2020 IEEE Conference on Control Technology and Applications (CCTA)
August 24-26, 2020. Montréal, Canada

A Receding-Horizon MDP Approach for Performance Evaluation of Moving Target Defense in Networks

Zhentian Qian, Jie Fu, and Quanyan Zhu

Abstract— **In this paper, we study the problem of assessing the effectiveness of a proactive defense-by-detection policy with a network-based moving target defense. We model the network system using a probabilistic attack graph–a graphical security model. Given a network system with a proactive defense strategy, an intelligent attacker needs to perform reconnaissance repeatedly to learn about the locations of intrusion detection systems and re-plan optimally to reach the target while avoiding detection. To compute the attacker's strategy for security evaluation, we develop a receding-horizon planning algorithm using a risk-sensitive Markov decision process with a time-varying reward function. Finally, we implement both defense and attack strategies in a synthetic network and analyze how the frequency of network randomization and the number of detection systems can influence the success rate of the attacker. This study provides insights for designing proactive defense strategies against online and multi-stage attacks by a resourceful attacker.**

I. Introduction

Cyber networks in industrial control systems are often targeted by malicious and resourceful attackers. An attacker can identify system vulnerabilities through reconnaissance and compromise the security of a network through calculated, multi-stage attacks. To counter the attacks, a network system can employ a mix of cybersecurity mechanisms, from traditional firewalls and intrusion detection to moving target defense [1] and cyberdeception [2] with honeypots [3]. However, it is difficult to measure the effectiveness of dynamic defense techniques. The lack of understanding their security gains hinders the practical deployment of advanced dynamic defenses.

Formal graphical security models, such as attack graphs [4] and attack-defense trees [5], have been developed [6] to evaluate security properties of a cyber system. An attack graph captures multiple paths that an attacker can carry out by exploiting vulnerabilities to reach the attack goal. Recent works [7], [8] have investigated the security property of Moving Target Defense (MTD) using probabilistic attack graphs, where probabilistic transitions are uncertainties created by network-based randomization. However, there has not been an analytical model for evaluating the effectiveness of MTD for detection.

To detect the presence of an attacker, network administrators often place Intrusion Detection Systems (IDSs) at several points in the network to monitor traffic to and from all devices on the network and detect suspicious activities. They are essential components of proactive defenses, where the defender is not aware of the existence of the attacker but deploys some pre-defined security protocols. The question we aim to address is that, given a proactive defense strategy and an attacker who performs a sequence of actions to reach the target, as in lateral movement attacks [9], how effective is a proactive defense strategy to detect the attacker before the attacker succeeds?

For IDSs at fixed locations, an attacker can learn their locations during reconnaissance and avoid them while carrying out an attack. An effective detection technique, called "roaming IDSs", is used to randomize the location of IDSs in the network. For example, a flow-based IDS [10] allows network flow to pass through and examined by IDS on a per-flow basis using software-defined networking. Roaming decoys [11] have also been used to mitigate Denial-of-Service attacks by shuffling the decoy locations in a network. This randomization creates uncertainty for the attacker and also increases his cost, as the attacker has to perform reconnaissance to determine the new IDS locations to avoid detection.

To understand how effective the defense strategy is, we need to understand how the attacker behaves given the uncertainty. To this end, we model the network with dynamic defense as a time-varying probabilistic attack graph, which can be modeled as a Markov Decision Process (MDP) with a time-varying probabilistic transition function and a reward function. Then, we compute the attack strategy using risk-sensitive finite-horizon planning, and iteratively re-plan the attack strategy using a receding horizon framework. Given the computed attack strategy, we can evaluate the effectiveness of the detection and defense strategy by characterizing the relation among the probability of successful and stealthy attack, the number of IDSs, and the shuffling frequency of the IDSs.

Finally, the paper is structured as follows: In Section III, we introduce preliminaries on attack graphs, roaming IDS defense strategy, and formulate the problem. In Section IV, we design the receding-horizon attack planning in the time-varying network. In Section V, we evaluate the performance of defense against the proposed online attacker planner. Section VI concludes the paper.

II. Related Work

In the context of moving target defenses, attack graph models [12], [13] and dynamic game models [14]–[18] have been proposed to capture the strategic interactions between

Z. Qian, J. Fu are with the Robotics Engineering Program and Dept. of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609 USA. {zqian, jfu2}@wpi.edu

Q. Zhu is with Dept. of Electrical and Computer Engineering, New York University, USA {quanyan.zhu}@nyu.edu

978-1-7281-7140-1/20/$31.00 ©2020 IEEE

an attacker and a defender. In [14], a multi-stage game has been proposed to model the kill chain of the adversary. In [19], the authors have proposed a multi-stage game of incomplete information to model a long-term interaction of a proactive defender and a stealthy attacker. In recent work [20]–[22], attack-defense trees are developed to incorporate defender's countermeasure [23] and capture the dependencies between actions and subgoals for both attacker and defender. These models are used to verify quantitative security properties expressed via temporal logic, based on the solutions of omega-regular games [24]–[26]. In [27], the authors have introduced online learning defense schemes that proactively interact with attackers to increase the attack cost and gather threat information. These approaches are applicable to synthesize reactive defenses: the defender is aware of the presence of the attacker and reacts to the attack actions in real time. In this work, we study proactive defense when the defender uses a fixed randomization strategy without knowing whether there is an attacker in the network.

For both reactive and proactive defense, one of the critical challenges in applying game theory to security is the performance evaluation of the attack behaviors. This work leverages a receding-horizon technique together with probabilistic attack graphs to assess the effectiveness of a class of cyber defenses that explicitly account for the attacker's uncertainties. The adversary model captures the key properties of the cyber kill chain [28], [29], in which an attacker explores the network and its vulnerability, moves laterally in the network, and takes actions to achieve the attack goals, such as data exfiltration, data destruction, or encryption for ransom. Performance evaluation is an essential first step toward the design of effective moving target defense. This work provides informative metrics that will be useful to address issues related to defense design, resource planning, and security investment.

III. Preliminaries and Problem Formulation

In this section, we present preliminaries on formal graphical security models, and then formulate the problem to evaluate the effectiveness of the proactive defense strategy.

Definition III.1 (Probabilistic attack graph)**.** A Probabilistic Attack Graph (PAG) is a probabilistic transition system $G = \langle S, A, P, s_0 \rangle$ where S is a set of network nodes, A is a set of attack actions, and $P : S \times A \to \text{Dist}(S)$ is a probabilistic transition function–that is, $P(s'|s, a)$ is the probability of the attacker reaching node s' from a (compromised) node s with an attack action a (targeted at s' only). The probability of failing to exploit a vulnerability results in a self-loop $P(s|s, a) = 1 - P(s'|s, a)$. The state s_0 is the initial entry node for the attacker.

The reader can think of the PAG as an MDP, in which the set of actions are the attacker's exploitation actions. The probability of an attacker successfully exploiting a vulnerability can be estimated based on the Common Vulnerability Scoring System (CVSS) [30], as used in [31], [32].

Using network-based MTD techniques, we can randomize the software/hardware or the topology of the network. We consider a case of IDS randomization techniques where the locations of IDSs can be sampled from the set of nodes of the network. For example, if $s \in S$ is sampled, then all flows into node s will be examined by an IDS and we say that node s is equipped with an IDS.

For simplicity, we assume that when the attacker sends a package to exploit the vulnerability of a target node and the node is equipped with an IDS, the attacker will be detected and blocked from the network.

We aim to evaluate the security level of the system for a proactive defense strategy, defined as follows.

Definition III.2. A *periodic* defender strategy $\delta(t + T_r) = \delta(t)$ that randomly selects k out of a subset $\mathcal{N} \subseteq \mathcal{S}$ of nodes in the network as the IDS locations every T_r steps.

Assumption III.1. The following assumptions are made for the attacker:

- The attacker knows the PAG but does not know the defender strategy δ and T_r.
- The attacker can exercise the network scan every step, before taking any attack action, to learn about the locations of IDSs at that moment.
- The defender's action of sampling IDSs is taken concurrently with the attack actions.

It is noted that if the defender uses a Poisson distribution over the period T_r, even if the attacker learns the mean and variance, he cannot know exactly when the IDSs have been shuffled. Thus, the assumption that the attacker does not know T_r is not necessary. If we assume that the attacker knows the defender's strategy, then the attacker's planning problem reduces to a standard MDP whose solution provides the worst case analysis of the network defense. In this work, we are interested in studying how the attacker's lack of information can lead to a less conservative assessment of defense strategy.

Definition III.3 (Reach-avoid attack objective)**.** Given the PAG and let $s_f \in S$ be the target for the attacker, the objective of the attacker is to avoid detection until reaching s_f.

Definition III.4 (Detection events)**.** An attacker can be detected if he attempts any action $a \in A$ at node s to reach target s', and s' is equipped with an IDS.

In other words, the attacker can be detected by exploiting a node equipped with an IDS, no matter whether the attack action is successful or not.

Example III.1. We introduce an example to illustrate the concept. Figure 1 depicts a small network with three hosts, equipped with SDN-enabled roaming IDS. At each time step, the IDS can be randomly assigned to a target host and monitor the flow. Figure 2a shows a transition in the PAG where the attacker has gained trust on host 1, which is an FTP server. The FTP server consists of a vulnerability

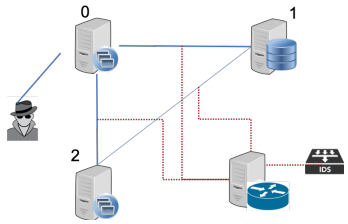
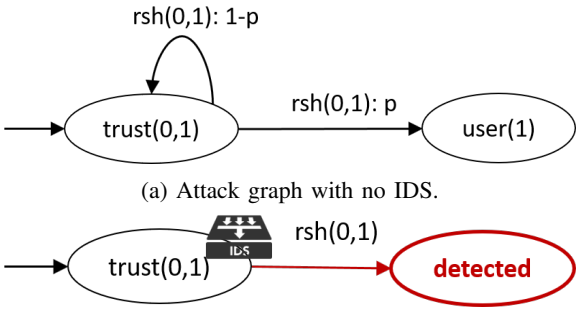


Fig. 1: An example of a small network with roaming IDS.



(a) Attack graph with no IDS.

(b) Attack graph with IDS.

Fig. 2: A fragment of the PAG: (a) Without IDS, the attacker carries out action to reach host 1 with some probability. (b) With IDS, the attack action is detected.

which allows the attacker to obtain reverse shell (rsh) on the system. By carrying out `rsh` attack on host 1, the attacker succeeds with probability p to gain user access on host 1, and with probability $1-p$ that his action fails. When the IDS is equipped with host 1, then the attacker's action `rsh` will be detected, leading to the sink state–detected–in Fig. 2b.

Problem 1. Given a defense strategy δ and an initial state $s_0 \in S$ of PAG, with what probability can the attacker achieve his attack objective? What is the best response of the attacker given the lack of knowledge in the defender's strategy?

IV. Attacker's Behavior Modeling

To understand how the attacker plans given the non-stationary environment, we introduce an attack behavior model using online planning in MDPs. In this section, we first introduce a preliminary on risk-sensitive, finite-horizon planning, and then present a receding horizon framework that iteratively solves finite-horizon problems in a time-varying MDP.

A. Preliminaries: Risk-sensitive planning in MDPs

Given an MDP $G = (S, A, P, s_0)$, where S, A, P are state, action spaces and transition function, respectively; s_0 is the initial state. We introduce an immediate reward function as:

$$r_t : S \times A \to \mathbb{R}^+, \ \forall\, t \in [t_0, t_0 + T - 1], \tag{1}$$

where $T \geq 0$ is a constant for finite horizon length. The terminal reward $r_{t_0+T} : S \to \mathbb{R}^+$ depends only upon the state $s \in S$. The finite-horizon risk-sensitive optimal planning problem is described as follows: Given the MDP, the immediate reward function $r_t, t \in [t_0, t_0 + T - 1]$ and the terminal reward function r_{t_0+T}, compute a policy $\Pi^{t_0} = (\pi_{t_0}, \pi_{t_0+1}, \ldots, \pi_{t_0+T-1})$ where $\pi_t : S \to \text{Dist}(A)$ maximizes the following objective:

$$J_{t_0}(\nu, \Pi^{t_0}) = E^{\nu,\Pi^{t_0}}\Big[\exp\Big(\lambda \sum_{n=t_0}^{t_0+T-1} r_n(S_n, A_n) + r_{t_0+T}(S_{t_0+T})\Big)\Big], \tag{2}$$

where λ is a discounting factor; ν is the distribution over states at $t = t_0$, in our case it only resides on a single state s_0; the expectation $E^{\nu,\Pi^{t_0}}$ is computed from the Markov chain induced using policy Π^{t_0}; i.e., the state and action processes $\{S_t\}_{t_0 \leq t \leq t_0+T}$, $\{A_t\}_{t_0 \leq t \leq t_0+T-1}$.

As shown in [33], the risk-sensitive objective can be minimized using linear programming, with the primal and dual linear programs formulated as follows.

Primal Linear Program:

$$\min_{\left\{\{u_t(s)\}_{s\in S, t_0 \leq t \leq t_0+T-1}\right\}} \sum_{s\in S} \nu(s) u_{t_0}(s),$$

subject to:

$$\begin{aligned} & u_{t_0+T-1}(s) \geq b_{s,a}, \quad \forall s \in S, \forall a \in A, \\ & u_t(s) - e^{r_t(s,a)} \sum_{s' \in S} P(s'|s,a) u_{t+1}(s') \geq 0, \\ & \forall\, s \in S, \forall a \in A, \forall t : t_0 \leq t \leq t_0 + T - 2, \end{aligned} \tag{3}$$

where

$$b_{s,a} := e^{r_{t_0+T-1}(s,a)} \sum_{s' \in S} P(s'|s,a) e^{r_{t_0+T}(s')}. \tag{4}$$

The solution of the primal LP provides $\{u_t(s) \mid s \in S, t_0 \leq t \leq t_0 + T - 1\}$, where $u_t(s) = \max_{\Pi^t} J_t(s, \Pi^t)$ (see (2)) with $\Pi^t = [\pi_t, \ldots, \pi_{t-t_0+T-1}]$.

Dual Linear Program:

$$\max_y \sum_{a\in A} \sum_{s\in S} b_{s,a} y(t_0 + T - 1, s, a)$$

subject to:

$$\begin{aligned} & \sum_{a\in A} y(t_0, s', a) = \nu(s'), \quad \forall s' \in S, \\ & \sum_{a\in A} y(t, s', a) = \\ & \sum_{a\in A} \sum_{s\in S} e^{r_{t-1}(s,a)} P(s'|s,a) y(t-1, s, a) \\ & \forall t : t_0 + 1 \leq t \leq t_0 + T - 1, \ \forall s' \in S. \end{aligned} \tag{5}$$

Here, the decision variables y are taken as $y = \{y(t, s, a) \mid t_0 \leq t \leq t_0 + T - 1\}$. The solution to the dual LP would define the optimal policy of the risk sensitive MDP: For each t such that $t_0 \leq t \leq t_0 + T - 1$, the nonstationary policy is

$$\pi_t(s, a) := \frac{y(t, s, a)}{\sum_{a'} y(t, s, a')}, \forall\, s \in S \text{ and } \forall a \in A. \tag{6}$$

B. Receding-horizon attack planning

The receding-horizon model captures the lateral movement of the reconnaissance-exploitation-actions kill chain of an attacker. At each horizon, the attacker intends to map out the locations of IDSs in the network using reconnaissance techniques. Then, the attacker exploits the vulnerability to act and move to the next node. This process iterates until the attacker reaches his target.

At each step t, the attacker solves an MDP with set $S_{\mathsf{IDS},t} \subseteq S$ of nodes equipped with IDSs. We treat these nodes as obstacles which the attacker aims to avoid. Given the MDP (S, A, P, s_t) with IDS placing at $S_{\mathsf{IDS},t} \subseteq S$ and the current state s_t, the reward function is defined as follows:

$$r_{t+k}(s,a) = 0,\ \forall s \in S, \forall a \in A; \forall k \in [t, t+T-1]; \quad (7)$$

and

$$r_{t+T}(s) = \begin{cases} 1 & \text{if } s = s_f; \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

In addition, let sink be an absorbing state with zero reward. The transition function is revised as follows: For each $s \in S$, for each $a \in A$, if $P(s'|s,a) > 0$ and $s' \in S_{\mathsf{IDS},t}$, then $P(\mathsf{sink}|s,a) = 1$. In other words, when the attacker exploits a vulnerability that has a positive probability to reach a node with IDS, then he will reach a sink state with probability one–that is, he is detected.

Remark 1. It is noted that the detection occurs due to the concurrency of actions by the defender and an attacker. If the attacker always knows where the IDSs are in the next moment, then he can avoid these IDSs by either doing nothing or exploits vulnerabilities only on hosts that are not equipped with IDSs. However, randomization and concurrency together create the unknown effects when the attacker exploits.

Remark 2. The length of the planning horizon T is assumed to be fixed. However, in practice, it can depend on the dynamic tempo of the dynamic defense and attacker's computational resources. Future work will consider attackers with bounded rationality [34]. In this paper, we examine one-time interaction, where the attacker does not have enough data to learn the defender's strategy. Adaptive attacker who can learn the defense strategy must collect data from multiple interactions.

This receding-horizon attack planner is described in Alg. 1. It starts with $t = 0$, the attacker scans the network and determines the location $S_{\mathsf{IDS},t}$ of IDSs. Then, the attacker generates the reward function r_{t+k} and r_{t+T} and solves the finite-horizon risk-sensitive MDP and obtain the policy Π^t. The attacker then takes an action a_t from the policy. This process iterates until either the attacker reaches the goal or becomes detected.

Given that the attacker uses an online planner, the performance can be evaluated based on regret. To evaluate this regret, we need to solve the optimal policy of the attacker assuming the attacker knows exactly the sequence of locations for IDSs sampled over his planning horizon. This optimal policy can be obtained from the following MDP as a stochastic shortest path problem, described below.

Algorithm 1: The receding horizon attack planning algorithm

Input: The PAG with initial state s_0 and target s_f. Finite planning horizon T and total attack horizon $T_{\max}$.
Output: π_t at each time step $t = 0 \dots T_{max}$.

(Initialization): $t = 0$.
while $t < T_{max}$ **do**
  Netscan, obtain $S_{\mathsf{IDS},t}$;
  Get rewards r_{t+k}, r_{t+T} from $S_{\mathsf{IDS},t}$ with (7) and (8).
  Solve $\Pi^t = \{\pi_t, \pi_{t+1}, \dots, \pi_{t+T-1}\}$ with (5).
  Take action $a_t \sim \pi_t(s_t)$ to reach s'.
  The IDSs replaced at $S_{\mathsf{IDS},t+1}$.{The network topology changes.}
  if $s' \in S_{\mathsf{IDS},t+1}$ **then**
    Break. {Attacker is detected.}
  else
    With probability p, reach s', $s_{t+1} \leftarrow s'$;
    With probability $1-p$, stay s_t, $s_{t+1} \leftarrow s_t$.
  end if
  if $s_{t+1} = s_f$ **then**
    Break. {Attacker succeed.}
  else
    $t \leftarrow t+1$; {Time increment.}
  end if
end while

Definition IV.1. Given an MDP $G = (S, A, P, s_0)$ and the attacker's goal state s_f, let $[S_{\mathsf{IDS},0}, S_{\mathsf{IDS},1}, \dots, S_{\mathsf{IDS},T_{\max}}]$ be a sequence of sampled subsets of nodes equipped with IDSs over the time horizon $[0, T_{\max}]$. A time-augmented MDP $\tilde{G} = \langle S \times [0,1,\dots,T_{\max}] \cup \{\mathsf{sink}\}, A, \tilde{P}, (s_0,0), \tilde{r} \rangle$ is defined as follows: $S \times [0,\dots,T_{\max}] \cup \{\mathsf{sink}\}$ are the set of states, A is the set of actions, $(s_0, 0)$ is the initial state. The transition function is defined as: For each $t \in [0, T_{\max}-1]$, each $a \in A$, and each $s \in S$, there are four cases:

1) If $s \neq s_f$, $P(s'|s,a) > 0$, $s' \notin S_{\mathsf{IDS},t+1}$ and $s' \neq s$, then let $\tilde{P}((s',t+1)|(s,t),a) = P(s'|s,a)$ and $\tilde{P}((s,t+1)|(s,t),a) = P(s|s,a)$.
2) If $s = s_f$, let $\tilde{P}(\mathsf{sink}|(s,t),a) = 1$, where sink is an absorbing state for any action $a \in A$.
3) If $P(s'|s,a) > 0$, $s' \in S_{\mathsf{IDS},t+1}$, and $s \neq s'$, then let $\tilde{P}(\mathsf{sink}|(s,t),a) = 1$.
4) If $t = T_{\max}$, $\tilde{P}(\mathsf{sink}|(s,T_{\max}),a) = 1$.

The reward function is defined by

$$\tilde{r}(s,t) = \mathbf{1}(s \equiv s_f). \quad (9)$$

Let $\tilde{\pi}^*$ be the optimal solution of $\tilde{G}$ that maximizes the following objective:

$$J((s_0,0),\tilde{\pi}) = E^{(s_0,0),\tilde{\pi}}\left[\exp\left(\lambda \sum_{n=0}^{h} r((s,n),a_n)\right)\right] \quad (10)$$

where h is the first time when the policy-induced chain reaches the sink state.

Let $\Pi^0 = [\pi_0, \pi_1, \dots \pi_h]$ be the sequence of policies performed by the attacker using the receding horizon planning with a finite horizon h. Based on the solution of time-augmented MDP $\tilde{G}$, we can compute the dynamic regret:

$$\mathcal{R}(\Pi^0) = \|J((s_0, 0), \Pi^0) - J((s_0, 0), \tilde{\pi}^*)\|, \qquad (11)$$

where $J((s_0, 0), \Pi^0)$ is the evaluation of the policy in the time-augmented MDP $\tilde{G}$, $J((s_0, 0), \tilde{\pi}^*)$ is the reward that can be obtained in the finite horizon by executing optimal policy $\tilde{\pi}^*$. The dynamic regret captures the performance difference of policy Π^0 and optimal policy $\tilde{\pi}^*$. We will use dynamic regret to analyze the performance of the defense strategy. The proposed attack planning algorithm does not learn and predict the changes in the network. Thus, it does not minimize the dynamic regret. In the future work, we will consider online attack learning-based planning with regret minimization.

V. Experiments

A. Experimental setup

We implement the proactive defense strategy in a synthetic network and the proposed attack planning algorithm to evaluate how effective the defense strategy is. All experiments in this section are performed on a computer equipped with an Intel R Core™ i7-5700HQ and 8GB of RAM running a python 3.6 script on a 64-bit Ubuntu R 18.04 LTS.

The layout of PAG from the synthetic network is shown in Fig. 3. The graph has twenty nodes. Note that the self-loops are omitted in the graph for clarity. The IDSs in the network are sampled using a random sampling process using a uniform distribution from subset $\mathcal{N} = \{0, 12, 2, 8, 1, 13, 15, 10, 9, 5\}$ at every T_r steps, i.e., sampled at $\frac{1}{T_r}$ frequency. When T_r approaches infinity (*i.e.*, 0 frequency), the locations of IDSs do not change. The attacker does not know T_r and recomputes his policy every step. We assume that once the IDSs are selected, the attacker knows their new locations of IDSs after scanning the network. Thus, the analysis using this type of attacker provides a lower bound on the security level of the system, measured by the probability that the attacker can reach the target while avoiding IDSs.

We conduct an experiment to investigate how the effectiveness of the roaming IDSs policy can be influenced by (1) the frequency in re-sampling and (2) the number of IDSs. In the experiment, the number of IDSs in the network ranges from one to five. The frequency of the sampling of the IDSs ranges from zero (i.e., the location of the IDSs never changes) to one (i.e., the locations of the IDSs change every time instant). Table I shows the parameters used in the attacker's receding horizon planning.

B. The frequency of the re-sampling of the IDSs

The experiment results are shown in Fig. 4 and Fig. 5. From Fig. 4, it is observed that the success rate of the attacker reaching the target decreases as the re-sampling frequency increases. The results are intuitive as a higher sampling frequency leads to a higher probability of an attacker reaching an IDS. However, the more frequent shuffle of IDSs may incur overhead costs including traffic delay and disruption. It is also interesting to observe that the success rate of attack when re-sampling at $\frac{1}{5}$Hz is higher than that at a frequency of zero. This is because re-sampling would sometimes free the attacker from a deadlock. For example, when the attacker is at state 0 and the IDS is at state 17, the best strategy for the attacker is to remain put. When the IDSs are being re-sampled every T_r steps, the deadlock is lifted. However, the same observation may not be obtained if the IDSs are located at different nodes initially or the attacker starts with different initial nodes in the network.

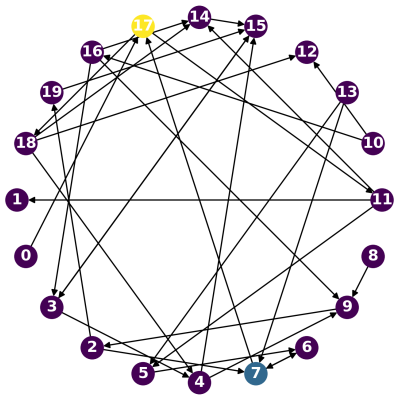


Fig. 3: The layout of the probabilistic attack graph from a synthetic network.

TABLE I: Experiment parameters

Parameters	Values
Finite horizon length T	19
Maximum time length T_{max}	100
Probability of successfully exploit a vulnerability p	0.9
Attacker initial state s_0	17
Attacker target state s_f	7
Discounting factor λ in (2)	1.0

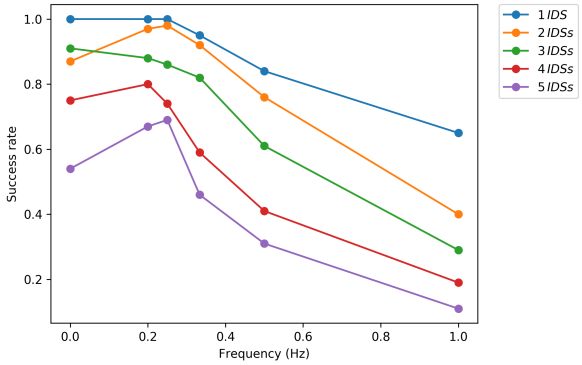


Fig. 4: The effect of the frequency of the re-sampling of the IDSs on the success rate of the attacker.

The choice of sampling locations of IDSs requires game-theoretic reasoning using, for example, resource-allocation games [35], and it will be analyzed in the future work.

C. Number of IDSs

In Fig. 5, we show the experiment results that describe how the number of the IDSs in the network influences the effectiveness of the roaming IDS policy.

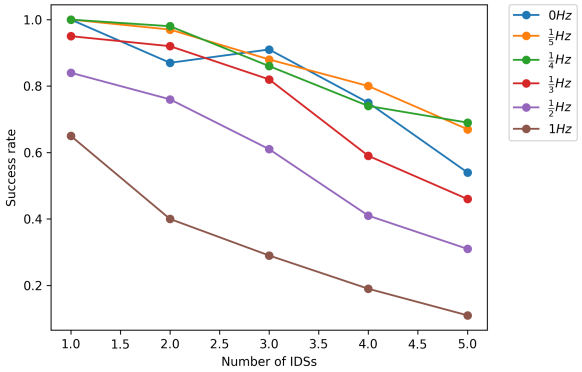


Fig. 5: The effects of the number of IDSs on the success rate of the attacker.

From Fig. 5, it can be seen that the success rate of the attacker reaching the target decreases as the number of the IDSs in the graph increases. It suggests that a higher number of IDSs leads to a more effective roaming IDS policy.

D. The distance to the target

In this experiment, we further evaluate the effect of the distance of the attacker initial state to the target on the MTD policy. Optimal and online policies are computed for ten sequences of random IDS configurations. In each IDS configuration, three IDSs are randomly sampled from the IDS set $\mathcal{N}$ at $\frac{1}{3}Hz$. Evaluation is performed for different initial positions of the attacker, i.e., $s_0 \in \{7, 13, 9, 11, 10, 0, 19\}$ with the distances to the target (measured by the shortest path in the graph) ranging from zero to six, respectively. The final results are show in Fig. 6 and 7. Fig. 6 shows the dynamic regrets computed according to (11) with $h = 19$. Based on the mean value of the regrets, the closer the attacker's initial position is to the target, the smaller the regret is, and hence the less effective the MTD policy is against the attacker. Particularly, when the distance to target is smaller than three, the regret approaches zero and the MTD policy has almost no effect.

Fig. 7 compares the success rate for optimal and online policies at different attacker initial states. From Fig. 7, it can be seen that with the optimal policy, regardless of the distance from the attacker initial node to the target, the attacker can always reach the target with a success rate of 1. On the other hand, with online policy, the attacker's success rate decreases as the distance to the target increases. Chi-squared test is performed on the two-way data set. The data are classified into two mutually exclusive classes: winning when the attacker reaches the target, and losing when the

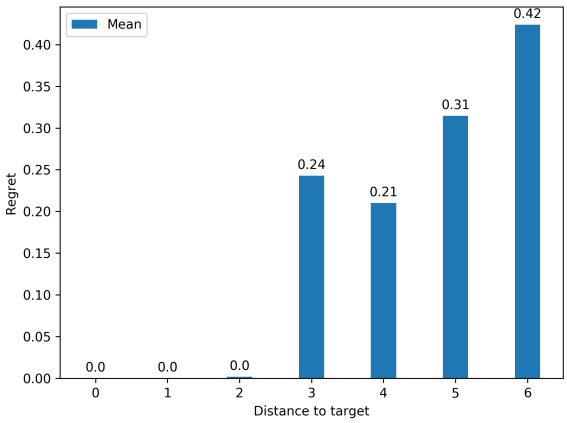


Fig. 6: Dynamic regret analysis

attacker fails to reach the target. The p value is 8.2×10^{-13}, indicating that there is indeed a strong correlation between the success rate and the distance to the target.

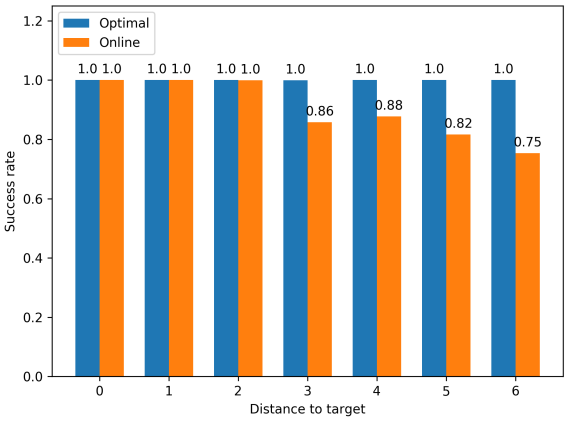


Fig. 7: Success rate analysis

VI. Conclusions and Discussion

In this paper, we have introduced a method to evaluate the effectiveness of a MTD policy to detect the presence of adversaries. Given time varying locations of detection systems in a network, we formulate planning problem for a stealthy attacker using receding horizon framework. The attacker repeatedly performs reconnaissance to figure out where IDSs are placed and solves a risk-sensitive finite-horizon planning problems based on a probabilistic attack graph. We have assessed the effectiveness of the proactive defense strategy using the detection rate in the presence of such an intelligent attacker. This work provides foundations for several future extensions. First, we will investigate an adaptive attacker, who learns the dynamics of the network from past iterations. Several no-regret learning algorithms and online planning in MDPs with regret bounds can be considered for attacker behavior modeling. Second, given the evaluation result, we can construct the game between a defender, who selects subsets of nodes for randomization,

and an intelligent, potentially adaptive attacker. Through game-theoretic reasoning, we can compute optimal detection strategy that trades off multiple objectives, including maximizing the detection rate and minimizing the operational cost.

ACKNOWLEDGMENT

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111990015. This work is also partially supported by grants CNS-1544782, SES-1541164 and ECCS-1847056 from National Science Foundation (NSF), and by award 2015-ST-061-CIRC01, U. S. Department of Homeland Security.

REFERENCES

[1] S. Sengupta, A. Chowdhary, A. Sabur, D. Huang, A. Alshamrani, and S. Kambhampati, "A Survey of Moving Target Defenses for Network Security," *arXiv:1905.00964 [cs]*, May 2019.

[2] S. Jajodia, V. S. Subrahmanian, V. Swarup, and C. Wang, Eds., *Cyber Deception: Building the Scientific Foundation*. Springer International Publishing, 2016.

[3] N. Provos and T. Holz, *Virtual honeypots: from botnet tracking to intrusion detection*. Pearson Education, 2007.

[4] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15*, Jun. 2002, pp. 49–63.

[5] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer, "Foundations of Attack–Defense Trees," in *Formal Aspects of Security and Trust*, ser. Lecture Notes in Computer Science, P. Degano, S. Etalle, and J. Guttman, Eds. Berlin, Heidelberg: Springer, 2011, pp. 80–95.

[6] B. Schneier, "Attack Trees," http://www.schneier.com/paper-attacktrees-ddj-ft.html, Aug. 2007.

[7] J. B. Hong and D. S. Kim, "Assessing the Effectiveness of Moving Target Defenses Using Security Models," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 163–177, Mar. 2016.

[8] J. Hong and D.-S. Kim, "HARMs: Hierarchical Attack Representation Models for Network Security Analysis," in *Australian Information Security Management Conference*. SRI Security Research Institute, Edith Cowan University, Perth, Western Australia, December 2012.

[9] "Network lateral movement from an attacker's perspective," https://searchsecurity.techtarget.com/news/450427135/Network-lateral-movement-from-an-attackers-perspective.

[10] G. A. Ajaeiya, N. Adalian, I. H. Elhajj, A. Kayssi, and A. Chehab, "Flow-based intrusion detection system for sdn," in *2017 IEEE Symposium on Computers and Communications (ISCC)*, July 2017, pp. 787–793.

[11] S. Khattab, C. Sangpachatanaruk, D. Mosse, R. Melhem, and T. Znati, "Roaming honeypots for mitigating service-level denial-of-service attacks," in *24th International Conference on Distributed Computing Systems, 2004. Proceedings.*, Mar. 2004, pp. 328–337.

[12] M. M. Islam, Q. Duan, and E. Al-Shaer, "Specification-driven moving target defense synthesis," in *Proceedings of the 6th ACM Workshop on Moving Target Defense*, ser. MTD'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 13–24. [Online]. Available: https://doi.org/10.1145/3338468.3356830

[13] J. B. Hong and D. S. Kim, "Assessing the effectiveness of moving target defenses using security models," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 163–177, March 2016.

[14] Q. Zhu and T. Başar, "Game-theoretic approach to feedback-driven multi-stage moving target defense," in *Decision and Game Theory for Security*. Springer, 2013, pp. 246–263.

[15] Y. Huang, J. Chen, L. Huang, and Q. Zhu, "Dynamic Games for Secure and Resilient Control System Design," *National Science Review*, 01 2020, nwz218. [Online]. Available: https://doi.org/10.1093/nsr/nwz218

[16] J. Chen, C. Touati, and Q. Zhu, "A dynamic game approach to strategic design of secure and resilient infrastructure network," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 462–474, 2019.

[17] J. Chen and Q. Zhu, "Control of multi-layer mobile autonomous systems in adversarial environments: A games-in-games approach," *IEEE Transactions on Control of Network Systems*, 2019.

[18] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, pp. 1–39, 2013.

[19] L. Huang and Q. Zhu, "A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems," *Computers & Security*, vol. 89, p. 101660, 2020.

[20] Z. Aslanyan, F. Nielson, and D. Parker, "Quantitative Verification and Synthesis of Attack-Defence Scenarios," in *IEEE Computer Security Foundations Symposium (CSF)*, Jun. 2016, pp. 105–119.

[21] R. R. Hansen, P. G. Jensen, K. G. Larsen, A. Legay, and D. B. Poulsen, "Quantitative evaluation of attack defense trees using stochastic timed automata," in *International Workshop on Graphical Models for Security*. Springer, 2017, pp. 75–90.

[22] B. Kordy and W. Wideł, "On Quantitative Analysis of Attack–Defense Trees with Repeated Labels," in *Principles of Security and Trust*, ser. Lecture Notes in Computer Science, L. Bauer and R. Küsters, Eds. Cham: Springer International Publishing, 2018, pp. 325–346.

[23] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer, "Foundations of attack–defense trees," in *International Workshop on Formal Aspects in Security and Trust*. Springer, 2010, pp. 80–95.

[24] C. Baier and J.-P. Katoen, *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

[25] N. Piterman, A. Pnueli, and Y. Sa'ar, "Synthesis of Reactive(1) Designs," in *Verification, Model Checking, and Abstract Interpretation*, ser. Lecture Notes in Computer Science, E. A. Emerson and K. S. Namjoshi, Eds. Berlin, Heidelberg: Springer, 2006, pp. 364–380.

[26] R. Bloem, K. Chatterjee, and B. Jobstmann, "Graph Games and Reactive Synthesis," in *Handbook of Model Checking*, E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, Eds. Cham: Springer International Publishing, 2018, pp. 921–962.

[27] L. Huang and Q. Zhu, "Strategic learning for active, adaptive, and autonomous cyber defense," in *Adaptive Autonomous Secure Cyber Systems*. Springer, 2020, pp. 205–230.

[28] S. Rass and Q. Zhu, "Gadapt: a sequential game-theoretic framework for designing defense-in-depth strategies against advanced persistent threats," in *International Conference on Decision and Game Theory for Security*. Springer, 2016, pp. 314–326.

[29] T. Yadav and A. M. Rao, "Technical aspects of cyber kill chain," in *Security in Computing and Communications*, J. H. Abawajy, S. Mukherjea, S. M. Thampi, and A. Ruiz-Martínez, Eds. Cham: Springer International Publishing, 2015, pp. 438–452.

[30] "Common Vulnerability Scoring System SIG," https://www.first.org/cvss.

[31] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring network security using dynamic bayesian network," in *Proceedings of the 4th ACM Workshop on Quality of Protection - QoP '08*. Alexandria, Virginia, USA: ACM Press, 2008, p. 23.

[32] L. Muñoz-González, D. Sgandurra, M. Barrère, and E. C. Lupu, "Exact inference techniques for the analysis of bayesian attack graphs," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 231–244, 2017.

[33] A. Kumar, V. Kavitha, and N. Hemachandra, "Finite horizon risk sensitive mdp and linear programming," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 7826–7831.

[34] H. A. Simon, "Bounded rationality," in *Utility and probability*. Springer, 1990, pp. 15–18.

[35] D. Korzhyk, V. Conitzer, and R. Parr, "Complexity of computing optimal Stackelberg strategies in security resource allocation games," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.