the attacker needs to spend more effort and spend time with experimenting different tools and techniques for the diversified network. However, in a mono-cultured network the attacker may reuse the same exploits for each node.

Diversity adds to the network operational cost incurred by the network admin. It also enhances network security and distracts the attacker and blocks its way. For instance, an attacker may not have the necessary tools to exploit all vulnerabilities in all honeypots. Therefore, diversity adds another layer of deception to protect the network.

Additionally, diversity introduces a new interesting tradeoff for the network defender, as implementing a more diversified network enhances the security level of the network. However, it adds to the network operational cost in terms of implementation, maintenance, patching, and configurations. On the other side, the attacker faces a new tradeoff in terms of the amount of effort to be spent in compromising the nodes since more effort will result in a stronger capability of compromising the different nodes. To study these tradeoffs, we formulate a game model between the two players. In this context, we separate the decision of the level of diversity from the decision of the allocation for two main reasons. First, the two games consider different aspects of the network. For example, the network connectivity and the criticality of each node both play an important role in deciding where to place honeypots, while it is more important to consider the cost incurred per compromise and the implementation cost associated with each new honeypot before deciding on the level of diversity as explained next. To avoid redundancy, we define a game $\beta(\mathcal{L}, \mathcal{A}, \mathcal{R})$, focusing more on defining the action spaces and payoff functions for both players. The solution concept follows the same as the solution of Γ.

For each honeypot $h \in \mathcal{H}$, there is running a software type $s \in SW$. A software type models the operating system, honeypot type, and application. Concisely, a matrix NSW stores all honeypot software assignment $|\mathcal{N}| \times |SW|$. Taking uncertainty into account, each software may have a subset of vulnerabilities. Let SWV be a $|SW| \times |\mathcal{V}|$ software-vulnerability matrix, where $0 \leq SWV[i,j] \leq 1$, is the probability that s_i has vulnerability v_j.

The attacker attempts to use a toolbox of exploits to compromise the different nodes. Each exploit can be used to compromise one or more vulnerabilities. Let $\mathcal{B}$ denote the set of all exploits and let $\mathbf{P}$ be the matrix that specifies which subset of vulnerabilities can be exploited such that $\mathbf{P}[i,j] = 1$ if exploit i can fit to compromise the j^{th} vulnerability, and is set to zero, otherwise. The attacker maximizes his/her payoff via maximizing the number of compromised nodes in the network. This can be achieved by using more exploits per attack. However, there is a cost associated with each attack denoted by $C_a(a_a)$. The attacker action space $\mathcal{A}_a$, is the set of all possible exploits $\mathcal{B}$. The defender, on the other side, provides more diversified vulnerabilities by using a larger number of software types from the set SW to be implemented in the network. In fact, $\mathcal{A}_d = SW$. Similarly, there is an associated cost with each action a_d denoted by $C_d(a_d)$. For

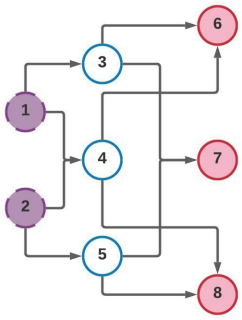

Fig. 2. An 8-node network topology with two entry nodes {1,2} and three target nodes {6,7,8}.

maximum diversity, selected software types are allocated such that neighboring honeypots will be of different software types. This can easily be accomplished by a graph coloring algorithm [29]. We have used a greedy algorithm for as the graph coloring problem is known to NP-complete.

The reward functions for each player is different as the cost metric per action for each player is different. Hence, the defined game β is a non-zerosum two player game. Let $k(a)$ be the total number of successfully compromised nodes due to a joint action $a = (a_d, a_a)$. The defender reward function can be expressed as:

$$R_d(a_d, a_a) = -K - C_d(a_d), \tag{6}$$

and the attacker reward function is expressed as:

$$R_a(a_d, a_a) = K - C_a(a_a). \tag{7}$$

The game finite game $\beta(\mathcal{L}, \mathcal{A}, \mathcal{R})$ admits a Nash equilibrium in terms of mixed strategies that can be found following the formulations of LPs as in (3) and (4).

A. Numerical Results

We now present numerical results that validate the proposed game model. First, we consider an 8-node network as shown in Figure. 2. The network considers asymmetric node value vector set as follows: non-target nodes such as nodes {3,4,5} each has a value of 10. Target nodes are set to have higher values where node {6,7} each has a value of 100, and node {8} is assigned a value of 200.

The formulated game Γ defined in Section III, we solve the game using the Fictitious Play algorithm [30]. The algorithm solves the game through iterative self-play against sampling opponent actions. For such an 8-node network with a single honeypot, the defender has 10 possible locations. The attacker on the other side chooses between 4 different attack paths connected the entry node to all the nodes in the subset of target nodes $\mathcal{T}$.

As shown in Figure. 3 and Figure. 4 both players converge to equilibrium in mixed strategies. When closely look into the obtained mixed strategy for the defender, we find that only 4 strategies are played with nonzero probability. Obviously, action a_d^6 and a_d^{10} are played with higher probability of 0.33,

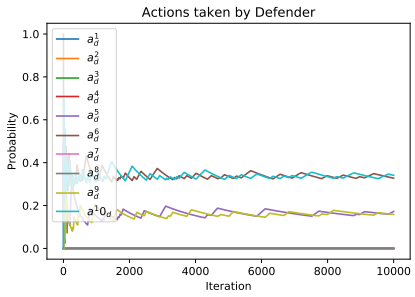

Fig. 3. Defender 10 strategies played probability converge to the Nash equilibrium mixed strategy

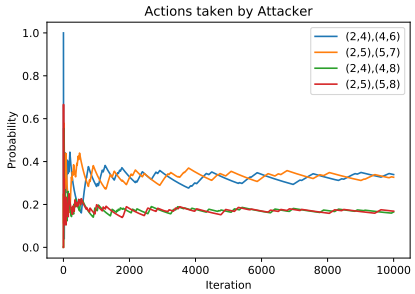

Fig. 4. Attacker 4 strategies played probability converge to the Nash equilibrium mixed strategy

and actions a_d^5 and a_d^9 are played with less probability of 0.166. The two policies, a_d^6 and a_d^{10}, place the honeypot on edges leading to the most valuable node (i.e., node {8}). The other actions a_d^5 and a_d^9 places the honeypot on edges leading to target node {6} and {7}, respectively. Such equilibrium policies actually made it more rewarding for the attacker to select paths leading to node {8} with lower probabilities as shown in Figure. 4. Nash equilibrium's attack policy suggests that the most valuable node is more likely to be protected, hence it decides to attack a rewarding path with lower probability. and instead attack paths leading to nodes {6} and {7} more often.

In order to show the effectiveness of the developed diversity approach, we show for a randomly generated 20-node network, in Figure. 5 the defender and the attacker rewards at different cost values per each software used by the defender at Nash equilibrium. The defender reward is affected adversely as the cost increases. The defender chooses to implement fewer software types to avoid the high cost C_d. The attacker on the other side shows an increase in terms of its reward as the defender reduces the level of diversity in the network and uses a mono-cultured deception scheme.

B. Conclusion

In this paper, we proposed a two-layer cyber deception approach in a game-theoretic framework. First, we propose a honeypot and decoy allocation scheme and optimize the locations of the honeypots considering the network topology as well as the features and importance of the nodes. We formulate

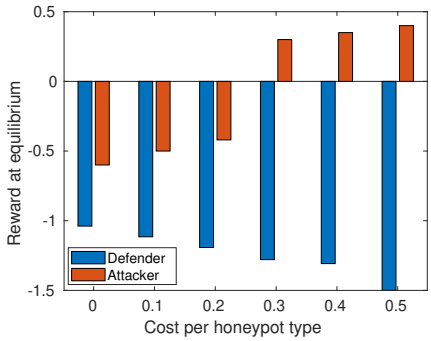

Fig. 5. Players reward at equilibrium as diversity cost increases.

a two-player zerosum game between the network defender and the attacker and characterize the Nash equilibrium in mixed strategies. The game is solved for numerical evaluation using a fictitious play algorithm. To enhance the security level and avoid a single point of failure, we investigate the effectiveness of diversity as a deception technique. Numerical results validate and support the effectiveness of the developed deception scheme. Ongoing research is extending the network model to study multi-layer attack graph to capture the vulnerabilities of the network over multiple levels of security including multi-domain networks.

ACKNOWLEDGMENT

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-19-2-0150. 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.

REFERENCES

[1] S. Peters, "A View From Inside a Deception, url = https://www.darkreading.com/edge/theedge/a-view-from-inside-a-deception/b/d-id/1341185."
[2] G. Forecast, "Cisco visual networking index: global mobile data traffic forecast update, 2017–2022," *Update*, vol. 2017, p. 2022, 2019.
[3] A. Kott, A. Swami, and B. J. West, "The internet of battle things," *Computer*, vol. 49, no. 12, pp. 70–75, 2016.
[4] C. A. Kamhoua, "Game theoretic modeling of cyber deception in the internet of battlefield things," in *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 862–862, IEEE, 2018.
[5] D. O. T. ARMY, "Army support to military deception," 2019.
[6] N. C. Rowe and H. C. Goh, "Thwarting cyber-attack reconnaissance with inconsistency and deception," in *Information Assurance and Security Workshop, 2007. IAW'07. IEEE SMC*, pp. 151–158, IEEE, 2007.
[7] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–40, 2019.
[8] G. F. Lyon, *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.
[9] R. De Smet and K. Marchal, "Advantages and limitations of current network inference methods," *Nature Reviews Microbiology*, vol. 8, no. 10, pp. 717–729, 2010.
[10] T. E. Carroll and D. Grosu, "A game theoretic investigation of deception in network security," *Security and Communication Networks*, vol. 4, no. 10, pp. 1162–1172, 2011.
[11] Y. Li, Y. Xiao, Y. Li, and J. Wu, "Which targets to protect in critical infrastructures-a game-theoretic solution from a network science perspective," *IEEE Access*, vol. 6, pp. 56214–56221, 2018.
[12] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pp. 46–57, ACM, 2005.
[13] A. Clark, Q. Zhu, R. Poovendran, and T. Başar, "Deceptive routing in relay networks," in *International Conference on Decision and Game Theory for Security*, pp. 171–185, Springer, 2012.
[14] S. Jajodia, P. Shakarian, V. Subrahmanian, V. Swarup, and C. Wang, *Cyber warfare: building the scientific foundation*, vol. 56. Springer, 2015.
[15] A. H. Anwar, C. Kamhoua, and N. Leslie, "Honeypot allocation over attack graphs in cyber deception games," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, pp. 502–506, IEEE, 2020.
[16] A. H. Anwar and C. Kamhoua, "Game theory on attack graph for cyber deception," in *International Conference on Decision and Game Theory for Security*, pp. 445–456, Springer, 2020.
[17] A. H. Anwar, N. O. Leslie, C. Kamhoua, and C. Kiekintveld, "A game theoretic framework for software diversity for network security," in *International Conference on Decision and Game Theory for Security*, pp. 297–311, Springer, 2020.
[18] H. Çeker, J. Zhuang, S. Upadhyaya, Q. D. La, and B.-H. Soong, "Deception-based game theoretical approach to mitigate dos attacks," in *International Conference on Decision and Game Theory for Security*, pp. 18–38, Springer, 2016.
[19] M. Bilinski, R. Gabrys, and J. Mauger, "Optimal placement of honeypots for network defense," in *International Conference on Decision and Game Theory for Security*, pp. 115–126, Springer, 2018.
[20] T. Zhang and Q. Zhu, "Hypothesis testing game for cyber deception," in *International Conference on Decision and Game Theory for Security*, pp. 540–555, Springer, 2018.
[21] A. Schlenker, O. Thakoor, H. Xu, F. Fang, M. Tambe, L. Tran-Thanh, P. Vayanos, and Y. Vorobeychik, "Deceiving cyber adversaries: A game theoretic approach," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 892–900, International Foundation for Autonomous Agents and Multiagent Systems, 2018.
[22] J. Letchford and Y. Vorobeychik, "Optimal interdiction of attack plans," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 199–206, International Foundation for Autonomous Agents and Multiagent Systems, 2013.
[23] S. Singh, P. K. Sharma, S. Y. Moon, D. Moon, and J. H. Park, "A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4543–4574, 2019.
[24] J. Briffaut, J.-F. Lalande, and C. Toinard, "Security and results of a large-scale high-interaction honeypot.," *JCP*, vol. 4, no. 5, pp. 395–404, 2009.
[25] A. B. Sarr, A. H. Anwar, C. Kamhoua, N. Leslie, and J. Acosta, "Software diversity for cyber deception," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–6, IEEE, 2020.
[26] B. An, M. Tambe, and A. Sinha, "Stackelberg security games (ssg): Basics and application overview," *Improving Homeland Security Decisions*, p. 485, 2017.
[27] T. Basar and G. J. Olsder, *Dynamic noncooperative game theory*, vol. 23. Siam, 1999.
[28] J. F. Nash *et al.*, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.
[29] M. Farzaneh, "Graph Coloring by Genetic Algorithm." https://www.mathworks.com/matlabcentral/fileexchange/74118-graph-coloring-by-genetic-algorithm, 2020. [MATLAB Central File Exchange. Retrieved September 2, 2021.].
[30] D. Fudenberg, F. Drew, D. K. Levine, and D. K. Levine, *The theory of learning in games*, vol. 2. MIT press, 1998.