A Game-Theoretic Approach to IP Address Randomization in Decoy-Based Cyber Defense

Andrew Clark[1], Kun Sun[2], Linda Bushnell[3], and Radha Poovendran[3]

[1] Dept. of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, 01609, USA. Email: aclark@wpi.edu

[2] Dept. of Computer Science, College of William and Mary, Williamsburg, VA, 23187, USA. Email: ksun@wm.edu

[3] Network Security Lab, Dept. of Electrical Engineering, University of Washington, Seattle, WA, 98195, USA. Email: {lb2,rp3}@uw.edu *

Abstract. Networks of decoy nodes protect cyber systems by distracting and misleading adversaries. Decoy defenses can be further enhanced by randomizing the space of node IP addresses, thus preventing an adversary from identifying and blacklisting decoy nodes over time. The decoy-based defense results in a time-varying interaction between the adversary, who attempts to identify and target real nodes, and the system, which deploys decoys and randomizes the address space in order to protect the identity of the real node. In this paper, we present a game-theoretic framework for modeling the strategic interaction between an external adversary and a network of decoy nodes. Our framework consists of two components. First, we model and study the interaction between the adversary and a single decoy node. We analyze the case where the adversary attempts to identify decoy nodes by examining the timing of node responses, as well as the case where the adversary identifies decoys via differences in protocol implementations between decoy and real nodes. Second, we formulate games with an adversary who attempts to find a real node in a network consisting of real and decoy nodes, where the time to detect whether a node is real or a decoy is derived from the equilibria of the games in first component. We derive the optimal policy of the system to randomize the IP address space in order to avoid detection of the real node, and prove that there is a unique threshold-based Stackelberg equilibrium for the game. Through simulation study, we find that the game between a single decoy and an adversary mounting timing-based attacks has a pure-strategy Nash equilibrium, while identification of decoy nodes via protocol implementation admits only mixed-strategy equilibria.

1 Introduction

Cyber systems are increasingly targeted by sophisticated attacks, which monitor the system over a period of time, identify vulnerabilities, and mount efficient and

* This work was supported by ARO grant W911NF-12-1-0448

effective attacks that are tailored to those vulnerabilities. An emerging approach to thwarting such attacks is through a *moving target defense*, which proactively varies the system protocol, operating system, and software configurations over time, thus rendering vulnerabilities observed by the adversary obsolete before the attack takes place.

One class of moving target defense consists of networks of virtual nodes, which are created and managed by the system and include both real nodes that implement services such as web servers and databases, as well as decoy nodes whose only purpose is to mislead the adversary [18]. If the real and decoy nodes have valid IP addresses that are visible to an external adversary, then the adversary may mount attacks on decoy nodes instead of the real node, wasting the resources of the adversary and providing information to the system regarding the goals and capabilities of the adversary. In order to maximize the probability that the adversary interacts with a decoy node instead of a real node, the decoy nodes should outnumber the real nodes in the network. When the number of decoys is large, however, the amount of memory and CPU time that can be allocated to each decoy is constrained, thus limiting the performance and functionality of each decoy.

While limiting the functionality of decoy nodes reduces their memory and processing cost, it also enables the adversary to detect decoys by observing deviations of the timing and content of node responses from their expected values [16]. Once a decoy node has been detected, its IP address is added to the adversary's blacklist and the decoy is not contacted again by the adversary. By querying and blacklisting decoy nodes over a period of time, the adversary can eventually eliminate all decoys from consideration and mount attacks on the real node. The time required to blacklist the decoy nodes depends on the amount of time needed to identify a node as real or a decoy, which is a function of the resources given to each decoy.

The effectiveness of decoy-based defenses can be further improved by periodically randomizing the IP address space [3]. IP randomization renders any blacklist obsolete, effectively forcing the adversary to re-scan all network nodes. This randomization, however, will also terminate higher-layer protocols such as TCP on the real nodes, which depend on a stable IP address and must be reestablished at a cost of extra latency to valid users [1]. Randomization of the IP address space should therefore be performed based on a trade-off between the performance degradation of valid users and the security benefit of mitigating attacks.

The security benefit of IP randomization and decoy-based defenses depends on the behavior of the adversary. The ability of the decoy nodes to mislead the adversary is determined by the adversary's strategy for detecting decoy nodes. Similarly, frequent IP randomization increases the latency of real users and hence is only warranted when the adversary scans a large number of nodes. Modeling and design of address randomization in decoy-based defenses should therefore incorporate the strategic interaction between an intelligent adversary and the system defense. Currently, however, no such analytical approach exists.

In this paper, we present a game-theoretic framework for modeling and design of decoy-based moving target defenses with IP randomization. Our modeling framework has two components, namely, the interaction between a single virtual node (real or decoy) and an adversary attempting to determine whether the node is real or a decoy, as well as the interaction between an adversary and a network of virtual nodes. These two components are interrelated, since the equilibria of the interaction games between a single virtual node and an adversary determine the time required for an adversary to detect a decoy node, and hence the rate at which an adversary can scan the network and identify real nodes. We make the following specific contributions:

- We develop game-theoretic models for two mechanisms used by adversaries to detect decoy nodes. In the timing-based mechanism, the adversary exploits the increased response times of resource-limited decoy nodes to detect decoys. In the fingerprinting-based mechanism, the adversary initiates a communication protocol with a node and, based on the responses, determines whether the node has fully implemented the protocol, or is a decoy with a partial implementation of the protocol.
- In the case of timing-based detection of a single decoy, we formulate a two-player game between an adversary who chooses the number of probe messages to send and a system that chooses the response time of the decoy subject to resource constraints. The utility of the system is equal to the total time spent by the adversary to query the network. We develop an efficient iterative procedure that converges to a mixed-strategy Nash equilibrium of the game.
- We present a game-theoretic model of decoy detection via protocol fingerprinting, in which we introduce protocol finite state machines as a modeling methodology for decoy detection. Under our approach, the system decides which states to implement, while the adversary attempts to drive the protocol to a state that has not been implemented in order to detect the decoy. We introduce algorithms for computing Nash equilibria of this interaction, which determine the optimal number of high- and low-interaction decoy nodes to be deployed.
- At the network level, we formulate a two-player Stackelberg game, in which the system (leader) chooses an IP address randomization policy, and the adversary (follower) chooses a rate at which to scan nodes after observing the randomization policy. We prove that the unique Stackelberg equilibrium of the game is achieved when both players follow threshold-based strategies. For the attacker, the trade-off is between the cost of scanning and the benefit of identifying and attacking the real node.
- We investigate the performance of the system under our framework through simulation study. For the timing-based game, we find that a pure strategy Nash equilibrium exists in all considered cases. For the fingerprinting game, we compute a mixed-strategy equilibrium, implying that at equilibrium the system should contain both high-interaction nodes that imple-

ment the full protocol and low-interaction nodes that only implement a subset of protocol states.

The paper is organized as follows. We discuss related work in Section 2. The system and adversary models are presented in Section 3. Our game-theoretic formulation for the interaction between the adversary and a single decoy node is given in Section 4. The interaction between an adversary scanning the decoy network and the system deciding when to randomize is considered in Section 5. Simulation results are contained in Section 6. Section 7 concludes the paper.

2 Related Work

Moving target defense is currently an active area of research aimed at preventing adversaries from gathering system information and launching attacks against specific vulnerabilities [13]. Moving target defense mechanisms in the literature include software diversity [9] and memory address layout randomization [10]. These approaches are distinct from decoy generation and IP address randomization and hence are orthogonal from our line of work.

Decoy networks are typically created using network virtualization packages such as honeyd [17]. Empirical studies on detection of decoys have focused on protocol fingerprinting, by identifying differences between the protocols simulated by decoys and the actual protocol specifications, including differences in IP fragmentation and implementation of TCP [11, 22]. Decoy nodes can also be detected due to their longer response times, caused by lack of memory, CPU, and bandwidth resources [16]. The existing studies on decoy networks, however, have focused on empirical evaluation of specific vulnerabilities of widely-used decoy systems, rather than a broader analytical framework for design of dynamic decoy networks.

IP address space randomization has been proposed as a defense against scanning worms [3, 1]. In [21], a framework for deciding when to randomize the IP address space in the presence of hitlist worms, based on a given estimate of whether the system is in a secure or insecure state, was proposed. A decision-theoretic approach to IP randomization in decoy networks was recently presented in [8], but this approach was concerned with the optimal system response to a given adversary strategy rather than the interaction between an intelligent adversary and the system. Furthermore, the work of [8] only considered timing-based attacks on decoy networks, and did not consider fingerprinting attacks.

Game-theoretic techniques have been used to model and mitigate a variety of network security threats [2]. A dynamic game-theoretic approach to designing a moving target defense configuration to maximize the uncertainty of the adversary was proposed in [26]. The method of [26], however, does not consider the timing of changes in the attack surface, and hence is complementary to our approach. The FlipIt game was formulated in [24] to model the timing of host takeover attacks; the FlipIt game does not, however, consider the presence of decoy resources.

In [6], platform randomization was formulated as a game, in which the goal of the system is to maximize the time until the platform is compromised by choosing a probability distribution over the space of available platforms. A game-theoretic approach to stochastic routing, in which packets are proactively allocated among multiple paths to minimize predictability, was proposed in [4]. In [12], game-theoretic methods for spatiotemporal address space randomization were introduced. While these approaches consider metrics such as time to compromise the system that are intuitively similar to our approach, the formulations are fundamentally different and hence the resulting algorithms are not directly applicable to our problem. To the best of our knowledge, game-theoretic approaches for decoy-based moving-target defenses are not present in the existing literature.

3 Model and Preliminaries

In this section, we present the models of the virtual network and the adversary.

3.1 Virtual Network Model

We consider a network consisting of n virtual nodes, including one real node and $(n-1)$ decoy nodes. Let $\pi = \left(1 - \frac{1}{n}\right)$ denote the fraction of nodes that are decoys. Decoy and real nodes have valid IP addresses that are chosen at random from a space of $M \gg n$ addresses, and hence decoy and real nodes cannot be distinguished based on the IP address. The assumption $M \gg n$ ensures that there is sufficient entropy in the IP address space for randomization to be effective. Decoy nodes are further classified as either high-interaction decoys, which implement the full operating system including application-layer services such as HTTP and FTP servers and SQL databases, and low-interaction decoys, which implement only partial versions of network and transport layer protocols such as IP, TCP, UDP, and ICMP [18].

Decoy nodes respond to messages from nodes outside the network. The decoy responses are determined by a configuration assigned to each decoy. Each possible configuration represents a different device (e.g., printer, PC, or server) and operating system that can be simulated by the decoy. Decoy nodes in the same network may have different configurations. Due to limited computation resources assigned to them, decoys will have longer communication delays than real nodes. The additional delay depends on the system CPU time and memory allocated to the decoy. Decoy node configurations can be randomized using software obfuscation techniques [15].

Based on models of service-oriented networks such as web servers, we assume that real nodes receive connection requests from valid users according to an M/G/1 queuing model [5]. Under this model, the service time of each incoming user is identically distributed and independent of both the service times of the other users and the number of users currently in the queue.

Since valid users have knowledge of the IP address of the real node, connections to decoy nodes are assumed to originate from errors or adversarial scanning. Decoy nodes will respond to suspicious, possibly adversarial queries in order to distract the adversary and delay the adversary from identifying and targeting the real node.

The virtual network is managed by a hypervisor, which creates, configures, and removes decoy nodes [7]. The hypervisor is assumed to be trusted and immune to compromise by the adversary. In addition to managing the decoy nodes, the hypervisor also assigns IP addresses to the nodes. In particular, the hypervisor can assign a new, uniformly random IP address to each node at any time. By choosing the new IP addresses to be independent of the previous IP addresses, the hypervisor prevents the adversary from targeting a node over a period of time based on its IP address. All IP addresses are assumed to be randomized simultaneously; generalizations to randomization policies that only update a subset of IP addresses at each time step are a direction for future work. Any communication sessions between valid users and the real node will be terminated when randomization occurs. Upon termination, the server sends the updated IP address to each authorized client. Each valid user must then reconnect to the real node, incurring an additional latency that depends on the connection migration protocol [23].

3.2 Adversary Model

We consider an external adversary with knowledge of the IP address space. The goal of the adversary is to determine the IP address of the real node in order to mount further targeted attacks. The adversary is assumed to know the set of possible IP addresses, if necessary by compromising firewalls or proxies, and attempts to identify the real node by sending query messages to IP addresses within this space. Based on the response characteristics, the adversary can evaluate whether a node is real or a decoy based on either timing analysis or protocol fingerprinting, as described below.

In timing-based blacklisting of nodes, an adversary exploits the response timing differences between real nodes and decoys. Since the decoy nodes have fewer CPU and memory resources than the real node, their response times will be longer. This longer delay can be used for detection. We assume that the adversary knows the response time distribution of a typical real node, which can be compared with response times of possible decoys for detection.

Protocol fingerprinting exploits the fact that the decoy nodes do not actually implement an operating system, but instead simulate an operating system using a prespecified configuration. As a result, differences between the decoys' behavior and the ideal behavior of the operating system allow the adversary to identify the decoy. Typical fingerprints include protocol versions, such as the sequence and acknowledgment numbers in TCP packets, the TCP options that are enabled, and the maximum segment size [25].

4 Modeling Interaction with Single Decoy

In this section, we provide a game-theoretic formulation for the interaction between the adversary and a single decoy node. We present a game-theoretic formulation for two attack types. First, we consider an adversary who attempts to identify decoy nodes through timing analysis. We then model detection based on fingerprinting techniques.

4.1 Timing-Based Decoy Detection Game

In timing-based detection, the adversary sends a sequence of probe packets (such as ICMP echo messages) and observes the delays of the responses from the node [16]. Let Z_k denote the delay of the response to the k-th probe packet. Based on the response times, the adversary decides whether the node is real or a decoy.

We let H_1 denote the event that the response is from a real node and H_0 denote the event that the response is from a decoy. The response times are assumed to be independent and exponentially distributed [16] with mean $\mu_1 = 1/\lambda_1$ for real nodes and $\mu_0 = 1/\lambda_0$ for decoys, where λ_1 and λ_0 represent the response rates of the real and decoy nodes, respectively. Note that the exponential response time is for a single query, while the M/G/1 assumption of Section 3.1 concerns the total length of a session between a valid user and the real node. The number of queries made by the adversary is denoted Q.

The adversary's utility function consists of three components, namely, the amount of time spent querying the node, the probability of falsely identifying a decoy as the real node (false positive), and the probability of falsely identifying the real node as a decoy (false negative). We let P_{FP} and P_{FN} denote the probabilities of false positive and false negative, respectively. The expected time spent querying is equal to $(\pi\mu_0 + (1-\pi)\mu_1)Q$, where π denotes the fraction of decoy nodes.

The action space of the adversary consists of the number of times Q that the virtual node is queried, so that $Q \in \mathbb{Z}_{\geq 0}$. We assume that the adversary makes the same number of queries Q to each node, corresponding to a predesigned, non-adaptive scanning strategy that does not consider feedback from past interactions. The system's action space consists of the mean of the decoy response time $\mu_0 \in [0, \infty)$.

The payoff of the adversary is equal to the total time required to scan the entire network. The expected utility of the adversary is given by

$$\begin{aligned} U_A(Q, \mu_0) = &-(\pi\mu_0 + (1-\pi)\mu_1)Q \\ &- \pi c_{FP} P_{FP}(Q, \mu_0) - (1-\pi) c_{FN} P_{FN}(Q, \mu_0), \end{aligned} \quad (1)$$

where c_{FP} and c_{FN} denote the delays arising from false positive and false negative, respectively. The first term of (1) is the expected time to query a node. The second term is the additional time spent querying decoy nodes after a false positive occurs, which causes the adversary to attempt additional, time-intensive attacks on the decoys. The third term is the additional time spent querying decoy nodes after a false negative, when an adversary mistakes a real node for a decoy and scanning the rest of the network.

The cost of a given response rate is the additional delay experienced by the real nodes. Assuming that requests to the real node occur at rate θ and the network has a total capacity of c with variance σ^2, which is determined by the bandwidth, CPU, and memory constraints of the physical device, this delay is equal to $g(\mu_0) = \frac{\sigma^2\theta}{2(1-\theta/(c-1/\mu_0))} + \frac{1}{c-1/\mu_0}$, based on the assumption that the real node is an M/G/1 system [20, Ch. 8.5] (the M/G/1 assumption follows from the assumption of a single real node; generalization to M/G/m networks with m real nodes is a direction of future work). The payoff of the system is equal to

$$\begin{aligned} U_S(Q,\mu_0) = (\mu_0\pi + (1-\pi)\mu_1)Q + \pi c_{FP}P_{FP}(Q,\mu_0) \\ + (1-\pi)c_{FN}P_{FN}(Q,\mu_0) - g(\mu_0). \end{aligned} \quad (2)$$

The utility of the system is the total time spent by the adversary scanning the network, which increase the security of the real node.

In what follows, we introduce an algorithm for computing the Nash equilibrium of the timing-based interaction game. We first introduce a two-player zero-sum game with equivalent Nash equilibrium strategies. We then prove concavity of the utility functions of each player, implying that a unique equilibrium exists that can be computed using fictitious play.

Proposition 1. *Define the utility function*

$$\begin{aligned} \tilde{U}_A(Q,\mu_0) = -\pi\mu_0 Q - (1-\pi)\mu_1 Q - \pi c_{FP}P_{FP}(Q,\mu_0) \\ - (1-\pi)c_{FN}P_{FN}(Q,\mu_0) + g(\mu_0). \end{aligned} \quad (3)$$

Then a pair of strategies (Q^*,μ_0^*) is a Nash equilibrium for the two-player game between a player 1 with utility function $\tilde{U}_A$ and a player 2 with utility function U_S if and only if it is the Nash equilibrium of a two-player game where player 1 has utility function U_A and player 2 has utility function U_S.

Proof. Let (Q^*,μ_0^*) be a Nash equilibrium for the game with utility functions $\tilde{U}_A$, U_S. The fact that μ_0^* is a best response to Q^* for the game with utility functions U_A and U_S follows trivially from the fact that U_S is the system's utility function in both cases. If Q^* satisfies $\tilde{U}_A(Q^*,\mu_0^*) \geq \tilde{U}_A(Q,\mu_0^*)$ for all $Q > 0$, then

$$\tilde{U}_A(Q^*,\mu_0^*) + g(\mu_0^*) \geq \tilde{U}_A(Q,\mu_0^*) + g(\mu_0^*),$$

and hence $U_A(Q^*,\mu_0^*) \geq U_A(Q,\mu_0^*)$, since $U_A(Q,\mu_0) = \tilde{U}_A(Q,\mu_0) + g(\mu_0)$ for all (Q,μ_0). Thus Q^* is the best response to μ_0^* under utility function U_A. The proof of the converse is similar.

By Proposition 1, it suffices to find a Nash equilibrium of the equivalent zero-sum game with adversary and system utilities $\tilde{U}_A$ and U_S, respectively. As a first step, we prove two lemmas regarding the structure of $\tilde{U}_A$ and U_S.

Lemma 1. *Let $\epsilon > 0$. Then there exists $\hat{Q}$ and a convex function $\hat{f} : \mathbb{R} \rightarrow \mathbb{R}$ such that $|\hat{f}(Q) - \tilde{U}_A(Q,\mu_0)| < \epsilon$ for all $Q > \hat{Q}$.*

Proof. Define $f(Q) = -(\pi\mu_0 + (1-\pi)\mu_1)Q - c_{FP}P_{FP}(Q,\mu_0) - c_{FN}P_{FN}(Q,\mu_0) + g(\mu_0)$. The first two terms are linear in Q and hence convex, while the last term does not depend on Q. In computing the probability of false positive, we first observe that the maximum-likelihood decision rule for the adversary is to decide that the node is real if $\mu_1 c_{FP} P_1(Z_1, \ldots, Z_Q) > \mu_0 c_{FN} P_0(Z_1, \ldots, Z_Q)$ and that the node is a decoy otherwise. Under the exponential assumption, this is equivalent to

$$Q \log \frac{\lambda_1}{\lambda_0} - (\lambda_1 - \lambda_0)\sum_{j=1}^{Q} Z_j > \log \frac{\mu_0 c_{FN}}{\mu_1 c_{FP}}.$$

Hence the probability of false positive is equal to

$$P_{FP}(Q) = Pr\left(Q \log \frac{\lambda_1}{\lambda_0} - (\lambda_1 - \lambda_0)\sum_{j=1}^{Q} Z_j > \log \frac{\mu_0 c_{FN}}{\mu_1 c_{FP}} \big| H_0\right).$$

Rearranging terms yields

$$P_{FP}(Q) = Pr\left(\overline{Z} < \frac{\log \lambda_1 - \log \lambda_0}{\lambda_1 - \lambda_0} - \frac{\log \frac{\mu_0 c_{FN}}{\mu_1 c_{FP}}}{Q(\lambda_1 - \lambda_0)} \big| H_0\right),$$

where $\overline{Z} = \frac{1}{Q}\sum_{j=1}^{Q} Z_j$.

By the Central Limit Theorem, $\overline{Z}$ can be approximated by an $N(\mu_0, \mu_0^2/Q)$-Gaussian random variable for Q sufficiently large. Letting $x = \frac{\log \lambda_1 - \log \lambda_0}{\lambda_1 - \lambda_0}$, the probability of false positive is equal to $Pr(X < \sqrt{Q}(x\lambda_0 - 1))$ where X is an $N(0,1)$-Gaussian random variable, so that

$$P_{FP} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\sqrt{Q}(x\lambda_0 - 1)} \exp\left(-\frac{x^2}{2}\right)\,dx.$$

Differentiating with respect to Q yields

$$\frac{x\lambda_0 - 1}{\sqrt{2\pi}} \frac{1}{2\sqrt{Q}} \exp\left(-\frac{Q(x\lambda_0 - 1)^2}{2}\right),$$

which is increasing in Q since $x\lambda_0 < 1$. Hence the probability of false positive can be approximated by a convex function for Q sufficiently large. The derivation for the probability of false negative is similar.

Approximate concavity of U_A implies that the best response of the adversary can be computed by enumerating the values of $U_A(Q, \mu_0)$ for $Q < \hat{Q}$, and using convex optimization to find the optimal value when $Q \geq \hat{Q}$.

The following lemma establishes concavity of the system utility function U_S as a function of μ_0 for a given T. The concavity of U_S enables efficient computation of the Nash equilibrium.

Lemma 2. *The function U_S is concave as a function of μ_0.*

Proof. It suffices to show that each term of U_S in Eq. (2) is concave. The first term of U_S is linear in μ_0 and therefore concave. The second derivative test implies that $g(\mu_0)$ is convex as a function of μ_0, and hence $-g(\mu_0)$ is concave. By the analysis of Lemma 1, in proving the concavity of the false positive probability, it is enough to show that $Pr\left(X < \frac{x\sqrt{Q}}{\mu_0} - \sqrt{T}\right)$ is concave as a function of μ_0. The derivative of $\frac{x}{\mu_0}$ with respect to μ_0 is equal to

$$\frac{\frac{1}{\mu_0}\left(\frac{\mu_0}{\mu_1} - 1\right) - (\log \mu_0 - \log \mu_1)\left(\frac{1}{\mu_1}\right)}{\left(\frac{\mu_0}{\mu_1} - 1\right)^2},$$

which is decreasing in μ_0. Hence the derivative of the false positive probability is equal to

$$\frac{\frac{1}{\mu_0}\left(\frac{\mu_0}{\mu_1} - 1\right) - (\log \mu_0 - \log \mu_1)\left(\frac{1}{\mu_1}\right)}{\left(\frac{\mu_0}{\mu_1} - 1\right)^2} \exp\left(-\frac{\left(\frac{x\sqrt{Q}}{\mu_0} - \sqrt{Q}\right)}{2}\right)^2,$$

which is monotonically decreasing in μ_0 and hence concave.

Fictitious play can be used to find the Nash equilibrium of the interaction between the adversary and the network. The algorithm to do so proceeds in iterations. At each iteration m, there are probability distributions p_A^m and p_S^m defined by the prior interactions between the system and adversary. The system chooses μ_0 in order to maximize $\mathbf{E}_{p_A}(U_S(\mu_0)) = \sum_Q p_A^m(Q) U_S(Q, \mu_0)$, while the adversary chooses Q to maximize $\mathbf{E}_{p_S^m}(U_A(Q)) = \int_0^\infty p_S^m(\mu_0) U_A(Q, \mu_0)\ d\mu_0$. The strategies of the system and adversary at each iteration can be computed efficiently due to the concavity of U_S and the approximate convexity of U_A. Convergence is implied by the following proposition.

Proposition 2. *The fictitious play procedure converges to a mixed-strategy Nash equilibrium.*

Proof. Since the utility functions satisfy $\tilde{U}_A(Q, \mu_0) + U_S(Q, \mu_0) = 0$, the iterative procedure implies converge to a mixed-strategy Nash equilibrium [19, pg. 297]. Furthermore, by Proposition 1, the mixed-strategy equilibrium is also an NE for the game with utility functions U_A and U_S.

4.2 Fingerprinting-Based Decoy Detection Game

Operating system fingerprinting techniques aim to differentiate between real and decoy nodes by exploiting differences between the simulated protocols of the decoy and the true protocol specifications. In order to quantify the strategies of the adversary and the system, we model the protocol to be simulated (e.g., TCP) as a finite state machine $\mathcal{F}$, defined by a set of states S, a set of inputs I, and a set of outputs O. The transition function $\delta : I \times S \to S$ determines the next state of the system as a function of the input and current state, while the output is determined by a function $f : I \times S \rightarrow O$. We write $\mathcal{F} = (S, I, O, \delta, f)$.

The real and decoy protocols are defined by finite state machines $\mathcal{F}_R = (S_R, I_R, O_R, \delta_R, f_R)$ and $\mathcal{F}_D = (S_D, I_D, O_D, \delta_D, f_D)$. The goal of the decoy protocol is to emulate the real system while minimizing the number of states required. Under this model, the adversary chooses a state $s \in S_R$ and attempts to determine whether that state is implemented correctly in the decoy, i.e., whether the output o corresponding to an input i satisfies $o = f_R(s, i)$. In order to reach state s, the adversary must send a sequence of d_s inputs, where d_s denotes the minimum number of state transitions required to reach the state s from the initial state s_0.

The system's action space is defined by the set of states S_D, while the adversary's action space is the set s that the adversary attempts to reach. The choice of s will determine the sequence of messages sent by the adversary. The adversary's utility function is therefore given by

$$U_A(s, S_D) = -d_S - c_{FP}P_{FP}(s, S_D) - c_{FN}P_{FN}(s, S_D).$$

We note that the real node implements the state s correctly for all $s \in S_R$, and hence the probability of false negative is zero. Furthermore, we assume that the decoy returns the correct output at state s with probability 1 if $s \in S_D$ and returns the correct output with probability 0 otherwise. Hence the adversary's utility function is

$$U_A(s, S_D) = -d_s - \mathbf{1}(s \in S_D)c_{FP}, \tag{4}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

For the system, the utility function is equal to the total time spent by the adversary querying a decoy node, minus the memory cost of the decoys. This utility is equal to

$$U_S(s, S_D) = d_s + \mathbf{1}(s \in S_D)c_{FP} - c_D(S_D), \tag{5}$$

where $c_D(S_D)$ is the cost of implementing a set of states. In order to avoid state space explosion for the system, we restrict the defender to strategies that implement all states within k steps of the initial state, where $k \in \{0, \ldots, |S_D|\}$. Intuitively, a strategy that implements a state $s \in S_D$ but does not implement a state $s' \in S_D$ with $d_{s'} < d_s$ may be suboptimal, because the protocol may reach state s before state s', thus enabling the adversary to identify the decoy in fewer steps.

A fictitious play algorithm for computing a mixed-strategy equilibrium is as follows. Probability distributions π_A^m and π_S^m, which represent the empirical frequency of each strategy of the adversary and system up to iteration m, are maintained. At the m-th iteration, the strategies $k^* = \arg\max \mathbf{E}_{\pi_A^m}(k)$ and $s^* = \arg\max \{\mathbf{E}_{\pi_S^m}(s)\}$ are computed and the corresponding entries of π_A^{m+1} and π_S^{m+1} are incremented. Since there is an equivalent zero-sum game with adversary utility function $\tilde{U}_A(s) = d_s + \mathbf{1}(s \in S_D)c_{FP} - c_D(S_D)$, the empirical frequencies of each player converge to the mixed strategy equilibrium [19].

5 Characterization of Optimal IP Address Randomization Strategy by Network

In this section, we present a game-theoretic formulation for the interaction between the virtual network, which decides when to randomize the IP address space, and the adversary, which decides the scanning strategy. The optimal randomization policy of the network and the probability of detecting the real node at equilibrium are derived.

5.1 Game Formulation

We consider a game in which the adversary chooses a scanning strategy, determined by the number of simultaneous connections α. The parameter α is bounded above by α_{max}, which is chosen by the hypervisor to limit the total number of connections and hence avoid overutilization of the system CPU. The adversary incurs a cost ω for maintaining each connection with a node. The number of nodes scanned by the adversary per unit time, denoted $\varDelta$, is given by $\varDelta = \frac{\alpha}{\tau}$, where τ is the time required to scan each node. The parameter τ depends on the detection method employed by the adversary, and is equal to the Nash equilibrium detection time of Section 4.1 if timing-based detection is used or the Nash equilibrium detection time of Section 4.2 if fingerprint-based detection is used.

At each time t, the system decides whether to randomize the IP address space; we let $t = 0$ denote the time when the previous randomization took place. Let R denote the time when randomization occurs. The system incurs two costs of randomization, namely, the probability that the adversary detects the real node and the number of connections that are terminated due to randomization. Since the real and decoy nodes cannot be distinguished based on IP addresses alone, the probability of detection at time t is equal to the fraction of nodes that are scanned up to time t, $\frac{\varDelta t}{n}$.

The cost resulting from terminating connections is equal to the delay β resulting from migrating each connection to the real node's new IP address; TCP migration mechanisms typically have cost that is linear in the number of connections [23]. The cost of breaking real connections is therefore equal to $\beta Y(t)$, where $Y(t)$ is equal to the number of connections to the real node, so that the utility function of the system is given by $U_S(\alpha, R) = -\mathbf{E}\left(\frac{\alpha}{\tau n}R + \beta Y(R)\right)$.

For the adversary, the utility is equal to the detection probability, minus the cost of maintaining each connection, for a utility function of $U_A(\alpha, R) = \mathbf{E}\left(\frac{\alpha}{\tau n}R - \omega\alpha\right)$. The resulting game has Stackelberg structure, since the system first chooses the randomization policy, and the adversary then chooses a scanning rate based on the randomization policy.

5.2 Optimal Strategy of the System

The information set of the system is equal to the current number of valid sessions $Y(t)$ and the fraction of decoy nodes scanned by the adversary $D(t)$ at time t.

The goal of the system is to choose a randomization time R in order to minimize its cost function, which can be expressed as the optimization problem

$$\begin{array}{l}\text{minimize } \mathbf{E}(D(R) + \beta Y(R)) \\ R\end{array} \tag{6}$$

where R is a random variable. The randomization policy can be viewed as a mapping from the information space $(Y(t), D(t))$ at time t to a $\{0, 1\}$ variable, with 1 corresponding to randomizing at time t and 0 corresponding to not randomizing at time t. Define L_t to be the number of decoy nodes that have been scanned during the time interval $[0, t]$.

The number of active sessions $Y(t)$ follows an M/G/1 queuing model with known arrival rate ζ and average service time $1/\phi$. We let $1/\phi_t$ denote the expected time for the next session with the real node to terminate, given that a time t has elapsed since the last termination. In what follows, we assume that ϕ_t is monotonically increasing in t; this is consistent with the M/M/1 and M/D/1 queuing models. The following theorem, which generalizes [8, Theorem 1] from an M/M/1 to an M/G/1 queuing model, describes the optimal strategy of the system.

Theorem 1. *The optimal policy of the system is to randomize immediately at time t if and only if $L_t = n$, $Y(t) = 0$, or $\frac{\varDelta}{n}\phi + \beta\zeta\phi - \beta > 0$, and to wait otherwise.*

Proof. In an optimal stopping problem of the form (6), the optimal policy is to randomize at a time t satisfying

$$D(t) + \beta Y(t) = \sup \{\mathbf{E}(D(t') + \beta Y(t') | D(t), Y(t)) : t' \geq t\}.$$

If $L_t = n$, then the address space must be randomized to avoid detection of the real node. If $Y(t) = 0$, then it is optimal to randomize since $D(t)$ is increasing as a function of t.

Suppose that $L_t < n$ and $Y(t) > 0$. Let $\xi_1, \xi_2, \ldots$ denote the times when connections terminate. We prove by induction that, for each l, $t' \in [\xi_{l-1}, \xi_l]$ implies that $\mathbf{E}(D(t') + \beta Y(t') | Y(t)) > D(t) + \beta Y(t)$. First, consider $l = 1$, with $\xi_0 = t$. Then if $t' \in [\xi_0, \xi_1)$, $D(t') + \beta Y(t') > D(t) + \beta Y(t)$, since D is nondecreasing in time and no connections have terminated since time t. At time ξ_1, we have that

$$\begin{aligned}\mathbf{E}(D(\xi_1) + \beta Y(\xi_1) | Y(t)) &= \frac{\varDelta}{n}\mathbf{E}(\xi_1) && (7)\\ &\quad +\beta(Y(t) + \zeta\mathbf{E}(\xi_1) - 1) && (8)\\ &= \left(\frac{\varDelta}{n} + \beta\zeta\right)\phi + \beta Y(t) - \beta && (9)\end{aligned}$$

and so $\mathbf{E}(D(\xi_1) + \beta Y(\xi_1) | Y(t)) < D(t) + \beta Y(t)$ iff $\frac{\varDelta}{n}\phi + \beta\zeta\phi - \beta > 0$.

Now, suppose that the result holds up to $(l-1)$. By a similar argument, $\mathbf{E}(D(\xi_{l-1}) + \beta Y(\xi_{l-1})|Y(t)) < \mathbf{E}(D(t') + \beta Y(t')|Y(t))$ for all $t' \in [\xi_{l-1}, \xi_l)$. The condition

$$\mathbf{E}(D(\xi_{l-1}) + \beta Y(\xi_{l-1})|Y(t)) < \mathbf{E}(D(\xi_l) + \beta Y(\xi_l)|Y(t))$$

holds iff $\frac{\Delta}{n}\phi + \beta\zeta\phi - \beta > 0$.

This result implies that a threshold-based policy is optimal for randomization over a broad class of real node dynamics.

5.3 Optimal Strategy of the Adversary

The optimal scanning rate is the solution to

$$\begin{array}{ll} \text{maximize} & \mathbf{E}(D(R) - \omega\alpha) \\ \text{s.t.} & \alpha \in [0, \alpha_{max}] \end{array} \tag{10}$$

which is a trade-off between the probability of identifying the real node and the adversary's cost of bandwidth.

The scanning rate is assumed to be constant and chosen based on the randomization policy of the system.

Since the scanning process is random, the detection probability at the time of randomization, $D(R)$, is equal to the fraction of the network scanned at time R, $\frac{\alpha}{\tau n}R$. Based on Theorem 1, the detection probability is given as

$$D(R) = \begin{cases} \frac{\alpha}{\tau n}T_0, & \left(\frac{\alpha}{\tau n} + \beta\zeta\right)\phi < \beta \\ 0, & \text{else} \end{cases} \tag{11}$$

where T_0 is the time for the number of connections to go to 0. Hence the value of α that maximizes $D(R)$ is $\alpha = \beta\tau n - \beta\zeta$. The overall utility of the adversary is equal to $\beta(\tau n - \zeta)(\tau n)\mathbf{E}(T_0) - \omega(\beta\tau n - \beta\zeta)$.

Proposition 3. *Let* $\alpha^* = \min\left\{\alpha_{max}, \beta\tau n\left(\frac{1}{\phi} - \frac{1}{\zeta}\right)\right\}$. *Then the unique Stackelberg equilibrium of the network interaction game is for the adversary to choose* α *based on*

$$\alpha = \begin{cases} \alpha^*, & \mathbf{E}(T_0) - \omega\tau n > 0 \\ 0, & \textit{else} \end{cases} \tag{12}$$

Proof. The proof follows from Theorem 1 and the fact that the adversary's utility is negative unless the condition $\mathbf{E}(T_0) - \omega\tau n$ holds.

Proposition 3 indicates that the adversary follows a threshold decision rule, in which the adversary scans the system at the rate α^* if the expected time before randomization, T_0, exceeds the expected time to scan the entire network, τn. The adversary can determine the optimal scanning rate over a period of time by initially scanning at a low rate and incrementally increasing the rate until randomization occurs, signifying that the threshold scanning rate α^* has been found.

6 Simulation Study

A numerical study was performed using Matlab, consisting of three components. First, we studied the timing-based detection game of Section 4.1. Second, we considered the fingerprinting-based detection game of Section 4.2. Third, we analyzed the network-level interaction of Section 5.

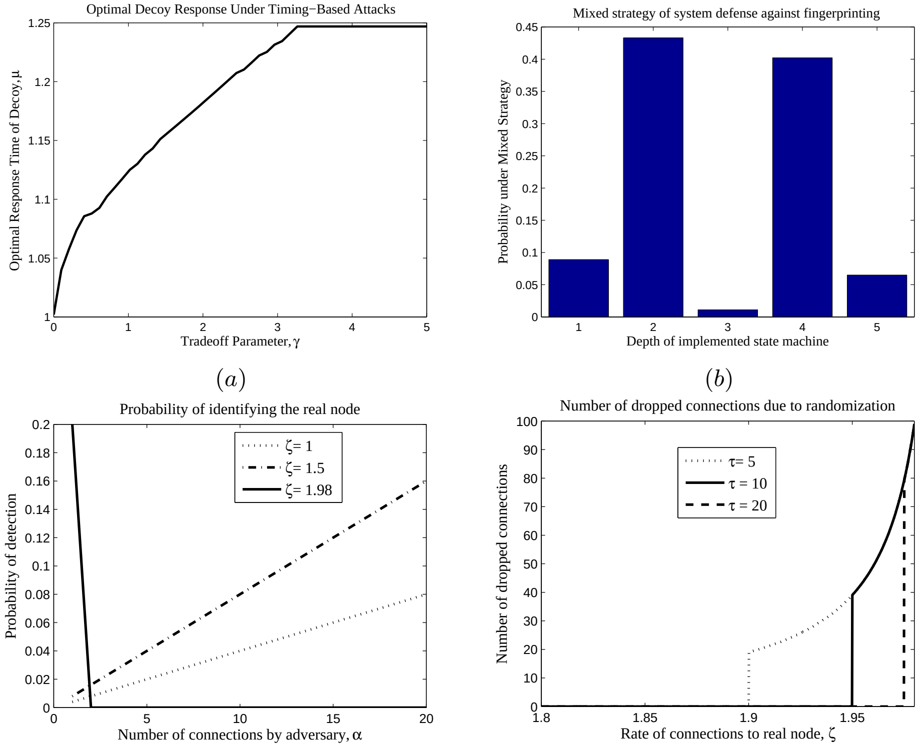
For the timing-based detection game, we considered a network of 100 nodes, with 1 real node and 99 decoy nodes. The real nodes were assumed to have mean response time of 1, while the response time of the decoys varied in the range $[1, 1.25]$. The parameter α, representing the amount of real traffic, was set equal to 0, while the capacity c of the virtual network was equal to 1. The trade-off parameter γ took values from 1 to 5, while the number of queries by the adversary ranged from $T = 1$ to $T = 50$.

We observed that the timing-based detection game converged to a pure-strategy Nash equilibrium in each simulated case. Figure 1(a) shows the mean response time of the decoy nodes as a function of the trade-off parameter, γ. As the cost of delays to the real nodes increases, the response time of the decoys increases as well. For lower values of γ, it is optimal for the real and decoy nodes to have the same response time.

For detection via system fingerprinting, we considered a state machine of diameter 4, consistent with the simplified TCP state machine of [14], implying that there are 5 possible strategies in the game of Section 4.2. We considered a cost of 0.2 for the system and adversary, so that the normalized cost of implementing the entire state machine was equal to 1. Figure 1(b) shows a histogram representing the mixed strategy of the system. The mixed strategy indicates that roughly half of the decoy nodes should implement only the first level of states in the state diagram, while the remaining half should implement the entire state machine, for this particular choice of the parameter values. This suggests an optimal allocation of half high-interaction and half low-interaction decoys, leading to a resource-expensive strategy.

In studying the network-level interaction between the system and adversary, we considered a network of $n = 100$ virtual nodes with detection time $\tau = 5$ based on the previous simulation results. The trade-off parameter $\beta = 0.1$. The real node was assumed to serve users according to an M/M/1 process with arrival rate $\zeta = 0.4$ and service rate $\phi = 2$. The cost of each connection to the adversary was set at $\omega = 2$. Figure 1(c) shows the probability of detection for the adversary as a function of the number of simultaneous connections initiated by the adversary. The probability of detection increases linearly until the threshold is reached; beyond the threshold, the system randomizes as soon as the scanning begins and the probability of detection is 0. Furthermore, as the rate of connection requests to the real node, quantified by the parameter ζ, increases, the cost of randomization for the real node increases, leading to longer waiting times between randomization and higher probability of detection.

As shown in Figure 1(d), the number of dropped connections due to randomization is zero when ζ is small, since the optimal strategy for the system is to wait until all connections terminate. As ζ approaches the capacity of the



(a) (b)

(c) (d)

Fig. 1. Numerical results based on our proposed game-theoretic framework. (a) The timing-based detection game of Section 4.1 converged to a pure-strategy equilibrium in all experimental studies. The pure strategy of the system is shown as a function of the trade-off parameter, γ. A larger value of γ results in a slower response rate due to increased delay to the real nodes. (b) Histogram of the mixed strategy of the system for the fingerprinting game of Section 4.2 using the TCP state machine. The optimal strategy is to implement only the initial states of the protocol and the entire protocol with roughly equal probability. (c) Detection probability as a function of the number of simultaneous connections by the adversary. The detection probability increases before dropping to zero when the randomization threshold is reached. (d) Number of dropped connections when the number of adversary connections $\alpha = 5$. The number of dropped connections is initially zero, as the adversary scanning rate is below threshold, and then increases as the rate of connection to the real node approaches the capacity of the real node.

real node, the number of dropped connections increases. The effectiveness of the decoy, described by the time τ required to detect the decoy, enables the system to operate for larger values of ζ (i.e., higher activity by the real nodes) without dropping connections.

7 Conclusion

We studied the problem of IP randomization in decoy-based moving target defense by formulating a game-theoretic framework. We considered two aspects of the design of decoy networks. First, we presented an analytical approach to modeling detection of nodes via timing-based analysis and protocol fingerprinting and identified decoy design strategies as equilibria of two-player games. For the fingerprinting attack, our approach was based on a finite state machine model of the protocol being fingerprinted, in which the adversary attempts to identify states of the protocol that the system has not implemented. Second, we formulated the interaction between an adversary scanning a virtual network and the hypervisor determining when to randomize the IP address space as a two-player Stackelberg game between the system and adversary. We proved that there exists a unique Stackelberg equilibrium to the interaction game in which the system randomizes only if the scanning rate crosses a specific threshold. Simulation study results showed that the timing-based game consistently has a pure-strategy Nash equilibrium with value that depends on the trade-off between detection probability and cost, while the fingerprinting game has a mixed strategy equilibrium, suggesting that networks should consist of a mixture of high- and low-interaction decoys.

While our current approach incorporates the equilibria of the single-node interaction games as parameters in the network-level game, a direction of future work will be to compute joint strategies at both the individual node and network level simultaneously. An additional direction of future work will be to investigate dynamic game structures, in which the utilities of the players, as well as parameters such as the number of nodes and the system resource constraints, change over time. We will also investigate "soft blacklisting" techniques, in which an adversary adaptively increases the delays when responding to requests from suspected adversaries, at both the real and decoy nodes. Finally, modeling the ability of decoys to gather information on the goals and capabilities of the adversary is a direction of future work.

References

1. Abu Rajab, M., Monrose, F., Terzis, A.: On the impact of dynamic addressing on malware propagation. Proceedings of the 4th ACM workshop on Recurring malcode pp. 51–56 (2006)
2. Alpcan, T., Başar, T.: Network Security: A Decision and Game-Theoretic Approach. Cambridge University Press (2010)

3. Antonatos, S., Akritidis, P., Markatos, E.P., Anagnostakis, K.G.: Defending against hitlist worms using network address space randomization. Computer Networks 51(12), 3471–3490 (2007)
4. Bohacek, S., Hespanha, J., Lee, J., Lim, C., Obraczka, K.: Game theoretic stochastic routing for fault tolerance and security in computer networks. IEEE Transactions on Parallel and Distributed Systems 18(9), 1227–1240 (2007)
5. Cao, J., Andersson, M., Nyberg, C., Kihl, M.: Web server performance modeling using an M/G/1/K PS queue. 10th IEEE International Conference on Telecommunications (ICT) pp. 1501–1506 (2003)
6. Carter, K.M., Riordan, J.F., Okhravi, H.: A game theoretic approach to strategy determination for dynamic platform defenses. Proceedings of the First ACM Workshop on Moving Target Defense pp. 21–30 (2014)
7. Chisnall, D.: The Definitive Guide to the Xen Hypervisor. Prentice Hall (2007)
8. Clark, A., Sun, K., Poovendran, R.: Effectiveness of IP address randomization in decoy-based moving target defense. Proceedings of the 52nd IEEE Conference on Decision and Control (CDC) pp. 678–685 (2013)
9. Franz, M.: E unibus pluram: massive-scale software diversity as a defense mechanism. Proceedings of the 2010 workshop on New security paradigms pp. 7–16 (2010)
10. Giuffrida, C., Kuijsten, A., Tanenbaum, A.S.: Enhanced operating system security through efficient and fine-grained address space randomization. USENIX Security Symposium (2012)
11. Holz, T., Raynal, F.: Detecting honeypots and other suspicious environments. In: IEEE Information Assurance and Security Workshop (IAW). pp. 29–36 (2005)
12. Jafarian, J.H.H., Al-Shaer, E., Duan, Q.: Spatio-temporal address mutation for proactive cyber agility against sophisticated attackers. Proceedings of the First ACM Workshop on Moving Target Defense pp. 69–78 (2014)
13. Jajodia, S., Ghosh, A.K., Subrahmanian, V., Swarup, V., Wang, C., Wang, X.S.: Moving Target Defense II. Springer (2013)
14. Kurose, J., Ross, K.: Computer Networking. Pearson Education (2012)
15. Larsen, P., Homescu, A., Brunthaler, S., Franz, M.: Sok: Automated software diversity. IEEE Symposium on Security and Privacy pp. 276–291 (2014)
16. Mukkamala, S., Yendrapalli, K., Basnet, R., Shankarapani, M., Sung, A.: Detection of virtual environments and low interaction honeypots. IEEE Information Assurance and Security Workshop (IAW) pp. 92–98 (2007)
17. Provos, N.: A virtual honeypot framework. Proceedings of the 13th USENIX security symposium 132 (2004)
18. Provos, N., Holz, T.: Virtual Honeypots: From Botnet Tracking to Intrusion Detection. Addison-Wesley Professional (2007)
19. Robinson, J.: An iterative method of solving a game. The Annals of Mathematics 54(2), 296–301 (1951)
20. Ross, S.M.: Introduction to Probability Models. Academic Press (2009)
21. Rowe, J., Levitt, K., Demir, T., Erbacher, R.: Artificial diversity as maneuvers in a control-theoretic moving target defense. Moving Target Research Symposium (2012)
22. Shamsi, Z., Nandwani, A., Leonard, D., Loguinov, D.: Hershel: single-packet os fingerprinting. ACM international conference on Measurement and modeling of computer systems pp. 195–206 (2014)
23. Sultan, F., Srinivasan, K., Iyer, D., Iftode, L.: Migratory TCP: Connection migration for service continuity in the internet. Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems pp. 469–470 (2002)

24. Van Dijk, M., Juels, A., Oprea, A., Rivest, R.L.: Flipit: The game of stealthy takeover. Journal of Cryptology 26(4), 655–713 (2013)
25. Wolfgang, M.: Host discovery with NMAP. http://moonpie.org/writings/discovery.pdf (2002)
26. Zhu, Q., Başar, T.: Game-theoretic approach to feedback-driven multi-stage moving target defense. Decision and Game Theory for Security pp. 246–263 (2013)