2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) | 978-1-6654-6483-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/PIMRC56721.2023.10293814

Optimal Honeypot Allocation using Core Attack Graph in Cyber Deception Games

Achile Leonel Nguemkam
University of Dschang, Cameroon
achilenguemkam@gmail.com

Ahmed H. Anwar
US Army Research Laboratory, USA
a.h.anwar@knights.ucf.edu

Vianney Kengne Tchendji
University of Dschang, Cameroon
vianneykengne@yahoo.fr

Deepak K. Tosh
University of Texas at El Paso, USA
dktosh@utep.edu

Charles Kamhoua
US Army Research Laboratory, USA
charles.a.kamhoua.civ@mail.mil

***Abstract*—Honeypots appear today as a defense strategy to trap intelligent cyber attackers who can detect traditional security measures. The scalability of existing algorithms for solving some classes of game theory is very limited due to large-scale networks. This paper opens the door to a new approach to allocate honeypots in the network, to increase attackers' costs, during the lateral movement of the APT attack. We use the core attack graph that can show the main routes an attacker can take toward the goal. This allows the defender to use a limited number of honeypots focusing its efforts only on critical nodes over the main attacker routes. The effectiveness and scalability of the proposed approach are evaluated over different network topologies, a varying number of honeypots, network size, and density. Numerical results show that the defender reward over the core attack graph is quite similar to that obtained on the original attack graph while significantly reducing the defender's actions and computation time.**

***Index Terms*—Core attack graph, Cyber Deception, Game theory, Honeypots.**

I. Introduction

Network security is a set of rules and configurations designed to protect the integrity, confidentiality, and accessibility of computer networks and data using software and hardware technologies. Today, the exponential rise of cyber attacks can exist in a broad number of areas, including data, devices, applications, users, and locations.

According to Spitzner [1], "a honeypot is a decoy computer resource whose value lies in being probed attacker, or compromised". The use of honeypots is a basic form of cyber deception used to create the appearance of important targets to the attacker. Thus, the attackers are in several cases disoriented from the initial target by attacking the honeypots rather than the real infrastructure. Even though the honeypots can be used for different purposes such that reconnaissance tools, using their intrusion attempts to assess the adversary's techniques, capabilities, and sophistication.

On the one hand, resources are needed to allocate the honeypots, thus the use of a large number of honeypots can lead to enormous resources, on another side the use of a very limited number of honeypots may prove ineffective in the face of sophisticated attacks. Game theory is one of the methods that develop the best strategies to respond to this dilemma. To better understand the attacker's goal, we adopt an attack graph framework to generate all network vulnerabilities using tools such as MulVAL[2]. Thus our research aims to develop a novel scalable optimal honeypot allocation strategy in the network, to use a minimal number of honeypots with higher efficiency. This induces the scalability of the proposed solution. The mains contributions of the paper are as follows:

- We propose a cyber deception game between the network administrator (defender) and an adversary (attacker). In this game the defender allocates a set of honeypots over the core attack graph to deceive cyber adversaries;
- We analytically prove that the use of the core attack graph yields very close game value as the full attack graph;
- We numerically show that the proposed approach enhanced the scalability of the game solver.

The rest of this paper is organized as follows. In section II, the related works are presented. Section III provides an overview of the core attack graph. Section IV described the proposed game model and our solution for solving this cyber deception game. Section V shows the scalability of the proposed method through simulations. Section VI concludes this paper and presents future research directions.

II. Related work

Initially, the attack graph is a succinct representation of all paths through a system that ends in a state where an intruder has successfully achieved his goal S. Jha et al. [3]. They have been used in cyber deception games for many purposes.

Barrerè et al. [5] introduce the concept of a core attack graph, a compact representation of the main routes an attacker can take toward specific network targets. In this work, we introduce a complementary view using game theory to find the optimal defense strategy allocation honeypots of the network, which is not explored in [5].

In recent years, many papers introduced the idea of honeypots allocating over the attack graph using game theory. The authors in [6, 7] use the Stackelberg game to increase the uncertainty of the attacker. Radek et al. [8] developed two game-theoretic models that provide insight into how valuable

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on November 12,2024 at 14:18:13 UTC from IEEE Xplore. Restrictions apply.

honeypots should look to maximize the probability that a rational attacker will attack a honeypot.

Ahmed et al.[4] developed a new approach to allocate a honeypot over the attack graph using two-person zero-sum strategic form games to reduce the capability of the attacker to reach a target node. Their game model is limited to the number of possible actions to choose from at a single decision point in the game.

However, the main limitation of the kind aforementioned one-shot games comes from the number of actions of players. In the lateral movement normal-form game, the defender's actions correspond to placing a finite number of honeypots over the edges of the attack graph. To solve this problem, in this paper, we propose an optimal allocation honeypot policies by finding mixed-strategy Nash path equilibrium using a core attack graph. This core attack graph allows the defender to determine the main routes an attacker can take toward the target node. Therefore reducing the number of defender actions and then, improving the scalability of the game.

III. Core attack graph

A. Core graph formalisation

Given a directed graph $G = (V, E)$, a source node s, and a target node t, the objective of a core graph is to structurally grasp the main attack routes from s to t. Like standard attack graphs, [5], nodes in a core graph represent host privileges (e.g. user, root privileges) while edges are based on reachable host vulnerabilities that require certain privileges and provide others. Core graphs are built systematically by summarizing multiple alternative paths between any two connected nodes in the input graph G, and keeping in the core graph only the attack paths that cannot be summarised into any other graph link. We call these paths core paths. Mathematically, a core path coincides with what in graph theory is known as an induced path [10]. A path p is an induced path of G if any two adjacent nodes in p are connected in G and any two non-adjacent nodes in p are not connected in G. In other words, a core path between two nodes v_0 and v_n in G is a path p where each node $v_i \in p$ only connects to its immediate subsequent node v_{i+1} and there are no shortcuts to any other node ahead in the sequence over G. Inflated paths, on the other hand, are paths that contain at least two non-adjacent nodes that are connected in G. In graph-theoretical terms, the relationship between core paths and inflated paths is a homeomorphism [11]. We use $p' \preceq p$ to denote that p is an inflated path of p'. We now formally define the concepts of core paths and core graphs.

B. Core path in a graph

A path $p(v_0, ..., v_n)$ in $G = (V, E), n > 0$, is a core path if and only if there is no other path $p' \neq p$ such that $p' \preceq p$, i.e.: $\forall v_i \in p, \forall k \in [2, n-i], (v_i, v_{i+k}) \notin E$. In addition, each edge $(v_i, v_{i+k}) \in p$ structurally reflects that v_i can not only reach v_{i+1} directly but also summarizes all the alternative paths in which v_i can reach v_{i+1} over G. To clarify, let us consider the example illustrated in Figure 1.

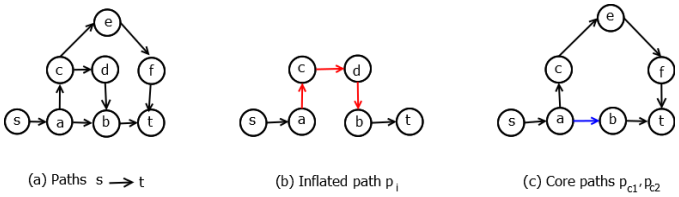


Fig. 1: Core attack graph

Figure 1a shows a small graph G with multiple paths from s to t. Figure 1b depicts one possible path p_i from s to t. It can be observed in the original graph G that a is connected to b and therefore path p_i violates the definition of core path, since $(a, b) \in E$. On the other hand, path $p_{c1}(\{s; a; b; t\})$, illustrated in Figure 1c, constitutes a core path from s to t since no shortcuts occur according to G, thus $p_{c1} \preceq p_i$. In addition, the core edge (a,b) summarises both paths in G, namely, $\{a; b\}$ and $\{a; c; d; b\}$. Path $p_{c_2}(\{s; a; c; e; f; t\})$ is also a core path since nodes a and t are not directly connected and therefore $\{c; e; f\}$ cannot be summarised into any edge. Note that core paths are not necessarily shortest paths: p_{c_1} has length 3 while p_{c_2} has length 5.

The core attack graph associated to graph G is a graph $C^G_{s,t}$ = ($V_c \subseteq V$, $E_c \subseteq$E) with $s \in V$ a source node (entry point) and $t \in V$ a target node is the result for transformation $\Gamma(G, s, t)$ defined as the union of core paths from s to t in G as follows:

$\Gamma(G, s, t) \equiv C^G_{s,t} \equiv \bigcup p$ such that $\nexists p' \in P^G_{s,t}$, $p' \preceq p$ with $p \in P^G_{s,t}$. Where $P^G_{s,t}$ is the set of all paths from node s to node t in G [5].

IV. Game model formulation

Attack paths are vital locations for optimal honeypot allocation [3]. We formulate a two-player zero-sum strategic game between a defender and an attacker to characterize the optimal honeypot allocation. This game is defined as a tuple $(\mathcal{N}, \mathcal{A}, \mathcal{R})$, where:

- $\mathcal{N} = \{1, 2\}$ is the set of two players. Player 1 is the network administrator and player 2 is the attacker;
- $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ is the game action space, where $\mathcal{A}_1$ is the defender action spaces and $\mathcal{A}_2$ is the attacker action spaces. $a = (a_1, a_2) \in \mathcal{A}$ is an action profile with $a_i \in \mathcal{A}_i$, $i \in \{1, 2\}$, that determines the reward received by both players.
- $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2\}$, where $\mathcal{R}_i : \mathcal{A} \mapsto \mathbb{R}$, $i \in \{1, 2\}$ is a reward function for player i, with $\mathcal{R}_1$+$\mathcal{R}_2 = 0$;

In the example shown in Figure 2a, The attacker wants to reach node v_7 from the entry point v_1 by traversing the edges of the graph while minimizing the cost to do so. Given an entry point, the network administrator constructs the core attack graph as defined in Figure 2b.

In our game formulation, we assume that the defender knows the probability that the attacker will penetrate the network through an entry point. Although similar to the game theory model in [4], the main difference is that: in this proposed approach, the defender uses only the edges in

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on November 12,2024 at 14:18:13 UTC from IEEE Xplore. Restrictions apply.

the core attack graph to allocate a finite set of honeypots. This is because if the defender allocates several honeypots to cover a set of edges over all the paths in the network, the game's complexity increases exponentially with the number of honeypots used. In doing so, we reduce defense resources by using a limited number of honeypots in the network [12].

A. Defender's actions

The defender actions are honeypot allocations. More generally the defender action space is
$\mathcal{A}_1 = E_c^{N_h}$ where $E_c^{N_h}$ denotes N_h-element subsets of E_c, and $E_c \subseteq E$ is the set of edges over the core attack graph. In this game example, the defender has all eight edges on which he can allocate the honeypots. These edges are given by E_c=$\{\{v_1, v_3\}; \{v_3, v_4\}; \{v_4, v_7\}; \{v_1, v_2\}; \{v_2, v_5\}; \{v_5, v_7\}; \{v_1, v_6\}; \{v_6, v_7\}\}$. The defender generates the core attack graph from an entry point with the highest probability as mentioned above. Since the defender is uncertain of the attacker's position, he also considers a no-allocation action as a possible choice to avoid the allocation cost. The defender allocates a set of h honeypots over the edges in the core attack graph to increase the attacker's cost, and to deter the attacker from reaching his target. In Figure 2, it is assumed that the defender has 3 honeypots to allocate the defender has up to 6.11^3 possible actions on the whole graph and 6.8^3 on the core graph.

B. Attacker's actions

The attacker action are directed paths in G (original graph) reaching the target node. Its action is to choose the path which minimizes the attacker's cost between the node which has been compromised and its successor which intends to compromise. Since the attacker wants to minimize the attack cost and remain stealthy, he may choose to completely back off in some cases to avoid excessive attack cost. An attack cost represents the risk of being captured by the defender.

C. Reward function

The player's utilities combine their reward from the entry toward the target node along the attacker's path. A path of length $k \geq 1$ is an ordered sequence $(e_1, ..., e_k)$ of k adjacent edges $e_i = \{v_i; v_{i+1}\}$. Let

$$\begin{array}{rccl} f & : & E & \rightarrow & \mathbb{R}^+ \\ & & e_i & \mapsto & f(e_i) = C \end{array}$$

the attacker cost at each edge, the constant C represented the attack cost on each edge in the graph.
The cost paid by an attacker to traverse each path p is equal to the sum of the costs paid on each edge of the path, this cost is given by: $C_{A,p} = \sum_{e_i \in p} f(e_i)$.

The defender tries to increase the attacker's cost by deploying h honeypots over the edges in the core attack graph. Thus the attacker cost increases to $f(e_h) = C_h$ using the edges e_h, where e_h denotes that the honeypot is deployed on edge e_i. In this case, the attacker cost is given by: $C_{h,P} = \sum_{e_i \in \tilde{P}} f(e_i) + 1_{h \in P}(f(e_h) - f(e_i))$ and

$$\tilde{P} = \begin{cases} P & \text{if } h \notin P \\ \{e_i\}\ i = 1, ..., h & otherwise. \end{cases}$$

As stated earlier, the defender may decide no allocate any honeypots. The attacker also has two possible choices, given the associated cost per attack, the attacker may decide to back off completely or go through the path.

The reward of the defender is the cost of the attacker, thus the general reward matrix of the defender is presented as follow:

$$R_1 = \begin{pmatrix} C_{h,P} & 0 \\ C_{A,P} & 0 \end{pmatrix}$$

The attacker reward matrix is $R_2 = -R_1$. The reward $R_1(1,1)$ represent a captured attacker. That is, the defender installed a honeypot on the attack path. On the other hand, $R_1(2,1)$ represent a successful attack, where the honeypot is installed at a different attack path. In $R_1(1,2)$ and $R_1(2,2)$, the attacker backs off in order to avoid the cost of attack or to be captured.

Proposition 1 The core paths are the best places to allocate honeypots.

Proof. We recall that the best places to allocate honeypots are the paths with the minimum expected attacker cost.

Let s and t respectively be the source node and the target node on the attack graph $G(V, E)$, the set of all paths from node s to node t in G is denote by $P_{s,t}^G$ as mention above.
Let $\mathbb{P}_\mathbb{I}$ and $\mathbb{P}_\mathbb{C}$ the set of all inflated paths and core paths respectively in graph G, we have $P_{s,t}^G = \mathbb{P}_\mathbb{I} \cup \mathbb{P}_\mathbb{C}$.
We recall that a path $p(\{v_s; ...; v_t\}) \in \mathbb{P}_\mathbb{C}$ if and only if there is no other path $p' \neq p$ such that $p' \preceq p$. i.e.: $\forall v_i \in p, \forall k \in [2, t-i], \{v_i, v_{i+k}\} \notin E$.
A path $p(\{v_s; ...; v_t\}) \in \mathbb{P}_\mathbb{I}$ if and only if there exists at least $p' \neq p$ such that $p' \preceq p$. In other words $\complement_{P_{s,t}^G}^{\mathbb{P}_\mathbb{C}} = \mathbb{P}_\mathbb{I}$.
The relation $p' \preceq p$ denote that p is an inflated path of p'.
It should be noted that if $p' \preceq p$, then p' is dominated by p.
We recall that the attacker's action is to choose the path which minimizes the attacker's cost between the node which has been compromised and its successor which intends to compromise.
Let $E_{min} = \{\ p \in P_{s,t}^G\ /\ \forall$ node $v_i \in p$ no shortcuts occur between node v_i and successor $(v_i)\ \}$, the set of dominant paths in G, we have $E_{min} = \mathbb{P}_\mathbb{C}$.

Let's show that $E_{min} \subseteq \mathbb{P}_\mathbb{C}$.
By contradiction, suppose that $E_{min} \not\subseteq \mathbb{P}_\mathbb{C} \Rightarrow E_{min} \subseteq \mathbb{P}_\mathbb{I}$ because $\complement_{P_{s,t}^G}^{\mathbb{P}_\mathbb{C}} = \mathbb{P}_\mathbb{I}$, which is a contradiction because $\forall\ p \in \mathbb{P}_\mathbb{I}$ there exists at least $p' \neq p$ such that $p' \preceq p$, we conclude that $E_{min} \subseteq \mathbb{P}_\mathbb{C}$.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on November 12,2024 at 14:18:13 UTC from IEEE Xplore. Restrictions apply.

On the other hand, let's show that $\mathbb{P}_{\mathbb{C}} \subseteq E_{min}$. indeed, $\forall\ p \in \mathbb{P}_{\mathbb{C}}$, there is no other path $p' \neq p$ such that $p' \preceq p$, hence $\mathbb{P}_{\mathbb{C}} \subseteq E_{min}$, and finally, $E_{min} = \mathbb{P}_{\mathbb{C}}$. We conclude that the core paths are the best places to install honeypots.

□

lemma 1 The Nash equilibrium of the game get on the original attack graph is mathematically equivalent to the Nash equilibrium get on the core attack graph.

Proof. From Proposition 1, the set of dominant paths on the original attack graph is a bijection to the set of paths on the core attack graph, furthermore, the Nash equilibrium can be obtained by eliminating dominated paths. □

D. Game example

Figure 2 represents the attack graph, there are seven hosts $\{v_1; ...; v_7\}$ the entry point is node v_1 and the target node is v_7, the attacker aims to reach node v_7 minimizing the cost to do so. The attacker has six possible paths listed as follows:

$$P^G_{s,t} = \{P_1\{v_1; v_6; v_7\}; P_2\{v_1; v_6; v_5; v_7\}; P_3\{v_1; v_2; v_5; v_7\}; P_4\{v_1; v_3; v_2, v_5; v_7\}; P_5\{v_1; v_3; v_4; v_7\}; P_6\{v_1; v_3; v_4; v_5; v_7\}\}$$

$$\mathbb{P}_{\mathbb{I}} = \{P_2\{v_1; v_6; v_5; v_7\}; P_4\{v_1; v_3; v_2; v_5; v_7\}; P_6\{v_1, v_3, v_4, v_5, v_7\}\}$$

$$\mathbb{P}_{\mathbb{C}} = \{P_1\{v_1; v_6; v_7\}; P_3\{v_1; v_2; v_5; v_7\}; P_5\{v_1; v_3; v_4; v_7\}\}$$

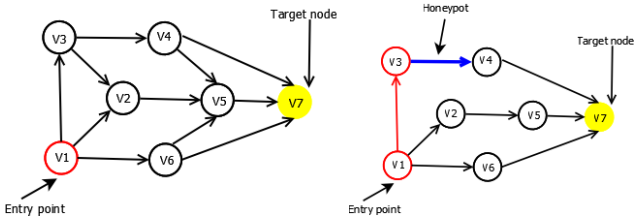
Let's build E_{min}.

We have $P_1 \preceq P_2 \Rightarrow P_2$ is strictly dominated by P_1 because between node v_6 and his succesor v_7, we have $d(v_6, v_7) < d(v_6, v_5)$ + $d(v_5, v_7)$

$P_3 \preceq P_4 \Rightarrow P_4$ is strictly dominated by P_3 because between node v_1 and his successor v_2, we have $d(v_1, v_2) < d(v_1, v_3)$ + $d(v_3, v_2)$

$P_5 \preceq P_6 \Rightarrow P_6$ is strictly dominated by P_5. Because between node v_4 and his successor v_7, we have $d(v_4, v_7) < d(v_4, v_5)$ + $d(v_5, v_7)$ where $d(v_i, v_{i+1})$ denote the distance between node v_i and v_{i+1}.

In this case $E_{min} = \{P_1; P_3; P_5\}$ we can conclude that $E_{min} = \mathbb{P}_{\mathbb{C}}$.



(a) Attack graph (b) Core attack graph

Fig. 2: Game example.

E. Solution concepts

In this section, we will introduce the definition of player's strategies, and Nash path equilibrium. The mixed strategy is a strategy in which the player selects a specific action between the overall actions with some probabilities. Let $\sigma_i \in \Sigma_i$ a mixed strategies of the players $i \in \{$ attacker; defender$\}$, with Σ_i the set of all mixed strategies, Let $u_i(\sigma_i, \sigma_{-i})$ the expected utility for player i if the players are following the strategies in S= (σ_i, σ_{-i}). A Nash equilibrium is any strategy profile S such that for all player i and strategy σ_i for player i, it holds: $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$; A Nash path is a path corresponding to a Nash equilibrium. Best response mixed strategy for player i against the strategy of the opponent s_i denoted $BR_i(\sigma_{-i})$ is such that $\forall \sigma_i \in \Sigma_i$: $u_i(\sigma_i, \sigma_{-i}) \leq u_i(BR_i(\sigma_{-i}), \sigma_{-i})$. Let us consider d the defender and a the attacker, the normal form equilibrium is a strategy profile:

$$(\sigma_d, \sigma_a) = \underset{\sigma'_d \in \Sigma_d; \sigma'_a \in BR_a(\sigma'_d)}{\text{argmax}}\ u_d(\sigma'_d, \sigma'_a)$$

F. Defender's linear program

The main goal in this section is to compute a Nash path equilibrium that maximizes the defender's expected utility, taking into account the fact that the attacker will play his best response.

The linear program (LP) developed below computes the defender's optimal mixed strategy verifying the constraint that the attacker plays a mixed strategy best response.

We recall that σ_d is a defender mixed strategy and σ_a is an attacker mixed strategy. In a mixed strategy $\sigma_d \in \Sigma_d$, action a^i_1 is played with probability x_i such that:
$\sigma_d = [x_1, x_2, ..., x_n]^T$ with $n = |A_1|$. Similarly in a mixed strategy $\sigma_a \in \Sigma_a$, action a^i_2 is played with probability y_i such that: $\sigma_a = [y_1, y_2, ..., y_n]^T$ with n=$|A_2|$.

The defender expected reward is define as follows:
$U_1 = \sigma_d^T R_1 \sigma_a$, where R_1 is a reward matrix.

The optimization problem of defender as linear program is given as follow:

$$\max_{\sigma_d} U_1$$

subject to :

$$\sum_{a_1 \in A_1} u_1(a_1, a_2) x_{a_1} - U_1 \geq 0\ \forall a_2 \in A_2 \qquad (4.6.1)$$

$$\sum_{i=1}^{n} x_i = 1,\ x_i \geq 0 \quad i = 1, ..., n \qquad (4.6.2)$$

The attacker expected reward is defined by $U_2 = -U_1$. According to the minimax theorem, U_1 holds constant in all equilibria, and is the same value that the defender achieves under a minimax strategy by the attacker.

The attacker also solves the minimizing LP to ensure that the optimal mixed strategy σ_a is a best response for every possible action played by the defender.

The inequality in equation (4.6.1) follows from the definition of the Nash equilibrium. Equations (4.6.2) ensure that the attacker and defender strategies are a valid probability

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on November 12,2024 at 14:18:13 UTC from IEEE Xplore. Restrictions apply.

distribution. The optimal mixed strategies for both players developed above form a Nash Equilibrium for both players. A Nash path is a path corresponding to this Nash equilibrium.

V. Numerical Results

In this section, we present numerical results that validate the effectiveness of the proposed approach and illustrate its scalability. The different steps of this experimental section are described as follows: In section V.A, a performance analysis of the core attack graph over different network topologies is made. In section V.B, we compare the defender reward over the core attack graph and the original attack graph. In section V.C, we investigate the scalability of models.

A. Performance of core attack graph over different network topologies

In this section, the defender allocates one honeypot. The number of possible defender actions depends on the size and density of the input graph. When the input graph tends to be dense and cyclic (the edge creation probability is 0.8), the number of possible defender actions is very small on the core attack graph compared to that obtained on the original graph. This is because the input graph is higher connected, and then the core attack graph is able to summarise a large portion of this graph. For example in Figure 3, for a network of 50 nodes, the number of possible defender actions is 1958 over the dense original graph and 316 over the core attack graph. On the contrary, when the input graph tends to be sparse (the edge creation probability is 0.2) with more independent paths and fewer cycles, the core graph will not show a significant structure reduction, and the number of possible defender actions will be comparable to that. For example in Figure 3, for a network of 50 nodes, the number of possible defender actions over the original sparse attack graph is 543 and 392 over the corresponding core attack graph.

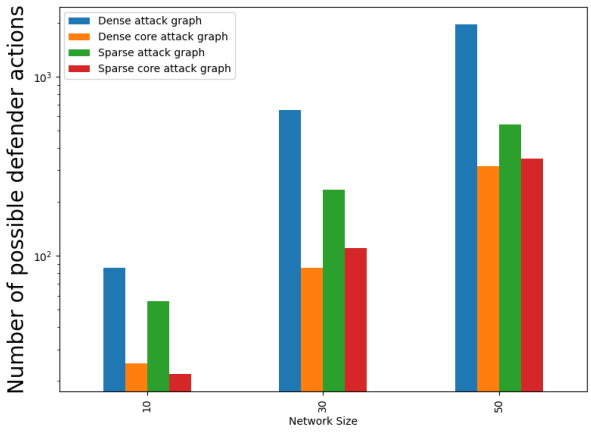
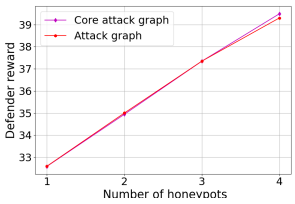


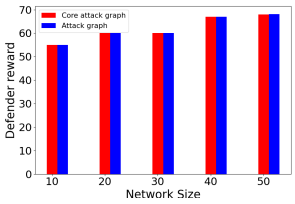
Fig. 3: Performance analysis over different topologies

B. Comparing of the defender reward

The defender reward depends also on the size and density of the input graph. In this section we consider the cyclic and dense input graph.Figure 4b compares the defender reward over the original attack graph and core attack graph using one honeypot when the network size varies between 10 and 50 nodes. In this figure4a, we also compare the defender reward varying the number of honeypots with input network at 50 nodes. We can observe that the defender reward over the core attack graph and the original attack graph is quite similar. For the input graph of 50 nodes, the defender reward over the original graph is 68.103 and 68.1002 over the core attack graph. Also in Figure 4a when the number of honeypots increases the defender reward over the core attack graph tends to be beneficial comparing to the defender reward over the original attack graph. For example, with 4 honeypots, and a network of 50 nodes, the defender reward over the original attack graph is 39.302 and 39.490 over the core attack graph. In Figure 6, we plot the defender Nash path equilibrium (NPE) strategy and defender random strategy in order to show the effectiveness of our game formalism, we can observe that the defender reward using Nash equilibrium strategy is always greater than the defender random strategies for both graphs (core attack graph and original attack graph). These results are obtained at attack cost on each edge C=10 and $C_h = 15$.



(a) Varying number of honeypots



(b) Varying network size

Fig. 4: Comparing defender reward.

C. Scalability analysis

We have also analyzed scalability aspects over cyclic input graphs of various sizes. The objective here is to compare the run-time required to obtain the defender reward over the original attack graph and core attack graph (Figure 5), compare the defender reward over the core attack graph and original attack graph (Figure 4b). We recall that run time includes only instances solved by the algorithm. Figure 5 shows the results on run time algorithms for the core attack graph and original attack graph while varying the size of the input graph between 10 and 50 nodes. For n=10 the average time is t=2.01 seconds over the original attack graph and t=1.98 second over the core attack graph, while for n=50 the average time over the core attack graph is t=5.21 seconds and t=43.16 seconds over the original attack graph, this is because the input network tends to be fully connected. We can observe that the run time of the algorithm over the core attack graph is very small than the run time of the algorithm over the original attack graph.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on November 12,2024 at 14:18:13 UTC from IEEE Xplore. Restrictions apply.

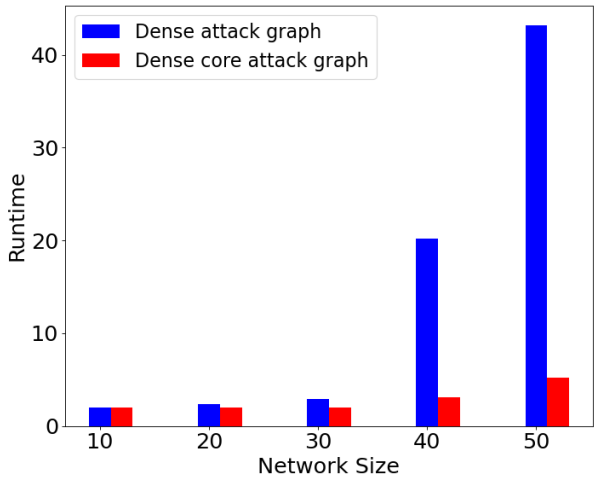


Fig. 5: Run-time

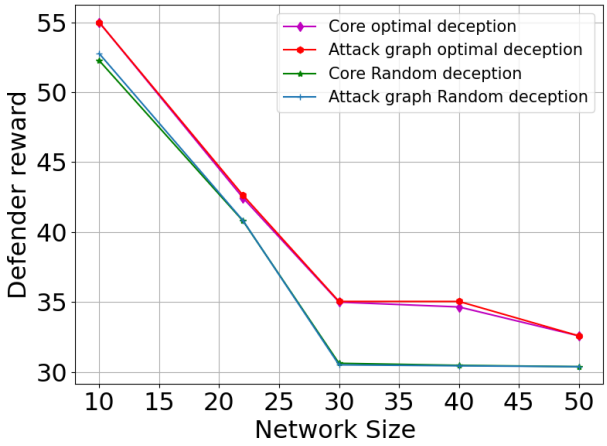


Fig. 6: Comparing defender reward at NPE and random allocation at different network sizes using one honeypot

VI. Conclusion and Future Work

In this paper, we propose a novel approach to allocating honeypots in the network to increase the attack cost and mitigate damages. To achieve this goal, core attack graph has been leveraged to generate the main routes an attacker can take to reach its target. This graph allows the defender to use a limited number of honeypots by focusing only on the critical node on the main routes. The interaction between the attacker and defender has been modeled as a zero-sum normal form game to find the optimal strategy for honeypot allocation. This strategy (Nash path) has been computed using a mixed strategy linear program method. Some numerical results obtained from this computation have shown the scalability of the proposed approach. One of the main limitations of this proposed approach come from the fact that is not always feasible to ensure a simultaneous move, especially in the case of competing scenarios. Hence, investigating a Stackelberg equilibrium in this proposed approach will be one of our further work.

Acknowledgments

Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-21-1-0326. 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 Office 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] SPITZNER, Lance. The honeynet project: Trapping the hackers. IEEE Security Privacy, 2003, vol. 1, no 2, p. 15-23.

[2] OU, Xinming, BOYER, Wayne F., et MCQUEEN, Miles A. A scalable approach to attack graph generation. In : Proceedings of the 13th ACM conference on Computer and communications security. 2006. p. 336-345.

[3] JHA, Somesh, SHEYNER, Oleg, et WING, Jeannette. Two formal analyses of attack graphs. In : Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15. IEEE, 2002. p. 49-63.

[4] ANWAR, Ahmed H. et KAMHOUA, Charles. Game theory on attack graph for cyber deception. In: International Conference on Decision and Game Theory for Security. Springer, Cham, 2020. p. 445-456.

[5] BARRÈRE, Martín et LUPU, Emil C. Naggen: A network attack graph generation tool—IEEE CNS 17 poster. In: 2017 IEEE Conference on Communications and Network Security (CNS). IEEE, 2017. p. 378-379.

[6] ZHANG, Ting, CHOI, Tsan-Ming, et ZHU, Xiaowei. Optimal green product's pricing and level of sustainability in supply chains: Effects of information and coordination. Annals of Operations Research, 2018, p. 1-26.

[7] NOUREDDINE, Mohammad A., FAWAZ, Ahmed, SANDERS, William H., et al. A game-theoretic approach to responding to attacker lateral movement. In: International Conference on Decision and Game Theory for Security. Springer, Cham, 2016. p. 294-313.

[8] PÍBIL, Radek, LISÝ, Viliam, KIEKINTVELD, Christopher, et al. Game theoretic model of strategic honeypot selection in computer networks. In : International conference on decision and game theory for security. Springer, Berlin, Heidelberg, 2012. p. 201-220.

[9] NEETIL, Jaroslav et DE MENDEZ, Patrice Ossona. Sparsity: graphs, structures, and algorithms. Springer Publishing Company, Incorporated, 2012.

[10] LAPAUGH, Andrea S. et RIVEST, Ronald L. The subgraph homeomorphism problem. Journal of Computer and System Sciences, 1980, vol. 20, no 2, p. 133-149.

[11] HORÁK, Karel et BOŠANSKÝ, Branislav. Solving partially observable stochastic games with public observations. In : Proceedings of the AAAI conference on artificial intelligence. 2019. p. 2029-2036.

Authorized licensed use limited to: Technical University of Denmark DTU Library. Downloaded on November 12,2024 at 14:18:13 UTC from IEEE Xplore. Restrictions apply.