Synthesis of Proactive Sensor Placement In Probabilistic Attack Graphs

Lening Li[1], Haoxiang Ma[2], Shuo Han[3] and Jie Fu[2]

***Abstract*— This paper studies the deployment of joint moving target defense (MTD) and deception against multi-stage cyber-attacks. Given the system equipped with MTD that randomizes between different configurations, we investigate how to allocate a bounded number of sensors in each configuration to optimize the attack detection rate before the attacker achieves its objective. Specifically, two types of sensors are considered: intrusion detectors that are observable by the attacker and stealthy sensors that are not observable to the attacker. We propose a two-step optimization-based approach for allocating intrusion detectors and stealthy sensors: Firstly, the defender allocates intrusion detectors assuming the attacker will best respond to evade detection by intrusion detectors. Secondly, the defender will allocate stealthy sensors, given the best response attack strategy computed in the first step, to further reduce the attacker's chance of success. We illustrate the effectiveness of the proposed methods using a cyber defense example.**

I. Introduction

This paper considers a game-theoretic design of a proactive cyber defense system using a combination of Moving Target Defense (MTD), intrusion detectors, and deception (with stealthy sensors). Proactive defense means that the defender does not know the attacker's presence or the progress made by the attack but employs randomization to thwart and mitigate attacks. For example, an attack action can fail if the system configuration changes and invalidates the targeted vulnerability. Meanwhile, the defender can deploy sensors to detect the attacker at the early stage of the attack. Nonetheless, with the increasingly advanced MTD [1], detection, and cyber deception [2], it remains a challenge to assess the effectiveness of a combination of MTD and sensor-based detection mechanisms, let alone to design an effective cyber defense system with joint MTD and deception. In this paper, we integrate a formal-method modeling and optimization-based approaches to address the following question: "how to allocate a limited number of (potentially heterogeneous) sensors in this system to maximize the probability of attack detection before that the attacker achieves its objective?" "What is the benefit of employing deceptive, stealthy sensors for proactive defense?"

To model the effects of MTD on the attack performance, we employ a variant of attack graphs [3], [4], which models the causal and logical dependencies between system's vulnerabilities or attacker's subgoals observed in multi-stage attacks. We introduce the Markov Chain as a formal model of a class of MTD in which the defender switches randomly between different system configurations [5], [6], [7]. Given a system equipped with such an MTD strategy, we first capture the attacker's decision-making problem using an Markov decision process (MDP) with a reachability objective; that is, the attacker aims to reach some goal states eventually while evading detection by sensors. For example, the attacker's goal state can be that the attacker gains root access to a critical database server. Then, we focus on the synthesis problem for the defender to minimize the attack success rate by optimally allocating two types of sensors: intrusion detectors that are observable by the attacker and stealthy sensors that are unobservable to the attacker. A stealthy sensor can be realized by honey patching [8] of a known vulnerability. When the attacker exploits a honey-patched vulnerability, he will be detected. We incorporate the attacker's safety constraints, i.e., evading sensor detection, into the attack objective and formulate a bi-level optimization problem. We then design Mixed-Integer Linear Programmings (MILPs)s in a two-step manner to approximately optimize intrusion detectors and stealthy sensors allocation for the defender.

Related work: The synthesis of proactive defense strategies studied herein is closely related to the Stackelberg security game (SSG) (surveyed in [9]). In an SSG, the defender is to defend a set of targets with limited resources, while the attacker selects the optimal attack strategy given the knowledge of the defender strategy. The solution concepts of Stackelberg Equilibrium are employed by [10] to design a mixed strategy for the defender to allocate intrusion detectors and implement the intrusion detectors randomization schedule using MTD. In [11], the authors formulate the security countermeasure-allocation problem as a resource-allocation game, where attack graphs are used to evaluate the security of the network given the allocated resources. A Bayesian attack graph is an empirical attack behavior model constructed from the data and exploitability of the targeted vulnerability [12]. In [13], the authors assume that a Bayesian attack graph [14] represents the attacker's behavior and design optimal defender strategies under partial observations using solutions of partially observable Markov decision processes. Another related formulation is the plan interdiction problem studied in [15], where the attacker is to reach a subset of goals with attack actions, and the defender is to mitigate the attack by interdicting or removing the attack actions. They formulated a mixed-integer programming problem to maximize the defender's objective function assuming the

*Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-22-1-0166.

[1]Lening Li is with Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA lli4@wpi.edu

[2]Haoxiang Ma, Jie Fu are with the Department of Electrical & Computer Engineering, University of Florida, Gainesville, FL, 32611, USA hma2, fujie@ufl.edu

[3]Shuo Han is with the Department of Electrical & Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA hanshuo@uic.edu

optimal plan of the attacker given the interdiction strategy.

In comparison to existing work, we introduce a formal model of MTD strategy and capture the effects of MTD on system configuration randomization as a probabilistic switching between different attack graphs. For allocating intrusion detectors given a randomization schedule, we consider the optimal allocation given a "worst-case" attacker who knows about the MTD schedule and the locations of intrusion detectors and plans to evade detection by intrusion detectors. In addition, we allocate stealthy sensors, which are unobservable to the attacker, to decrease the attack success rate further. To the best of our knowledge, the combined effect of MTD and cyber deception has not been investigated in the literature. This work contributes a formal method-based approach for modeling and synthesizing approximately optimal cyber defense with a class of sensor deception.

II. Problem Formulation

Our modeling of the attack-defend interaction is inspired by the formal graphical security model called *attack graphs*, introduced in [16] for modeling sequential attacks in a network. Specifically, in network security, an attack graph is constructed from the attack actions (vulnerabilities in a program/network) and the pre- and post-conditions of actions.

Besides cybersecurity applications, attack graphs are commonly used for analyzing terrorist networks, counter-terrorism networks, and transportation networks (see a survey in [17]). In this work, though the examples are set with cyber security applications in mind, similar solution approaches are applicable for general security problems modeled using attack graphs.

Definition 1 (Attack Graph)**.** Given a system configuration, the corresponding attack graph is represented as a probabilistic transition system $TS = \langle S, A, T, \nu_0, F\rangle$, where 1) S is a finite set of states, representing security-related attributes of the system and the attacker; 2) A is a finite set of attack actions; 3) $T\colon S \times A \to \mathsf{Dist}(S)$ is a probabilistic transition function that maps a state-action pair into a distribution over next states; 4) ν_0 is the initial state distribution; 5) $F \subseteq S$ is a subset of states. The attacker's objective is to reach one of the states in F.

A *path* $\rho = s_0 a_0 s_1 a_1 \ldots$ of TS is a state-action sequence such that for any $i \geq 0$, there exist $a \in A$, for which $T(s_{i+1} \mid s_i, a) > 0$.

In cybersecurity, an example of a state can be "the attack is at host 1, and host 2 running an `ftp` server". An attack action can be to exploit a known vulnerability on the `ftp` server, to reach a state where "the attacker has user access to host 2." In relation to logical attack graph [18], one can employ PDDL language [19] to generate a (deterministic) transition system from the pre- and post-conditions of exploitation actions in logical attack graphs.

Defender's proactive, randomized moves A defense configuration i describes the network connectivity, node configurations, defensive countermeasures, and the allocation of sensors. It is observed that the changes in system configuration can be directly captured by the changes in the attack graph, including removing/adding transitions or changing the probability distributions given state-action pairs.

Let Γ be the set of indices of different system configurations among which the defender switches. Each configuration $i \in \Gamma$ generates a probabilistic transition system $TS^i = \langle S, A, T^i, \nu_0, F\rangle$. To simplify notations, we assume that different configurations i and j will have different transition functions but share all other components. Note that if the transition systems constructed from two attack graphs of different configurations do not have the same set of states, then we can make the union of the state sets as the set S. The same argument applies to justify the same set of attack actions with different configurations.

Next, we introduce a computational model of proactive defense strategies using randomization.

Definition 2 (Proactive Defense Strategy)**.** A proactive defense strategy is defined by a Markov Chain

$$MC = \langle \Gamma, P, \gamma\rangle,$$

where

- Γ is a finite set of system configurations.
- $P\colon \Gamma \to \mathsf{Dist}(\Gamma)$ is the probabilistic transition function. Given the current configuration i, the probability of reaching configuration j is $P(i, j)$.
- γ is an initial state distribution of configurations.

Defender's Proactive Intrusion Detection with Deceptive Sensors Besides randomization, the defender can allocate sensors to monitor different subsets of states. The defender can block the attacker from the network when a sensor detects an attack.

Specifically, we consider two kinds of sensors: the first kind, called *intrusion detectors*, can be detectable by the attacker; and the second kind, called *stealthy sensors*, cannot be detected by the attacker unless the attacker directly interacts with it. In practice, intrusion detectors are intrusion detection systems or firewalls. Stealthy sensors can be realized by honeypots and honey patching [8]. A honey patch misleads the attacker into believing a specific vulnerability exists. However, such a vulnerability is patched, and exploitation of it will be directly detected by the defender. Honey patching has been recently proposed as an effective detection mechanism using cyber deception.

Definition 3 (Sensor Allocation)**.** The defender's sensor allocation design is a pair of Boolean-valued vectors $(\vec{x}, \vec{y})$, where $\vec{x}, \vec{y} \in \{0, 1\}^{|S\times\Gamma\times A|}$ such that

- $\vec{x}_{s,i,a} = 1$ if and only if under the configuration i, the intrusion detector is placed on state-action pair (s, a).
- $\vec{y}_{s,i,a} = 1$ if and only if under the configuration i, the stealthy sensor is placed on state-action pair (s, a).

It is observed that this modeling of the defender's observation captures realistic sensing modalities. For example, an intrusion detector may only be able to detect one type of

exploitation/action from a given state. A similar argument applies to honey patching, which is used to detect the exploitation of a specific known vulnerability on a target system. Note that the action can still be detected even if the attack fails (with the probabilistic action outcomes).

Assumption 1 (Sensor Allocation Constraints)**.** For any configuration $i \in \Gamma$, the intrusion detector can be allocated to a subset $I \subset S \times A$ of state-action pairs and the stealthy sensor can be allocated to a subset $H \subset S \times A$. The set I and H may have a nonempty intersection. However, if, for configuration $i \in \Gamma$, (s, a) is allocated with the intrusion detector, then it cannot be allocated with a stealthy sensor at the same configuration, and vice versa.

Problem 1. Consider the set of attack graphs $\{TS^i \mid i \in \Gamma\}$ for different system configurations, the set of goal states F, and the defender's randomization schedule modeled as a Markov Chain MC. Assuming the sensor allocation constraints in Assumption 1, compute a sensor allocation strategy given a finite number $h \in \mathbb{Z}$ of intrusion detectors and $k \in \mathbb{Z}$ of stealthy sensors such that the defender can maximize the probability of detecting the attacker before the attacker reaches a goal state in F.

III. A Stackelberg Game Formulation

To formulate the sensor allocation problem, we first construct a model that describes the attacker's interaction with the defense system using randomization but no sensors, then we show how a fixed sensor allocation $(\vec{x}, \vec{y})$ can change such a model to different models perceived by the attacker and the defender.

Assumption 2. It is assumed that the defender and attacker move concurrently. At every time step, the attacker selects an attack action and the defender makes a probabilistic move.

Definition 4 (Attacker's Markov Decision Process without Sensors)**.** Given a proactive defense strategy $MC = \langle \Gamma, P, \gamma \rangle$, a set of probabilistic transitions systems $\{TS^i = \langle S, A, T^i, \nu_0, F \rangle \mid i \in \Gamma\}$ generated from different network configurations, the attacker's planning problem is captured by the MDP:

$$\mathrm{M} = \langle Z, A, \mathcal{P}, \iota, \mathcal{F} \rangle$$

with following components:

- $Z \colon S \times \Gamma$ is the set of states.
- A is the set of attack actions.
- $\mathcal{P} \colon Z \times A \to \mathsf{Dist}(Z)$ is a probabilistic transition function defined as follows. Consider $(s, i), (s', j) \in Z$, for each action $a \in A$, we consider two cases:
  (a) If $T^j(s' \mid s, a) > 0$, then $\mathcal{P}((s', j) \mid (s, i), a) = P(i, j) \cdot T^j(s' \mid s, a)$;
  (b) If $T^j(s' \mid s, a)$ is not defined, then we have $\mathcal{P}((s, j) \mid (s, i), a) = P(i, j)$. In this case, the defense state changes, but no progress is made by the attacker. This is because that the attack action is invalid given the updated configuration.
- ι is the initial state distribution, defined by the joint distribution of initial state distribution ν_0 in the attack graph and the initial state distribution γ of the proactive defense strategy.
- $\mathcal{F} = F \times \Gamma$ is the set of final states which the attacker is to reach.

The probabilistic transition function $\mathcal{P}$ is understood as follows: When the attacker takes an action at the current state, the outcomes of its action will be probabilistic due to the randomized switching of system configurations predefined by the defender's proactive defense strategy and the probabilistic outcome of successfully exploiting the vulnerability. For example, if the system shuffles the IP address, an attack action using the IP address in configuration i will be invalid given the updated system configuration j.

We introduce false negative rates for intrusion detectors as follows.

Assumption 3. Given a state-action pair (s, a), if the attack action a is monitored at state s, then with probability $1 - \epsilon(s, a)$, the attack action will be detected. The value $\epsilon(s, a) \in (0, 1)$ is false negative rate of the detector.

Next, we capture the effects of sensors on the attacker's MDP.

Definition 5 (Attacker's MDP given Incomplete Information about Sensor Allocation)**.** Given a sensor allocation $(\vec{x}, \vec{y})$, the attacker's planning problem is captured by the following MDP:

$$\mathrm{M}^{\vec{x}} = \langle Z, A, \mathcal{P}^{\vec{x}}, \iota, \mathcal{F} \rangle,$$

where $Z, A, \iota, \mathcal{F}$ are the same as those in the MDP without sensors M. The transition function $\mathcal{P}^{\vec{x}}$ is obtained as follows. Consider $(s, i) \in Z$, for each action $a \in A$,

(a) If $T^j(s' \mid s, a) > 0$ and $\vec{x}_{s,j,a} = 0$, then we have $\mathcal{P}^{\vec{x}}((s', j) \mid (s, i), a) = \mathcal{P}((s', j) \mid (s, i), a)$;
(b) If $T^j(s' \mid s, a) > 0$ and $\vec{x}_{s,j,a} = 1$, $\mathcal{P}^{\vec{x}}((s', j) \mid (s, i), a) = P(i, j) T^j(s' \mid s, a) \epsilon(s, a)$, where $\epsilon(s, a)$ is the state-action dependent false negative rate; In words, if the updated configuration has a detector to monitor the exploitation (s, a) but has a false negative rate $\epsilon(s, a)$, then the attacker may reach the next state (s', j) at the chance of a detection failure.
(c) If $T^j(s' \mid s, a)$ is not defined, then $\mathcal{P}^{\vec{x}}((s, j) \mid (s, i), a) = P(i, j)$, which means the defense state changes but no change in the state from the attack graph.
(d) $\mathcal{P}^{\vec{x}}(\mathsf{sink} \mid (s, i), a) = \sum_{j \in \Gamma} P(i, j) \cdot (1 - \epsilon(s, a)) \cdot \vec{x}_{s,j,a}$; In words, the probability of reaching the state sink is the probability of getting detected in a configuration at which the intrusion detector is allocated to monitor state-action pair (s, a).

The defender's model of the attack planning problem, described below, is however different due to the use of stealthy sensors. The following assumption is made.

Assumption 4. A stealthy sensor has a false negative rate of zero.

This assumption is due to the nature of honey patching. It can be relaxed, however, to have false negative rates similar to the treatment for intrusion detector.

Definition 6 (Defender's MDP given Complete Information about Sensor Allocation)**.** Given a sensor allocation $(\vec{x}, \vec{y})$, the defender's model of the attack planning problem is captured by the following MDP:

$$\mathrm{M}^{\vec{x},\vec{y}} = \langle Z, A, \mathcal{P}^{\vec{x},\vec{y}}, \iota, \mathcal{F} \rangle,$$

where $Z, A, \iota, \mathcal{F}$ are the same as those in the MDP without sensors M. Consider $(s, i), (s', j) \in Z$, for each action $a \in A$, the transition function $\mathcal{P}^{\vec{x},\vec{y}}$ is obtained from the transition function $\mathcal{P}^{\vec{x}}$ in the attacker's MDP by letting $\mathcal{P}^{\vec{x},\vec{y}}((s', j) \mid (s, i), a) = \mathcal{P}^{\vec{x}}((s', j) \mid (s, i), a)(1 - \vec{y}_{s,j,a})$; and $\mathcal{P}^{\vec{x},\vec{y}}(\mathsf{sink} \mid (s, i), a) = \sum_{j \in \Gamma} P(i, j)\vec{y}_{s,j,a}$.

Next, we formulate the defender's value function and the attacker's value function respectively.

By the construction of the attacker's MDP, the objective is equivalent to maximizing the probability of reaching the set $\mathcal{F}$, which is a stochastic shortest path problem [20]. The optimal attacker's strategy π^* can be computed by solving the stochastic shortest path problem with the following reward function:

$$R(z) = \begin{cases} 1 & \text{if } z \in \mathcal{F}, \\ 0 & \text{otherwise.} \end{cases}$$

This reward function means that a reward of 1 is received only if the agent reaches a state in $\mathcal{F}$. In this stochastic shortest path problem, the MDP terminates at an absorbing state. The sink state sink and $\mathcal{F}$ are absorbing.

The attacker's perceptual value given the policy π and the attacker's MDP $\mathrm{M}^{\vec{x}}$ is

$$V_2^{\pi}(\iota; \vec{x}) = \mathbf{E}\left[\sum_{k \geq 0} R(z_k) \mid z_{k+1} \sim \mathcal{P}^{\vec{x}}(\cdot \mid z_k, \pi(z_k)), z_0 \sim \iota\right] \quad (1)$$

where the expectation is taken with respect to the stochastic process induced by policy π in $\mathrm{M}^{\vec{x}}$ terminating at absorbing states. That is, $a_k \sim \pi(z_k)$ and $z_{k+1} \sim \mathcal{P}^{\vec{x}}(\cdot \mid z_k, a_k)$, for all $k > 0$.

And for the same policy π, the defender's value is given by

$$V_1^{\pi}(\iota; \vec{x}, \vec{y}) = \mathbf{E}\left[\sum_{k \geq 0} R(z_k) \mid z_{k+1} \sim \mathcal{P}^{\vec{x},\vec{y}}(\cdot \mid z_k, \pi(z_k)), z_0 \sim \iota\right] \quad (2)$$

where the expectation is taken with respect to the stochastic process induced by policy π in $\mathrm{M}^{\vec{x},\vec{y}}$ terminating at absorbing states. Note that the terminating time is perceived differently in the attacker's MDP $\mathrm{M}^{\vec{x}}$ and the defender's MDP $\mathrm{M}^{\vec{x},\vec{y}}$ because the attacker cannot observe the stealthy sensors.

The synthesis of sensor allocation is now formulated as a Stackelberg game, in which the defender designs the allocation, in anticipation of the attacker's best response, in the attacker's MDP with incomplete information.

Problem 2. Let $X \times Y$ be the domains of sensor allocation variables $(\vec{x}, \vec{y})$ under the allocation constraints (Assumption 1). The sensor allocation design is a bi-level optimization problem:

$$\begin{aligned} \min_{(\vec{x},\vec{y}) \in X \times Y} \quad & V_1^{\pi^*}(\iota; \vec{x}, \vec{y}) \\ \text{s.t.} \quad & \pi^* \in \operatorname*{argmax}_{\pi} V_2^{\pi}(\iota; \vec{x}). \end{aligned}$$

The bi-level optimization problem is known to be strongly NP-hard [21]. However, we show that due to the special properties of the sensor allocation problem, an optimal solution can be found by reducing it to two single-level MILP problems. The first one considers optimally allocating intrusion detectors in the absence of stealthy sensors. The second one allocates stealthy sensors given the knowledge of the attacker's best response.

Here, we review Linear Programming (LP) formulation [22] for solving the optimal attack policy. Later, we will show how this LP formulation facilitates the solution of sensor allocation problems.

Let the optimal value vector be defined by $\vec{v}^* = [v_z^*]_{z \in Z}$, where v_z^* is the probability of reaching $\mathcal{F}$ from z under the optimal attack policy. We introduce a decision vector $\vec{v} = [v_z]_{z \in Z}$, where v_z is an upper bound on v_z^* for each $z \in Z$. Consider the following LP:

$$\min_{\vec{v}} \quad \sum_{z \in V} c_z v_z \quad (3)$$

$$\text{s.t.} \quad v_z \geq \sum_{z' \in Z} \mathcal{P}(z' \mid z, a) v_{z'}, \quad \forall a \in A,\ \forall z \in Z, \quad (4)$$

$$v_z = 0, \quad \forall z \in \{\mathsf{sink}\}, \quad (5)$$

$$v_z = R(z), \quad \forall z \in \mathcal{F}, \quad (6)$$

$$v_z \geq 0, \quad \forall z \in Z \quad (7)$$

where $\vec{c} = [c_z]_{z \in Z}$ is a positive vector, termed as state-relevance weights. The state-relevance weights can be selected to be the initial distribution over the states Z. It is shown in [22] that any vector $\vec{v}$ that satisfies (4) is an upper bound on the optimal value vector $\vec{v}^*$. The objective function is equivalent to minimizing a weighted norm between the upper bound $\vec{v}$ and $\vec{v}^*$, given the weight vector $\vec{c} = [c_z]_{z \in Z}$. The solution $\vec{v}$ is shown to be equal to the optimal value vector $\vec{v}^*$ [22].

From a value function $\vec{v}$, a stochastic attack policy, $\pi : Z \to \mathsf{Dist}(A)$, can be computed as the following equation:

$$\pi(a \mid z) = \exp((\mathcal{Q}(z, a) - v_z)/\mu), \quad (8)$$

where $\mu > 0$ is a customized temperature. As the μ goes to 0, equation (8) recovers hardmax operation. The state-action value function $\mathcal{Q}(z, a)$ is defined by

$$\mathcal{Q}(z, a) = \sum_{z' \in Z} \mathcal{P}(z' \mid z, a) v_{z'}. \quad (9)$$

IV. Synthesizing the (sub)-optimal sensor allocation

A. Step 1: Optimal intrusion detector allocation without stealthy sensors

We first consider the case that the defender only allocates detectors but not stealthy sensors. We propose a mixed integer program to solve the optimal intrusion detector allocation strategy as follows. For clarity, we use $x_{s,i,a}$ and $y_{s,i,a}$ to represent $\vec{x}_{s,i,a}$ and $\vec{y}_{s,i,a}$.

$$\min_{\vec{x}\in\mathcal{X},\vec{v}} \quad \sum_{z\in Z} c_z v_z \tag{10}$$

$$\text{s.t.} \quad v_z \geq \sum_{(s',j)\in Z} \big(\mathcal{P}((s',j)\mid z,a)v_{s',j}(1-x_{s,j,a}) + \mathcal{P}((s',j)\mid z,a)v_{s',j}\cdot x_{s,j,a}\cdot\epsilon(s,a)\big), \tag{11}$$

$$\forall a\in A, \forall z=(s,i)\in Z\setminus(\mathcal{F}\cup\{\mathsf{sink}\}),$$

$$\sum_{(s,a)\in S\times A} x_{s,i,a}\leq k, \quad \forall i\in\Gamma, \tag{12}$$

$$(5), (6), \text{ and } (7),$$

where the domain of variable $\vec{x}$ is $\mathcal{X}$ that restricts the allocation to satisfy the constraints in Assumption 1. When $x_{s,j,a}=1$, the right-hand side of constraint (11) is the value given two cases of the next state: The first case is when the attack action is taken but not detected by the intrusion detector. In this case of detection failure, the attack reaches the next state $z'=(s',j)$ from the current state $z=(s,i)$ by taking action a with a probability obtained by the original probability $\mathcal{P}(z'\mid z,a)$ multiplied with the false negative rate $\epsilon(s,a)$. The second case is when the attack action is taken and detected, the attacker will reach the sink state and the attack terminates. If $x_{s,j,a}=0$, then no intrusion detector is allocated in configuration j to monitor the state-action pair (s,a), then the value is given by $\mathcal{P}((s',j)\mid z,a)v_{s',j}$.

The constraint (11) in the optimization problem is nonlinear due to the product between the variable v_z and the integer variable $x_{s,j,a}$. However, we can introduce new variables to rewrite the problem as an MILP. Note that the constraint (11) is equivalent to

$$v_z\geq\sum_{z'\in Z}\mathcal{P}(z'\mid z,a)w_{z,a,z'}, \quad \forall z\in Z,\ \forall a\in A, \tag{13}$$

where for $z'=(s',j)$,

$$w_{z,a,z'}=\begin{cases} v_{z'}\cdot\epsilon(s,a) & \text{if } x_{s,j,a}=1,\\ v_{z'} & \text{if } x_{s,j,a}=0.\end{cases} \tag{14}$$

Using the big-M method, we can rewrite (14) as the following linear constraints:

$$w_{z,a,z'}-v_{z'}\cdot\epsilon(s,a)\leq M\cdot(1-x_{s,j,a}), \tag{15a}$$

$$w_{z,a,z'}-v_{z'}\cdot\epsilon(s,a)\geq m\cdot(1-x_{s,j,a}), \tag{15b}$$

$$w_{z,a,z'}-v_{z'}\leq M\cdot x_{s,j,a}, \tag{15c}$$

$$w_{z,a,z'}-v_{z'}\geq m\cdot x_{s,j,a}, \tag{15d}$$

where M and m are constants to be defined shortly. When $x_{s,j,a}=1$, the constraints (15a) and (15b) together recover $w_{z,a,z'}=v_{z'}\cdot\epsilon(s,a)$, whereas the constraints (15c) and (15d) become non-binding as long as M and m are chosen appropriately. For this problem, it is not difficult to verify that it suffices to choose $M=1$ and $m=-1$. A similar argument can be made for the case when $x_{s,j,a}=0$. The final form of the MILP is given as follows:

$$\min_{\vec{x}\in\mathcal{X},\vec{v}} \quad \sum_{z\in Z} c_z v_z$$

$$\text{s.t.} \quad (5), (6), (7), (12), (13), (15),$$

$$w_{z,a,z'}\geq 0, \quad \forall z\in Z,\ \forall a\in A,\ \forall z'\in Z.$$

B. Step 2: Optimal stealthy sensor allocation for a fixed detector allocation

Next, we allocate a bounded number of stealthy sensors given the attacker's policy π^*, calculated from the attacker's MDP $\mathrm{M}^{\vec{x}}$. In addition, we introduce decision variables $\vec{v}=[v_z]_{z\in Z}$, where v_z is the optimal attack success rate given both intrusion detector and stealthy sensors and new decision variables $\vec{q}=[q_{z,a,z'}]_{(z,a,z')\in Z\times A\times Z}$. We propose another MILP for computing the optimal stealthy sensor allocation strategy:

$$\min_{\vec{q},\vec{v},\vec{y}\in\mathcal{Y}} \quad \sum_{z\in Z} c_z v_z \tag{16}$$

$$\text{s.t.} \quad v_z=\sum_{(a,z')\in A\times Z} q_{z,a,z'}, \quad \forall z\in Z, \tag{17}$$

$$q_{z,a,z'}\leq M\cdot(1-y_{s,i,a}), \tag{18}$$

$$\mathcal{P}(z'\mid z,a)\pi^*(z,a)v_{z'}-q_{z,a,z'}\geq m\cdot y_{s,i,a}, \tag{19}$$

$$\mathcal{P}(z'\mid z,a)\pi^*(z,a)v_{z'}-q_{z,a,z'}\leq M\cdot y_{s,i,a}, \tag{20}$$

$$q_{z,a,z'}\geq 0, \tag{21}$$

$$\forall a\in A, \forall z=(s,i)\in Z, \forall z'=(s',j)\in Z,$$

$$\sum_{s,a} y_{s,i,a}\leq h, \quad \forall i\in\Gamma, \tag{22}$$

$$\text{and} (5), (6), (7),$$

where $M=1$ and $m=-1$ are constants. The domain of variable $\vec{y}$ is $\mathcal{Y}$ that restricts the allocation to satisfy the constraints in Assumption 1. For this optimization problem, we aim to minimize the weighted sum of attack success rate $\vec{v}$ in (16). Note that if the weights $\vec{c}=[c_z]_{z\in Z}$ are chosen to be the initial state distribution, the objective function in (16) is equivalent to minimizing the attack success rate given the initial distribution.

Constraint (17) enforces that the state value v_z is the summation over state-action-state value $q_{z,a,z'}$ for all actions $a\in A$ and next states $z'\in Z$. Constraint (18) means that if $y_{s,i,a}=1$, then the state-action-state value $q_{z,a,z'}=0$ as the attacker will be detected. If $y_{s,i,a}=0$, constraints (19) and (20) enforce

$$\mathcal{P}(z'\mid z,a)\pi^*(z,a)v_{z'}=q_{z,a,z'}. \tag{23}$$

Substituting $q_{z,a,z'}$ into (17), we have policy evaluation of π^* given the stealthy sensors and intrusion detectors allocation.

In the end, we consider finite number of stealthy sensors constrained by inequality (22). Constraint (21) means the state-action-state values are non-negative.

V. Case Study

To illustrate the effectiveness of the proposed method, we consider an example of a cyber system shown in Fig. 1 inspired by [12]. The system has three hosts: the workstation h_1 handles users' requests, the webserver h_2 handles web service requests, and the database server h_3 houses critical data such as personal credentials. In addition, there are a few network security functions, such as firewall, intrusion detectors, and stealthy sensors available to be deployed in the network. The firewall divides hosts into hosts that internal entities can access and hosts that outside entities can access. In this example, h_1 and h_2 can be accessed by outside entities, and h_3 can only be accessed by internal entities. The attacker is initially outside the network system, and the goal is to acquire root privilege on host h_3.

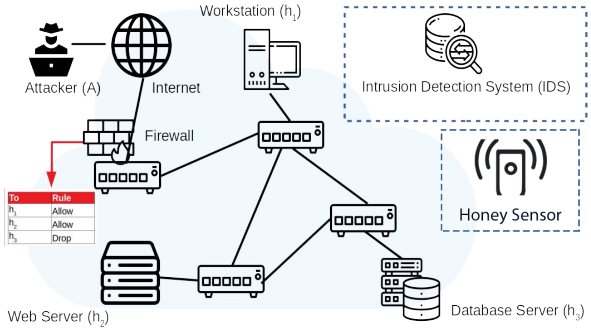

Fig. 1: Network example.

We equip this network with a proactive redundancy-based MTD strategy; that is, we have replicas of Operating System (OS) for network components, and the network configuration is updated dynamically. More specifically, the hosts 1 and 2 probabilistically switch between default OSs and backup OSs [1] This proactive MTD strategy is captured by a Markov Chain shown in Fig. 2.

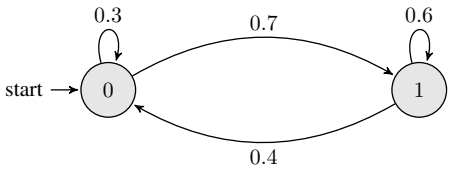

Fig. 2: Two-state proactive MTD strategy.

The Markov Chain can be understood as follows: at the state 0, the network MTD controller either switches to backup OSs with probability 0.7 or stay with the default OSs with probability 0.3; at the state 1, the network system switch back to default OSs with a probability 0.4 or stay with the backup OSs with probability 0.6. In this example, the finite defender states 0 and 1 have one-to-one mappings to the set of network configurations (default and backup).

For each network configuration, we generate its corresponding host-based attack graphs [4] based on the vulnerabilities from Common Vulnerability Scoring System (CVSS) [23]. Note that state (h_2, user) is not reachable in the attack graph for state 0 and thus omitted from the figure. Given the attacker's objective is to reach root privilege in host h_3, the set of goal states in the attack graphs is $\{(h_3, \text{root})\}$ for both attack graphs. The set of final states in the attacker's planning problem (Def. 4) is $\{((h_3, \text{root}), 0), ((h_3, \text{root}), 1)\}$. To illustrate the attack planning problem, we plot a fragment of the attacker's MDP in Fig. 3. The initial state is $(\text{A}, 0)$, and the attacker can take action w_1 to reach state $((h_1, \text{user}), 0)$ with probability 0.063, which is calculated based on the product of three quantities: 1) the probability of staying in configuration 0 (0.3); 2) the probability of exploiting the vulnerability w_1 successfully (0.7); 3) the false negative rate $\epsilon = 0.3$ for the intrusion detector is deployed in $((h_1, \text{user}), 0, w_1)$ but missed the detection. We assume for each state except for the target $(h3, root)$, for each attack action, an intrusion detector or a stealthy sensor can be allocated to monitor that state-action pair.

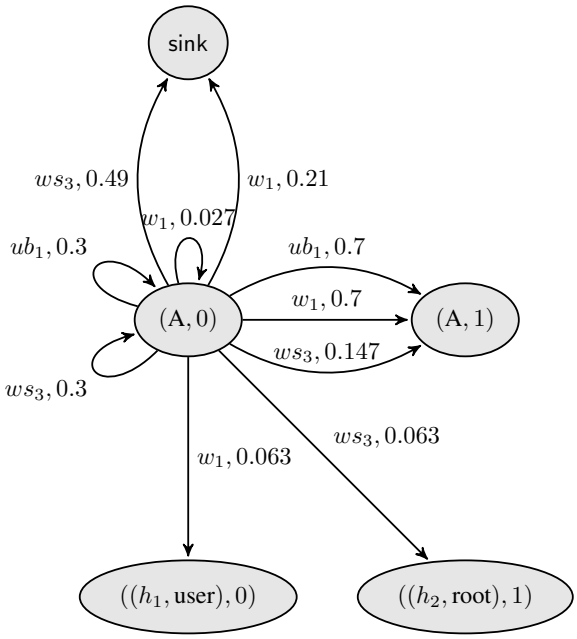

Fig. 3: A fragment of MDP constructed from Def 4.

In the first step, we solve the optimal intrusion detector allocation problem, with varying upper bounds on the number of deployable intrusion detectors and varying false negative rates. We assume the same false negative rates for all intrusion detectors to illustrate how the false negative rate affects the effectiveness of defense. Note that the algorithm allows different intrusion detectors with different false negative rates. Fig 4 summarizes the results. When the false negative rate is fixed, the attack success rates are monotone and non-increasing as the number of intrusion

[1]The information about the default OSs and backup OSs, along with all the vulnerabilities, i.e., attack actions, can be found in `https://bit.ly/3xWxdDa`.

detectors increases. That means with the more intrusion detectors the system can deploy, the attacker has less chance to achieve the target because although he observes intrusion detectors, due to the randomization it cannot always evade intrusion detectors. When the number of intrusion detectors is fixed, the success rates are monotone and non-increasing as the false negative rate decreases. When the false negative rate $\epsilon = 0.3$ and the number of intrusion detectors is 4, intrusion detectors should be placed at $\{(\mathrm{A}, r_1), (\mathrm{A}, w_1), ((h_1, \mathrm{root}), b_3), ((h_2, \mathrm{root}), b_3)\}$ at state 0 and $\{(\mathrm{A}, ws_3), ((h_1, \mathrm{user}), b_3), ((h_2, \mathrm{root}), b_1), ((h_2, \mathrm{root}), b_3)\}$ at state 1.

When the number of intrusion detectors is 1, for all ϵ ranging from 0 to 0.5, we show that the attacker can reach the target state (h_3, root) with probability 1. The solution suggests placing intrusion detectors at $((h_2, \mathrm{root}), b_3)$ at 0 and $((h_1, \mathrm{user}), b_3)$ at 1, but one intrusion detector at each configuration is not sufficient to block alternative attack actions. For example, when the configuration is at 0 and the attacker reaches the state (h_2, root), the intrusion detector is located at $((h_2, \mathrm{root}), b_3)$, but the attacker take action b_1 to reaches the target with probability 1.

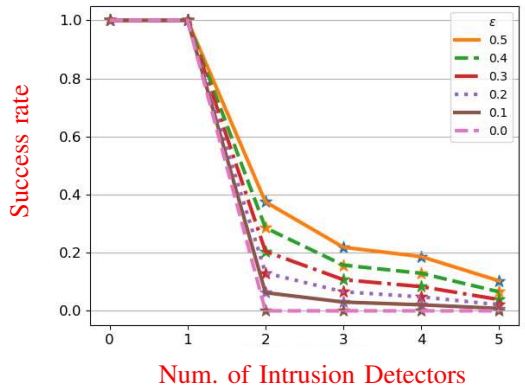

Fig. 4: The number of intrusion detectors versus the attack success rates under different false negative rates ϵ.

After solving the optimal intrusion detector allocation, we synthesize the optimal stealthy sensor allocation strategy. We first extract the optimal attacker's policy according to (8) and (9), where the temperature μ is 0.1. We vary the number of intrusion detectors and the number of stealthy sensors and fix the false negative rate $\epsilon = 0.3$. Fig. 5 summarizes the attack success rates and indicates that, if we fix the number of intrusion detectors and the corresponding policy, the success rates are monotone and non-increasing as the number of stealthy sensors increases.

Furthermore, we compare two cases with a false negative rate $\epsilon = 0.3$: (a) one intrusion detector and one stealthy sensor; (b) two intrusion detectors. For case (a), the attack success rate is 0.167; for case (b), the attack success rate is 0.205. This comparison shows that, for the same number of sensors, deploying stealthy sensors is more effective (with 18.5% reduction in the attack success rate) because first, the

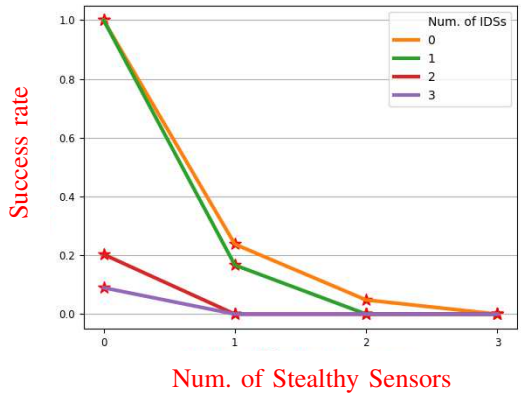

Fig. 5: The number of stealthy sensors versus the attack success rates, where the number of intrusion detectors is $k \in \{0, 1, 2, 3\}$, and false negative rate is $\epsilon = 0.3$.

stealthy sensor has zero false negative rate, and second, the attacker cannot observe these stealthy sensors and plan to evade them. We consider a case when false negative rate 0.3, and there are 2 intrusion detectors and 1 stealthy sensor available. The solution suggests we deploy intrusion detectors at $\{(\mathrm{A}, w_1), (\mathrm{A}, r_1)\}$ and stealthy sensor at $\{(\mathrm{A}, r_1)\}$ at 0; we deploy intrusion detectors at $\{(\mathrm{A}, w_1), (\mathrm{A}, ws_3)\}$ and stealthy sensor at $\{(\mathrm{A}, r_1)\}$ at 1 [2].

The MILPs are solved using the Python-MIP package with Gurobi 9.1.2 on a Windows 10 machine with Intel(R) Xeon (R) E5-1607 v3 CPU and 16 GB RAM. The average computational time of intrusion detectors allocation is 0.72 s, and the average computational time of stealthy sensors allocation is 0.58 s.

VI. Conclusions

For an attacker compromising a cyber system equipped with a proactive MTD mechanism, we developed a formal method-based modeling framework to capture the attacker's planning problems and synthesis algorithms for optimally allocating sensors that minimize the attack success rate. We specifically considered two types of sensors: intrusion detectors that are observable to the attacker and stealthy sensors that are not observable to the attacker. The experiment results demonstrate the combined benefit of MTD, intrusion detection, and deception. In our future work, the following extensions will be investigated: First, our current formulation to allocate stealthy sensors assumes that the attacker is unaware of the use of cyber deception. It remains open to investigate the design of stealthy sensor allocation given deception-aware attacker. Second, the current formulation considers one-time interaction. To mitigate persistent attackers, one must consider that the attacker may learn the deployment of stealthy sensors from past interactions and improve its

[2]We provide the constructed attack graphs and solutions for all intrusion detectors and stealthy sensor allocations in the following link: `https://bit.ly/3zwHrtm`.

attack policy. Lastly, we assume a powerful attacker who can observe the defender's states. In practice, if the defender's states are different from network configurations, then the attacker may not be able to construct the defender's MTD strategy or observe partially the states in the attack planning problem. It is of practical interest to investigate the sensor allocation against attackers with partially observations.

REFERENCES

[1] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," in *Proceedings of the First ACM Workshop on Moving Target Defense*, pp. 31–40, 2014.

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

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

[4] R. Hewett and P. Kijsanayothin, "Host-centric model checking for network vulnerability analysis," in *2008 Annual Computer Security Applications Conference (ACSAC)*, pp. 225–234, IEEE, 2008.

[5] 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 - MTD'19*, pp. 13–24, ACM Press, 2019.

[6] E. Al-Shaer, Q. Duan, and J. H. Jafarian, "Random host mutation for moving target defense," in *International Conference on Security and Privacy in Communication Systems*, pp. 310–327, Springer, 2012.

[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, pp. 163–177, Mar. 2016.

[8] F. Araujo, K. W. Hamlen, S. Biedermann, and S. Katzenbeisser, "From patches to honey-patches: Lightweight attacker misdirection, deception, and disinformation," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 942–953, 2014.

[9] A. SINHA, F. FANG, B. AN, C. KIEKINTVELD, and M. TAMBE, "Stackelberg security games: Looking beyond a decade of success," *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18),Stockholm, Sweden, July 13-19*, pp. 5494–5501, July 2018.

[10] S. Sengupta, A. Chowdhary, D. Huang, and S. Kambhampati, "Moving Target Defense for the Placement of Intrusion Detection Systems in the Cloud," in *Decision and Game Theory for Security* (L. Bushnell, R. Poovendran, and T. Başar, eds.), vol. 11199, pp. 326–345, Cham: Springer International Publishing, 2018.

[11] T. H. Nguyen, M. Wright, M. P. Wellman, and S. Singh, "Multistage Attack Graph Security Games: Heuristic Strategies, with Empirical Game-Theoretic Analysis," *Security and Communication Networks*, vol. 2018, pp. 1–28, Dec. 2018.

[12] S. Yoon, J.-H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, "Attack Graph-based Moving Target Defense in Software-Defined Networks," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2020.

[13] E. Miehling, M. Rasouli, and D. Teneketzis, "Optimal Defense Policies for Partially Observable Spreading Processes on Bayesian Attack Graphs," in *Proceedings of the Second ACM Workshop on Moving Target Defense - MTD '15*, (Denver, Colorado, USA), pp. 67–76, ACM Press, 2015.

[14] M. Frigault and L. Wang, "Measuring Network Security Using Bayesian Network-Based Attack Graphs," in *2008 32nd Annual IEEE International Computer Software and Applications Conference*, pp. 698–703, July 2008.

[15] 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, 2013.

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

[17] D. Ionita, M. Ford, A. Vasenev, and R. Wieringa, "Graphical modeling of security arguments: current state and future directions," in *International Workshop on Graphical Models for Security*, pp. 1–16, Springer, 2017.

[18] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 336–345, 2006.

[19] A. E. Gerevini, P. Haslum, D. Long, A. Saetti, and Y. Dimopoulos, "Deterministic planning in the fifth international planning competition: Pddl3 and experimental evaluation of the planners," *Artificial Intelligence*, vol. 173, no. 5-6, pp. 619–668, 2009.

[20] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[21] P. Hansen, B. Jaumard, and G. Savard, "New branch-and-bound rules for linear bilevel programming," *SIAM Journal on scientific and Statistical Computing*, vol. 13, no. 5, pp. 1194–1217, 1992.

[22] D. P. De Farias and B. Van Roy, "The linear programming approach to approximate dynamic programming," *Operations research*, vol. 51, no. 6, pp. 850–865, 2003.

[23] "CVSS v3.1 Specification Document." https://www.first.org/cvss/specification-document.