Optimal Timing in Dynamic and Robust Attacker Engagement During Advanced Persistent Threats

Jeffrey Pawlick
NYU Tandon School of Eng. and
US Army Research Lab
jpawlick@nyu.edu

Thi Thu Hang Nguyen
LAAS-CNRS and
SAS Torus Actions
tthnguye@laas.fr

Edward Colbert
US Army Research Lab
and Virginia Tech
ecolbert@vt.edu

Quanyan Zhu
NYU Tandon School of Eng.
quanyan.zhu@nyu.edu

***Abstract*—Advanced persistent threats (APTs) are stealthy attacks which make use of social engineering and deception to give adversaries insider access to networked systems. Against APTs, active defense technologies aim to create and exploit information asymmetry for defenders. In this paper, we study a scenario in which a powerful defender uses honeynets for active defense in order to observe an attacker who has penetrated the network. Rather than immediately eject the attacker, the defender may elect to gather information. We introduce an undiscounted, infinite-horizon Markov decision process on a continuous state space in order to model the defender's problem. We find a threshold of information that the defender should gather about the attacker before ejecting him. Then we study the robustness of this policy using a Stackelberg game. Finally, we simulate the policy for a conceptual network. Our results provide a quantitative foundation for studying optimal timing for attacker engagement in network defense.**

***Index Terms*—Security, Markov decision process, Stackelberg game, advanced persistent threat, attacker engagement**

I. Introduction

Traditional cybersecurity techniques such as firewall defense and role-based access control have been shown to be insufficient against advanced and persistent threats (APTs). Recent breaches of the Democratic National Committee [16] and the U.S. Office of Personal Management [3] have highlighted that advanced actors are capable of undermining these defenses through social engineering, zero-day exploits, and deceptively mimicking benign code. Intruders establish themselves with a network using techniques such as spear-phishing or direct physical access. Bring your own device (BYOD) aspects of wireless networks expose additional routes for malware entry [10]. After entry, attackers move laterally within the network to escalate privileges and advance towards a target asset.

This work is partially supported by an NSF IGERT grant through the Center for Interdisciplinary Studies in Security and Privacy (CRISSP) at New York University, by the grant CNS-1544782, EFRI-1441140, and SES-1541164 from National Science Foundation (NSF) and DE-NE0008571 from the Department of Energy. Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-17-2-0104. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

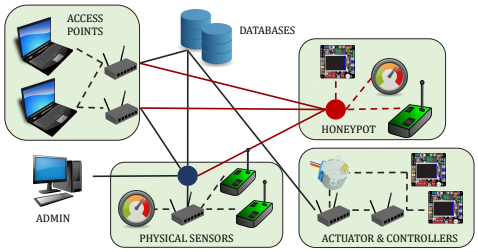


Fig. 1: A honeynet in a process control network. Dashed lines represent wireless connections. At the top right, a honeynet records activity in order to learn about attackers.

A. Active Cyber Defense and Honeynets

Often security research studies deceptive attackers and purely reactive defenders. But new techniques aim to allow defenders to gain the upper hand in information asymmetry. The U.S. Department of Defense has defined *active cyber defense* as "synchronized, real-time capability to discover, detect, analyze, and mitigate threats and vulnerabilities... using sensors, software, and intelligence..." [13]. These techniques both investigate attackers and manipulate their beliefs [15]. *Honeynets* and *virtual attack surfaces* are emerging techniques which accomplish both purposes. They create false network views in order to lure the attacker into a designated part of a network where he can be contained and observed within a controlled environment [2]. Figure 1 gives a conceptual example of a honeynet placed within a process control network in critical infrastructure. A wired backbone connects wireless routers that serve sensors, actuators, controllers, and access points. A honeynet emulates a set of sensors and controllers and records attacker activities. Engaging with an attacker in order to gather information allows defenders to update their threat models and develop more effective defenses.

B. Timing in Attacker Engagement

Our work considers this seldom studied case of a powerful defender who observes multiple attacker movements within a network. This sustained engagement with an attacker comes at the risk of added exposure. The situation gives rise to an interesting trade-off between information gathering and

short-term security. How long should administrators allow an attacker to remain in a honeypot before ejecting the attacker? How long should they attempt to lure an attacker from an operational system to a honeypot? Our abstracts away from network topology or protocol in order to focus exclusively on these questions of timing in attacker engagement.

C. Contributions

We make the following principle contributions:

1) We introduce an undiscounted, infinite-horizon Markov decision process (MDP) on a continuous state space to model attacker movement constrained by a defender who can eject the attacker from the network at any time, or allow him to remain in the network in order to gather information.
2) We analytically obtain the value function and optimal policy for the defender, and verify these numerically.
3) These results obtain closed-form conditions under which it is optimal to retain an attacker in the network.
4) To test the robustness of the optimal policy, we develop a zero-sum, Stackelberg game model in which the attacker leads by choosing a parameter of the game. We obtain a worst-case bound on the defender's utility.
5) We use simulations to illustrate the optimal policy for a conceptual network.

D. Related Work

Game theory and decision theory are often used to study cybersecurity [20], [19], [11], [8] due to its adversarial nature. In particular, game-theoretic design of honeypot deployment has been an active research area. Signaling games are used to model attacker beliefs about honeypots in [5], [14]. Honeynet deployment from a network point of view is systematized in [2]. Ref. [12] develops a model for lateral movements and formulates a game by which an automated defense agent protects a network asset. Durkota et al. model dynamic attacker engagement using attack graphs and a MDP [6]. Zhuang et al. study security investment and deception using a multiple round signaling game [21]. Our work fits within the context of these papers, but we focus on questions of timing. Reference [7] studies the belief of the attacker, and suggests that the attacker should be ejected when he becomes suspicious that he may be in a honeypot. This is a useful complement to the present work. Finally, this paper fits within the general category of optimal stopping problems. Optimal stopping problems with a finite horizon can be solved directly by dynamic programming, but our problem has an infinite horizon (and is undiscounted).

II. Problem Formulation

A discrete-time, continuous state MDP can be summarized by the tuple $\langle \mathbb{X}, \mathbb{A}, \mu, q \rangle$, where $\mathbb{X}$ is the continuous state space, $\mathbb{A}$ is the set of actions, $\mu : \mathbb{X} \times \mathbb{A} \to \mathbb{R}$ is the reward function, and $q : \mathbb{X} \times \mathbb{A} \times \mathbb{X} \to \mathbb{R}_+$ is the transition kernel. In this section, we describe each of the elements of $\langle \mathbb{X}, \mathbb{A}, \mu, q \rangle$.

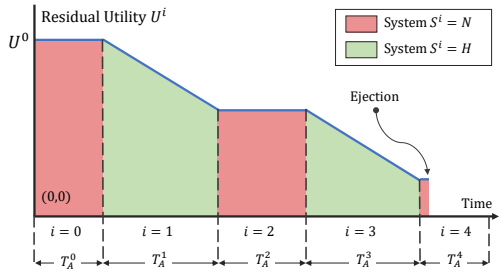


Fig. 2: A moves throughout a network between honeypots H and normal systems N. D can earn a total of U^0 utility for investigating A. When A is in a honeypot, D learns and the residual utility for future investigation decreases. Near $U^i = 0$, the risk of exposure outweighs the benefit of surveillance, and D ejects A at stage $i = 4$ (in this example).

A. State Space $\mathbb{X}$

An attacker A moves throughout a network containing two types of systems S: honeypots H and normal systems N. At any time, a network defender D can eject A from the network. L denotes having left the network. Together, we have $S \in \mathbb{S} \triangleq \{H, N, L\}$.

Let $i \in 0, 1, 2, \ldots$ denote the discrete *stage* of the game, *i.e.*, i indicates the order of the systems visited. D observes the types S^i of the systems that A visits. The attacker, on the other hand, does not know the system types.

We assume that there is a maximum amount of information that D can learn from investigating A. Let U^0 denote the corresponding utility that D receives for this information. At stage $i \in 0, 1, 2, \ldots$, let $U^i \in \mathbb{U} \triangleq [0, U^0]$ denote the *residual utility*[1] available to D for investigating A. For instance, at $i = 5$, D may have recorded the attacker's time of infiltration, malware type and operating system, but not yet any privilege escalation attempts, which could reveal the attacker's objective. In that case, D may estimate that $U^5 \approx 0.6U^0$, *i.e.*, D has learned approximately 60% of all possible information[2] about A.

D should use U^i together with S^i to form his policy. For instance, with $U^5 \approx 0.6U^0$, D may allow A to remain in a honeypot $S^5 = H$. But after observing a privilege escalation attempt, with $U^6 \approx 0.8U^0$, D may eject A from $S^6 = H$, since there is little more to be learned about him. Therefore, U^i and S^i are both states. The full state space is $\mathbb{X} = \mathbb{U} \times \mathbb{S}$. Figure 2 summarizes the interaction.

B. One-Stage Actions $\mathbb{A}$

Let $\aleph_0$ denote the cardinality of the set of natural numbers and $\mathbb{R}_+$ denote the set of non-negative real numbers. Then

[1] Quantification of control- and game-theoretic utility parameters (such as U^0) is never exact, and requires substantial effort. In this scenario, D must first judge what type of attacker he is facing. If it is a simple, brute-force botnet scan, U^0 will be low, while if it is a complex, targeted attack coming from a state actor, U^0 will be high.

[2] Here D must specify the pieces of information that he hopes to learn about A, as well as their relative values.

define $T_D = \{T_D^0, T_D^1, T_D^2, \ldots\} \in \mathbb{R}_+^{\aleph_0}$ such that T_D^i denotes the time that D plans to wait at stage i before ejecting A from the network. The single-stage action of D is to choose $T_D^i \in \mathbb{A} = \mathbb{R}_+$.

C. Reward Function μ

To formulate the reward, we also need to define $T_A = \{T_A^0, T_A^1, T_A^2, \ldots\} \in \mathbb{R}_+^{\aleph_0}$. For each $i \in 0, 1, 2, \ldots,$ T_A^i denotes the duration of time that A plans to wait at stage i before changing to a new system.

Let $C_N < 0$ denote the average cost per unit time that D incurs while A resides in normal systems[3]. This cost may be estimated by a sum of the costs $\phi_j^m < 0$ per unit time of each known vulnerability $j \in 1, 2, \ldots, J-1$ on each the systems $m \in 1, 2, \ldots, M$ in the network, weighted by the likelihoods $\rho_j^m \in [0, 1]$ that A exploits the vulnerability. This data can be obtained from the National Vulnerabilities Database [1]. Of course, D may not be aware of some system vulnerabilities. Let ϕ_J^m denote an estimate of damage that could be caused by an unknown vulnerability J on each system $m \in 1, 2, \ldots, M$, and let ρ_J^m denote a heuristic likelihood that A can exploit the vulnerability[4].

$$C_N = \frac{1}{M} \sum_{m=1}^{M} C_N^m = \frac{1}{M} \sum_{m=1}^{M} \sum_{j=1}^{J} \rho_j^m \phi_j^m.$$

Let $C_H \leq 0$ denote a cost that D pays to maintain A in a honeypot. This cost could represent the expense of hiring personnel to monitor the honeypot, the expense of redeployment, or the loss of informational advantage from A reconnoitering the honeypot. Let $\mathbb{R}_{++}$ denote the set of strictly positive real numbers. Finally, let $v \in \mathbb{R}_{++}$ denote the utility per unit time that D gains from learning about A while he is in honeypots. We assume that $v > -C_H$, *i.e.*, that the benefit per unit time from observing A in a honeypot exceeds the cost.

Define the function $\mu : \mathbb{U} \times \mathbb{S} \times \mathbb{R}_+ \to \mathbb{R}$ such that $\mu(U^i, S^i, T_D^i \,|\, T_A^i)$ gives the one-stage reward to D if the residual utility is U^i, A is in system S^i, A waits for T_A^i before moving, and D waits for T_D^i before ejecting A. Let $T^i \triangleq \min(T_A^i, T_D^i)$ denote the time for which A remains at system S^i before moving or being ejected. Also let $\mathbf{1}\{P\}$ be the indicator function which returns 1 if it the statement P is true. We have $\mu(U^i, S^i, T_D^i \,|\, T_A^i) =$

$$\mathbf{1}\{S = N\} C_N T^i + \mathbf{1}\{S = H\} \left(\min\left(T^i v, U^i\right) + C_H T^i\right).$$

D. Transition Kernel q

Let $\mathbb{R}_+$ denote the set of non-negative real numbers. For stage $i \in 0, 1, 2, \ldots,$ and given attacker and defender move times T_A^i and T_D^i, respectively, define the transition kernel $q : \mathbb{U} \times \mathbb{S} \times \mathbb{R}_+ \times \mathbb{U} \times \mathbb{S} \to \mathbb{R}_+$ such that, for all residual utilities $U^i \in \mathbb{U}$ and system types $S^i \in \mathbb{S}$

$$\int_{U^{i+1} \in \mathbb{U}} \int_{S^{i+1} \in \mathbb{S}} q\left(U^{i+1}, S^{i+1}, T_D^i, U^i, S^i \,|\, T_A^i\right) = 1,$$

where U^{i+1} and S^{i+1} denote the residual utility and system type, respectively, at the next stage.

Let $p \in [0, 1]$ denote the fraction of normal systems in the network[5]. For a real number y, let $\delta(y)$ be the Dirac delta function. For brevity, let $\Phi(U^i, T) \triangleq \max\{U^i - vT, 0\}$. If $T_A^i > T_D^i$, then D ejects A from the system, and we have $q(U^{i+1}, S^{i+1}, T_D^i, U^i, S^i \,|\, T_A^i) =$

$$\begin{aligned} &\mathbf{1}\left\{S^i = L \cap S^{i+1} = L\right\} \delta\left(U^{i+1} - U^i\right) + \\ &\quad \mathbf{1}\left\{S^i = N \cap S^{i+1} = L\right\} \delta\left(U^{i+1} - U^i\right) + \\ &\quad \mathbf{1}\left\{S^i = H \cap S^{i+1} = L\right\} \delta\left(U^{i+1} - \Phi(U^i, T_D^i)\right). \end{aligned} \quad (1)$$

If $T_A^i \leq T_D^i$, then A changes systems, and we have $q\left(U^{i+1}, S^{i+1}, T_D^i, U^i, S^i \,|\, T_A^i\right) =$

$$\begin{aligned} &p\mathbf{1}\left\{S^i = N \cap S^{i+1} = N\right\} \delta\left(U^{i+1} - U^i\right) + \\ &\quad (1-p)\,\mathbf{1}\left\{S^i = N \cap S^{i+1} = H\right\} \delta\left(U^{i+1} - U^i\right) + \\ &\quad p\mathbf{1}\left\{S^i = H \cap S^{i+1} = N\right\} \delta\left(U^{i+1} - \Phi(U^i, T_A^i)\right) + \\ &(1-p)\,\mathbf{1}\left\{S^i = H \cap S^{i+1} = H\right\} \delta\left(U^{i+1} - \Phi(U^i, T_A^i)\right). \end{aligned} \quad (2)$$

The equations can be understood by considering an example. If $T_A^i \leq T_D^i$ and A is currently in a honeypot, then Eq. (2) shows that the remaining utility will be $\delta\left(U^{i+1} - \Phi(U^i, T_A^i)\right)$. There is p probability that the next system is N, and $(1-p)$ that it is H.

E. Infinite-Horizon, Undiscounted Reward

For stage $i \in 0, 1, 2, \ldots,$ define the stationary deterministic feedback policy $\theta : \mathbb{U} \times \mathbb{S} \to \mathbb{R}_+$ such that $T_D^i = \theta(U^i, S^i)$ gives the time that D waits before ejecting A if the residual utility is U^i and the system type is S^i. Let Θ denote the space of all such stationary policies. Define the expected infinite-horizon, undiscounted reward by $\mathcal{V}_\theta^i : \mathbb{U} \times \mathbb{S} \to \mathbb{R}$ such that $\mathcal{V}_\theta^i(U^i, S^i)$ gives the expected reward from stage i onward for using the policy θ when the residual utility is U^i and the type of the system is S^i. $\mathcal{V}_\theta^i(U^i, S^i)$ also depends on the attacker's choice of T_A^k, but for a given T_A^k, it is expressed by

$$\mathcal{V}_\theta^i\left(U^i, S^i\right) = \mathbb{E}\left\{\sum_{k=i}^{\infty} \mu\left(U^k, S^k, \theta\left(U^k, S^k\right) \,|\, T_A^k\right)\right\},$$

where the states transition according to Eq. (1-2). Given an initial system type $S^0 \in \{H, N\}$, the overall problem for D is to find θ^* such that

$$\theta^* \in \arg\max_{\theta \in \Theta} \mathcal{V}_\theta^0\left(U^0, S^0\right).$$

[3]Future work can consider different costs for each individual system in a structured network.

[4]One alternate approach that can be used to quantify the impact of unknown zero-day attacks is k-zero day safety [18]

[5]Again, in a formal network, the kernel will differ among different honeypots and different normal systems. The fraction p is an approximation which is exact for a fully-connected network.

The undiscounted utility function demands Proposition 1.

Proposition 1. *$\mathcal{V}^i_{\theta^*}(U^i, S^i)$ is finite.*

Proof: See Appendix A. ∎

It is also convenient to define the *value function* as the reward for the optimal policy:

$$\mathcal{V}^i\left(U^i, S^i\right) \triangleq \mathcal{V}^i_{\theta^*}\left(U^i, S^i\right) = \max_{\theta\in\Theta} \mathcal{V}^i_{\theta}\left(U^i, S^i\right).$$

The Bellman principle [4] implies that for an optimal stationary policy θ^*, and for $i \in 0, 1, 2, \ldots$, $\theta^*\left(U^i, S^i\right) \in$

$$\arg\max_{T^i_D\in\mathbb{R}_+} \mu\left(U^i, S^i, T^i_D \,|\, T^i_A\right) + \int_{U^{i+1}\in\mathbb{U}} \int_{S^{i+1}\in\mathbb{S}} \mathcal{V}^{i+1}\left(U^{i+1}, S^{i+1}\right) q(U^{i+1}, S^{i+1}, T^i_D, U^i, S^i \,|\, T^i_A).$$

III. Analysis and Results

In this section, we solve for the value function and optimal policy. We start by obtaining the optimal policy in honeypots, and reducing the space of candidates for an optimal policy in normal systems. Then we present the value function and optimal policy separately, although they are derived simultaneously.

A. Reduced Action Spaces

Lemma 1 obtains the optimal waiting time for $S^i = H$.

Lemma 1. *(Optimal Policy for $S^i = H$) In honeypots, for any $i \in 0, 1, 2, \ldots$ and $U^i \in \mathbb{U}$, the value function is optimized by playing $T^i_D = U^i/v$.*

Proof: The value of the game is maximized if A passes through only honeypots and D ejects A when the residual utility is 0. D can achieve this by playing $T^i_D = U^i/v$ if $T^i_A > U^i/v$. On the other hand, if $T^i_A \leq U^i/v$, then it is optimal for D to allow A to change systems. This is optimal because the value function at stage $i+1$ is non-negative, since in the worst case D can eject A immediately if A arrives at a normal system. D can allow A to change systems by playing any $T^i_D \geq T^i_A$, although it is convenient for brevity of notation to choose $T^i_D = T^i_A$. ∎

Lemma 2 narrows the optimal waiting times for $S^i = N$.

Lemma 2. *(Reduced Action Space for $S^i = N$) In normal systems, for any $i \in 0, 1, 2, \ldots$ and $U^i \in \mathbb{U}$, the value function is optimized by playing either $T^i_D = 0$ or $T^i_D = T^i_A$.*

Proof: First, note that it is always suboptimal for D to eject A at a time less that T^i_A. That is, for stage $i \in 0, 1, 2, \ldots$, $\mathcal{V}^i_{\tilde{\theta}}(U^i, N) < \mathcal{V}^i_{\hat{\theta}}(U^i, N)$ for $0 = \hat{\theta}(U^i, N) < \tilde{\theta}(U^i, N) < T^i_A$. Second, note that D receives the same utility for ejecting A at any time greater than or equal to T^i_A, *i.e.*, $\mathcal{V}^i_{\hat{\theta}}(U^i, N) = \mathcal{V}^i_{\tilde{\theta}}(U^i, N)$ for $T^i_A \leq \hat{\theta}(U^i, N) \leq \tilde{\theta}(U^i, N)$. Then either 0 or T^i_A is optimal. ∎

Remark 1 summarizes Lemmas 1-2.

Remark 1. Lemma 1 obtains the unique optimal waiting time in honeypots. Lemma 2 reduces the candidate set of optimal waiting times in normal systems to two times: $T^i_D \in \{0, T^i_A\}$. These times are equivalent to stopping the Markov chain and allowing it to continue, respectively. Thus, Lemmas 1-2 show that the MDP is an optimal stopping problem.

B. Value Function Structure

To solve the optimal stopping problem, we must find the value function. We obtain the value function for a constant attacker action, *i.e.*, $T^0_A = T^1_A = ... \triangleq \bar{T}_A$. This means that $\mathcal{V}^i \equiv \mathcal{V}$. Define the following notation:

$$\delta \triangleq \bar{T}_A v, \quad \delta^D_1 \triangleq \bar{T}_A\left(v + C_H\right), \tag{3}$$

$$\lambda^D_N \triangleq \frac{-C_N}{1-p}, \quad \chi^D_H \triangleq \frac{v + C_H}{v}. \tag{4}$$

Note that δ and δ^D_1 are in units of utility, λ^D_N is in units of utility per second, and χ^D_H is unitless.

First, $\mathcal{V}(U^i, L) = 0$ for all $U^i \in \mathbb{U}$, because no further utility can be earned after D ejects A. Next, $\mathcal{V}(0, S) = 0$ for both $S \in \{H, N\}$, because no positive utility can be earned in either type of system. $\mathcal{V}$ can now be solved backwards in U^i from $U^i = 0$ to $U^i = U^0$ using these terminal conditions. Depending on the parameters, it is possible that $\forall U^i \in \mathbb{U}$, $\theta^*(U^i, N) = 0$ and $\mathcal{V}(U^i, N) = 0$, *i.e.*, D should eject A from all normal systems immediately. Lemma 3 describes the structure of the optimal policy outside of this case.

Lemma 3. *(Optimal Policy Structure) Outside of the case that $\forall U^i \in \mathbb{U}$, $\theta^*(U^i, N) = 0$, there exists a residual utility $\omega \in \mathbb{U}$ such that:*

- *for $U^i < \omega$, $\theta^*(U^i, N) = 0$ and $\mathcal{V}(U^i, N) = 0$,*
- *for $U^i > \omega$, $\theta^*(U^i, N) = \bar{T}_A$ and $\mathcal{V}(U^i, N) > 0$.*

Proof: See Appendix B. ∎

Remark 2. Typical intuition dictates that a security professional should immediately eject a detected attacker from normal systems in a network. Lemma 3 shows that this is indeed optimal when $\omega \geq U^0$. When $\omega < U^0$, however, it is better to allow the attacker to remain. A principle contribution of our work is finding this threshold ω.

C. Value Function Threshold

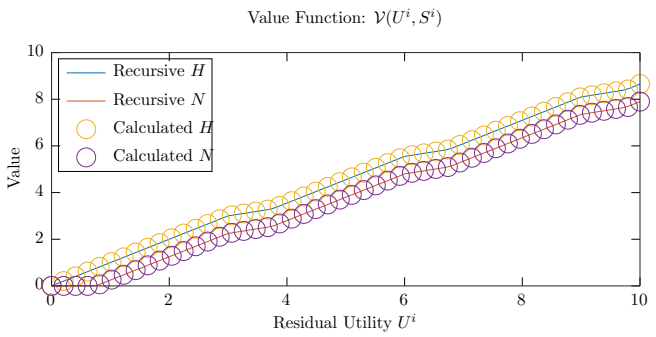
Next, for $x \in \mathbb{R}$, define

$$k\left[x\right] \triangleq \begin{cases} \lfloor x/\delta \rfloor, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}, \tag{5}$$

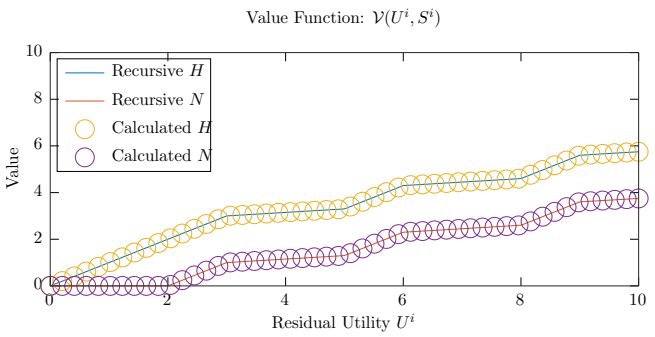
where $\lfloor \bullet \rfloor$ is the floor function. The floor function is required because μ is nonlinear in U^i. Then Theorem 1 gives ω in closed form.

Theorem 1. *(Threshold ω) Outside of the trivial case, the threshold ω of residual utility beyond which D should eject A is given by*

$$\omega = \delta\left(k\left[\omega\right] + \frac{\lambda^D_N}{\left(v + C_H\right)\left(1-p\right)^{k[\omega]}} - \frac{1 - \left(1-p\right)^{k[\omega]}}{p\left(1-p\right)^{k[\omega]}}\right),$$



(a) $p = 0.60$, $\omega \approx 0.83$, $\delta = 3.0$, $U^0 = 10$



(b) $p = 0.85$, $\omega \approx 2.2$, $\delta = 3.0$, $U^0 = 10$

Fig. 3: Value functions with $p = 0.60$ and $p = 0.85$. The top and bottom curves depict $\mathcal{V}(U^i, H)$ and $\mathcal{V}(U^i, N)$, respectively, as a function of U^i. The circles plot the analytical $\mathcal{V}(U^i, S)$, $S \in \{H, N\}$ from Theorem 2, and the solid lines verify this using an iterative numerical method.

where $k[\omega]$ is defined as in Eq. (5), and it can be shown that

$$k\left[\omega\right] = \left\lfloor \log_{1-p}\left(1 + \frac{pC_N}{(1-p)\left(v + C_H\right)}\right)\right\rfloor,$$

if the argument of the logarithm is positive. If not, then the optimal policy is for D to eject A from normal systems immediately.

Proof: See Appendix C. ∎

Remark 3 gives some intuition about Theorem 1.

Remark 3. Numerical results suggest that in many cases (such as those in Fig. 3), $k[\omega] = 0$. In that case, we have $\omega = -\delta C_N / \left((v + C_H)(1-p)\right)$. The threshold ω increases as the cost for normal systems (C_N) increases, decreases as the rate at which utility is gained in normal systems (v) increases, and decreases as the proportion of normal systems (p) increases.

Finally, Theorem 2 summarizes the value function.

Theorem 2. *(Value Function) The value function is given by*

$$\mathcal{V}\left(U^i, S^i\right) = \begin{cases} 0, & \text{if } S^i = L \\ f^D(U^i), & \text{if } S^i = H \\ \left\{f^D(U^i) - \bar{T}_A \lambda_N^D\right\}_+, & \text{if } S^i = N \end{cases},$$

where $\{\bullet\}_+$ denotes $\max\{\bullet, 0\}$, and $f^D : \mathbb{U} \to \mathbb{R}_+$ is

$$f^D\left(U^i\right) \triangleq \chi_H^D\left(U^i - \delta k[U^i]\right)(1-p)^{k[U^i]-k[U^i-\omega]} + \frac{\delta_1^D}{p}\left(1 - (1-p)^{k[U^i]-k[U^i-\omega]}\right) + k[U^i - \omega]\left(\delta_1^D - p\lambda_N^D \bar{T}_A\right).$$

Proof: See Appendix B. ∎

Remark 4 discusses the interpretation of Theorem 2.

Remark 4. The quantity $f^D(U^i)$ is the expected reward for future surveillance, while $\bar{T}_A \lambda_N^D$ is the expected damage that will be caused by A. In normal systems, when $U^i \leq \omega$, we have $f^D(U^i) \leq \bar{T}_A \lambda_N^D$, and the risk of damage outweighs the reward of future surveillance. Therefore, it is optimal for D to eject A, and $\mathcal{V}(U^i, N) = 0$. On the other hand, for $U^i > \omega$, it is optimal for D to allow A to remain for $\bar{T}_A$ before moving, so $\mathcal{V}(U^i, N) > 0$. Figure 3 gives examples of the value function.

D. Optimal Policy Function

Theorem 3 summarizes the optimal policy.

Theorem 3. *(Defender Optimal Policy) D achieves an optimal policy for $S^i \in \{H, N\}$ by playing*

$$\theta^*\left(U^i, S^i\right) = \begin{cases} U^i/v, & \text{if } S^i = H \\ \bar{T}_A, & \text{if } S^i = N \text{ and } U^i \geq \omega \\ 0, & \text{if } S^i = N \text{ and } U^i < \omega \end{cases}.$$

Proof: See Appendix B. ∎

IV. Robustness Evaluation

In this section, we evaluate the robustness of the policy θ^* by allowing A to choose the worst-case $\bar{T}_A$.

A. Equilibrium Concept

Let us write $\mathcal{V}_\theta(U^i, S^i \,|\, \bar{T}_A)$ and $\theta^*(U^i, S^i, \,|\, \bar{T}_A)$ to denote the dependence of the value and optimal policy, respectively, on $\bar{T}_A$. Next, define $\bar{\mathcal{V}} : \mathbb{R}_+ \to \mathbb{R}$ such that $\bar{\mathcal{V}}(\bar{T}_A)$ gives the expected utility to D over possible types of initial systems for playing θ^* as a function of $\bar{T}_A$. This is given by

$$\bar{\mathcal{V}}\left(\bar{T}_A\right) = p\mathcal{V}\left(U^0, N \,|\, \bar{T}_A\right) + (1-p)\,\mathcal{V}\left(U^0, H \,|\, \bar{T}_A\right). \quad (6)$$

Definition 1 formulates a *zero-sum Stackelberg equilibrium* [17] in which A chooses $\bar{T}_A$ to minimize Eq. (6), and D plays the optimal policy given $\bar{T}_A$ from Theorem 3.

Definition 1. (Stackelberg Equilibrium) A Stackelberg equilibrium (SE) of the zero-sum attacker-defender game is a strategy pair $(\bar{T}_A^*, \theta^*)$ such that

$$\bar{T}_A^* \in \arg\min_{\bar{T}_A} \bar{\mathcal{V}}_{\theta^*(U^i, S^i \,|\, \bar{T}_A)}\left(\bar{T}_A\right),$$

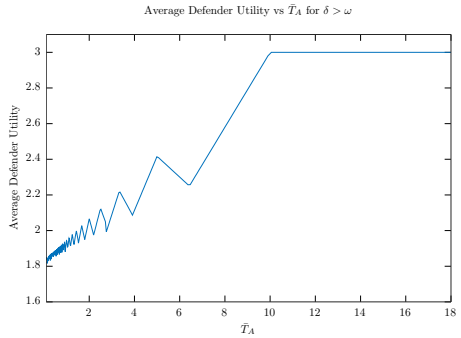


Fig. 4: $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A)$ for the case that $\delta < \omega$. Here, the worst case value is $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A^*) \approx 1.8$, which occurs as $\bar{T}_A \to 0$.

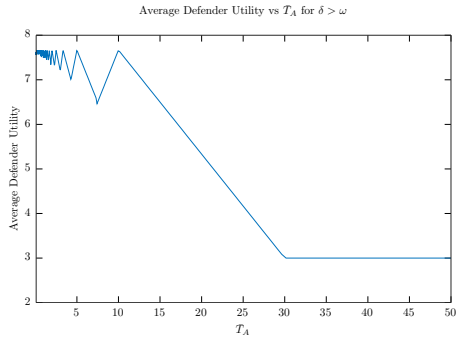


Fig. 5: $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A)$ for the case that $\delta > \omega$. Here, the worst case value is $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A^*) \approx 3.0$, which occurs for $\bar{T}_A > \omega \approx 30$.

and $\forall U^i \in \mathbb{U}$, $\forall S^i \in \mathbb{S}$,

$$\theta^*\left(U^i, S^i \,|\, \bar{T}_A^*\right) \in \arg\max_{\theta \in \Theta} \mathcal{V}_\theta\left(U^i, S^i \,|\, \bar{T}_A^*\right).$$

Definition 1 considers A as the Stackelberg game leader because our problem models an intelligent defender who reacts to the strategy of an observed attacker.

B. Equilibrium Analysis

$\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A)$ takes two possible forms, based on the values of δ and ω. Figure 4 depicts $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A)$ for $\delta < \omega$, and Fig. 5 depicts $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A)$ for $\delta > \omega$. Note that the oscillations are not produced by numerical approximation, but rather by the nonlinear value function[6]. The worst-case $\bar{T}_A^*$ is as small as possible for $\delta < \omega$ and is large for $\delta > \omega$. Theorem 4 states this result formally.

Theorem 4. *(Value as a function of $\bar{T}_A$) For low $\bar{T}_A$:*

$$\lim_{\bar{T}_A \to 0} \bar{\mathcal{V}}_{\theta^*}(\bar{T}_A) = U^0\left(1 + \frac{1}{v}\left(C_H + C_N \frac{p}{1-p}\right)\right). \quad (7)$$

Define $\bar{T}_\omega$ as $\bar{T}_A$ such that $U^0 = \omega$. Then for $\bar{T}_A \geq \max\{r, \bar{T}_\omega\}$, we have

$$\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A) = U^0(1-p)\frac{v + C_H}{v}. \quad (8)$$

[6]As $\bar{T}_A$ varies, the number of systems that A can visit before $U^i < \omega$ changes in a discrete manner. This causes the oscillations.

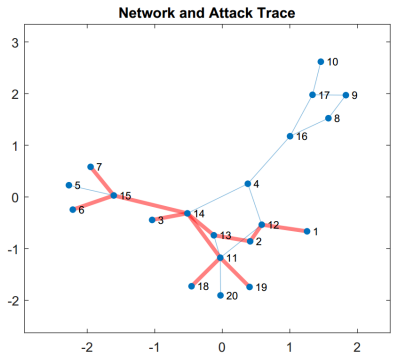


Fig. 6: The blue nodes and edges illustrate a 20-node network, and the red highlights indicate an example attack trace.

Proof: See Appendix D. ∎

Remarks 5-6 discuss Theorem 4 and Fig. 4-5.

Remark 5. The parameters of Fig. 4 and Fig. 5 differ only in C_N, which has a higher absolute value in Fig. 4. Since C_N only affects $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A)$ as $\bar{T}_A \to 0$, the plots are the same for high $\bar{T}_A$.

Remark 6. The connection between Fig. 4 and Fig. 5 can be visualized by translating the left sides of the curves vertically, while the right sides remain fixed. This gives network designers an intuition of how the worst-case value can be manipulated by changing the parameters of the game.

Finally, Corollary 1 summarizes the worst-case value.

Corollary 1. *(Worst-Case Value) The worst case value $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A^*)$ is approximated by*

$$U^0 \min_{\bar{T}_A}\left\{\left(1 + \frac{1}{v}\left(C_H + C_N \frac{p}{1-p}\right)\right), (1-p)\frac{v + C_H}{v}\right\}.$$

V. Simulation

In this section, we simulate a network which sustains five attacks and implements D's optimal policy θ^*. Consider the example network depicted in Fig. 1 in Section I. This network has 16 production nodes, including routers, wireless access points, wired admin access, and a database. It also has sensors, actuators, and controllers, which form part of a SCADA system. The network has 4 honeypots (in the top-right of the figure), configured to appear as additional SCADA system components.

Figure 6 depicts a view of the network in MATLAB [9]. The red line indicates an attack path, which enters through the wireless access point at node 1, passes through the honeynet in nodes 11, 18, and 19, and enters the SCADA components in nodes 6 and 7. The transitions are realized randomly.

Figure 7 depicts the cumulative utility of D over time for five simulated attacks. Towards the beginning of the attacks, D gains utility. But after learning nears completion (*i.e.*, $U^i \approx 0$), the losses C_N from normal systems dominate. The filled boxes in each trace indicate the ejection point dictated by θ^*. At these points, $U^i \leq \omega$. The ejection points are

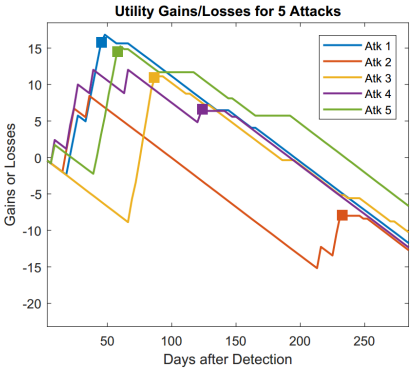


Fig. 7: The curves indicate the cumulative utility gains or losses for five simulated attacks. The solid squares indicate the optimal ejection time according to θ^*.

approximately at the maximum utility for traces 1, 3, and 5, and obtain a positive utility in trace 4. Trace 5 involves a long period in which $S^i = N$, and D sustains heavy losses. Since the traces are realized randomly, θ^* maximizes expected utility rather than realized utility.

VI. Discussion of Results

This paper aimed to assess how long an intelligent network defender that detects an attacker should observe the attacker before ejecting him. We found that the defender should keep the attacker in a honeypot as long as information remains to be learned and in a normal system until a threshold amount of information remains. This threshold is ω, at which the benefits of observation exactly balance the risks of information loss. Using this model, network designers can vary parameters (*e.g.*, the number of honeypots and the rate at which they gather information) in order to maximize the value function $\mathcal{V}$. In particular, we have examined the effect of the attacker move period $\bar{T}_A$ using a Stackelberg game in which A chooses the worst-case $\bar{T}_A$. Future work can use signaling games to calculate attacker beliefs p and $1-p$ based on defender strategies. Another direction, for distributed sensor-actuator networks, is to quantify the risk C_N of system compromise using optimal control theory.

Appendix A
Proof of Finite Expected Value

The maximum value of $\mathcal{V}^i_\theta(U^i, S^i)$ is achieved if A only visits honeypots. In this case, $\mathcal{V}^i_\theta(U^i, S^i) = (v + C_H)U^0/v$, so the expected utility is bounded from above. If D chooses a poor policy (for example, $\theta(U^i, S^i) = T^i_A$ for all $U^i \in \mathbb{U}$ and $S^i \in \mathbb{S}$), then $\mathcal{V}^i_\theta(U^i, S^i)$ can be unbounded below. On the other hand, D can always guarantee $\mathcal{V}^i_\theta(U^i, S^i) = 0$ (for example, by choosing $\theta(U^i, S^i) = 0$ for all $U^i \in \mathbb{U}$ and $S^i \in \mathbb{S}$). Therefore, the value of the *optimal* policy is bounded from below as well as from above.

Appendix B
Derivation of Value Function and Optimal Policy

For $S^i \in \{H, N\}$, the value function $\mathcal{V}(U^i, S^i)$ is piecewise-linear in U^i. The pieces result from different discrete numbers of systems that A visits. Let $\mathcal{V}(U^i, S^i)[a, b]$ denote $\mathcal{V}(U^i, S^i)$ restricted to the domain $U^i \in [a, b] \subset \mathbb{R}$. First, we find $\mathcal{V}(U^i, N)$ in terms of $\mathcal{V}(U^i, H)$. For any non-negative integer k, one step of the Bellman equation gives $\mathcal{V}(U^i, N)[k\delta, (k+1)\,\delta] =$

$$\left\{C_N \bar{T}_A + p\mathcal{V}\left(U^i, N\right)[k\delta, (k+1)\,\delta] + (1-p)\,\mathcal{V}\left(U^i, H\right)[k\delta, (k+1)\,\delta]\right\}_+,$$

where $\{\bullet\}_+$ denotes $\max\{\bullet, 0\}$. D achieves this maximization by continuing the game if the expected value for continuing is positive, and ejecting A if the expected value is negative.

Rearranging terms and using Eq. (3-4) gives $\mathcal{V}(U^i, N)\,[k\delta, (k+1)\,\delta] = \left\{\mathcal{V}\left(U^i, H\right)[k\delta, (k+1)\,\delta] - \lambda^D_N \bar{T}_A\right\}_+$. Now, we have defined ω as $U^i \in \mathbb{R}_+$ which makes the argument on the right side equal to zero. This obtains $\mathcal{V}\left(U^i, N\right)[k\delta, (k+1)\,\delta] =$

$$\begin{cases} 0, & \text{if } U^i \le \omega \\ \mathcal{V}\left(U^i, H\right)[k\delta, (k+1)\,\delta] - \lambda^D_N \bar{T}_A, & \text{if } U^i > \omega. \end{cases}$$

Next, we find $\mathcal{V}(U^i, H)$. First, consider $\mathcal{V}(U^i, H)[0, \delta]$. D keeps A in the honeypot until all residual utility is depleted, and then ejects him. Thus $\mathcal{V}(U^i, H)[0, \delta] = U^i \chi^D_H$. Next, for $k \in 1, 2, \ldots,$ consider $\mathcal{V}(U^i, H)[k\delta, (k+1)\delta]$. We have $\mathcal{V}\left(U^i, H\right)[k\delta, (k+1)\,\delta] =$

$$(v + C_H)\,\bar{T}_A + p\mathcal{V}\left(U^i - \delta, N\right)[(k-1)\,\delta, k\delta] + (1-p)\,\mathcal{V}\left(U^i - \delta, H\right)[(k-1)\,\delta, k\delta].$$

A bit of algebra gives $\mathcal{V}(U^i, H)[k\delta, (k+1)\,\delta] = \delta^D_1 + (1-p)\,\mathcal{V}\left(U^i - \delta, H\right)[(k-1)\,\delta, k\delta]$, if $U^i \le \omega + \delta$, and $\mathcal{V}(U^i, H)[k\delta, (k+1)\,\delta] = \delta^D_1 + \mathcal{V}\left(U^i - \delta, H\right)[(k-1)\,\delta, k\delta] - p\lambda^D_N \bar{T}_A$, otherwise. Solving this recursive equation for the case of $U^i \le \omega + \delta$ gives $\mathcal{V}(U^i, H)[k\delta, (k+1)\,\delta] =$

$$\delta^D_1 + \delta^D_1 (1-p) + \ldots + \delta^D_1 (1-p)^{k-1} + (1-p)^k\,\mathcal{V}\left(U^i - \delta k, H\right)[0, \delta]. \quad (9)$$

Using initial condition $\mathcal{V}(U, H)[0, \delta] = U\chi^D_H$ produces $f^D(U^i)$ for $U^i \le \omega$. For $U^i > \omega + \delta$, consider the integer k_1 such that $(k - k_1 - 1)\delta \le \omega < (k - k_1)\delta$. Then

$$\mathcal{V}(U^i, H)[k\delta, (k+1)\,\delta] = k_1\left(\delta^D_1 - p\lambda^D_N \bar{T}_A\right) + \mathcal{V}\left(U^i - k_1\delta, H\right)[(k - k_1 - 1)\delta, (k - k_1)\delta].$$

But the last term is simply $f^D\left(U^i - k_1\delta\right)$, and $k_1 = k\left[U^i - \omega\right]$ defined in Eq. (5). Substituting from Eq. (9) gives the entire function $f^D(U^i)$, $U^i \in \mathbb{U}$.

APPENDIX C
DERIVATION OF $k[\omega]$ AND ω

We solve first for $k[\omega]$ and then for ω. Because of the floor function in $k[\omega]$, we have that $\omega \in [k[\omega]\delta, (k[\omega]+1)\delta)$. Then for some $\epsilon \in [0,1)$, $\omega = (k[\omega]+\epsilon)\delta$.

Note that $f^D(\omega) = \bar{T}_A \lambda_N^D$, *i.e.*, the expected gain of surveillance is equal to the security risk at $U^i = \omega$. Therefore, we have $\bar{T}_A \lambda_N^D =$

$$\chi_H^D(\omega - \delta k[\omega])(1-p)^{k[\omega]} + \frac{\delta_1^D}{p}\left(1-(1-p)^{k[\omega]}\right). \quad (10)$$

Substituting for ω,

$$\bar{T}_A \lambda_N^D - \frac{\delta_1^D}{p} = (k[\omega]+\epsilon)\delta\chi_H^D(1-p)^{k[\omega]} - \delta k[\omega]\chi_H^D(1-p)^{k[\omega]} - (1-p)^{k[\omega]}.$$

This reduces to

$$\bar{T}_A \lambda_N^D - \frac{\delta_1^D}{p} = \epsilon\delta\chi_H^D(1-p)^{k[\omega]} - (1-p)^{k[\omega]},$$

which is uniquely solved by the $k[\omega]$ in Theorem 1. Now solving Eq. (10) for ω obtains the result in Lemma 1.

APPENDIX D
DERIVATION OF $\bar{\mathcal{V}}_{\theta^*}(\bar{T}_A)$

We solve the value function in two cases.

A. Limit as $\bar{T}_A \to 0$

As $\bar{T}_A \to 0$, ω and δ decrease, so $U^0 > \omega + \delta$, and the value functions follow f_2^D. Therefore, we find the limit of f_2^D as $\bar{T}_A \to 0$. As $\bar{T}_A \to 0$, $k[U^0] - k_1[U^0]$ remains finite, but $\delta_1^D \to 0$, and $\delta k[U^0]$ approaches U^0. Therefore, the first two terms of f_2^D approach zero. The last term expands to

$$\bar{T}_A \left\lfloor \frac{U^0 - \omega}{v\bar{T}_A} \right\rfloor \left(v + C_H + C_N \frac{p}{1-p}\right).$$

As $\bar{T}_A \to 0$, this approaches

$$U^0\left(1 + \frac{1}{v}\left(C_H + C_N\frac{p}{1-p}\right)\right). \quad (11)$$

Now, manipulation of Eq. (6) yields $\mathcal{V}_{\theta^\dagger}(\bar{T}_A) = f_2^D(U^0) + \bar{T}_A C_N \frac{p}{1-p}$. But as $\bar{T}_A \to 0$, the second term approaches zero. Thus $\mathcal{V}_{\theta^\dagger}(\bar{T}_A)$ approaches Eq. (11). We have proved Eq. (7).

B. Large $\bar{T}_A$

There are several cases. First, consider $\delta < \omega$ and $\bar{T}_A \geq U^0/v$. The second condition implies that D keeps A in the first honeypot that he enters until all residual utility is exhausted, which produces utility $(v + C_H)U^0/v$. The first condition implies that $U^0/v > \bar{T}_\omega$, so $\bar{T}_A > \bar{T}_\omega$, which means that D ejects A from the first normal system that he enters, which produces 0 utility. The weighted sum of these utilities gives Eq. (8). Next, consider $\delta > \omega$ and $\bar{T}_A \geq U^0/\omega$. The first condition implies that $U^0/v < \bar{T}_\omega$, so it not guaranteed that $\bar{T}_A \geq \bar{T}_\omega$. But if $\bar{T}_A \geq \bar{T}_\omega$, D ejects A from the first normal system that he enters, and we have Eq. (8).

REFERENCES

[1] National vulnerability database. [Online] Available: https://nvd.nist.gov/. [Accessed: April 2019].

[2] Massimiliano Albanese, Ermanno Battista, and Sushil Jajodia. Deceiving attackers by creating a virtual attack surface. In *Cyber Deception*, pages 169–201. Springer, 2016.

[3] Devlin Barrett, Danny Yadron, and Damian Paletta. U.S. suspects hackers in China breached about 4 million people's records, officials say. *The Wall Street Journal*, 2015. [Online] Available: https://www.wsj.com/.

[4] Richard Bellman. On the theory of dynamic programming. *Proc. Natl. Academy of Sciences*, 38(8):716–719, 1952.

[5] Thomas E Carroll and Daniel Grosu. A game theoretic investigation of deception in network security. *Security and Communication Networks*, 4(10):1162–1172, 2011.

[6] Karel Durkota, Viliam Lisỳ, Branislav Bosanskỳ, and Christopher Kiekintveld. Optimal network security hardening using attack graph games. In *Intl. Joint Conf. on Artificial Intelligence*, pages 526–532, 2015.

[7] Karel Horák, Quanyan Zhu, and Branislav Bošanský. Manipulating adversary's belief: A dynamic game approach to deception by design for proactive network security. In *Decision and Game Theory for Security*, pages 273–294. Springer, 2017.

[8] Linan Huang and Quanyan Zhu. Analysis and computation of adaptive defense strategies against advanced persistent threats for cyber-physical systems. In *Decision and Game Theory for Security*, pages 205–226. Springer, 2018.

[9] MATLAB. *R2017b*. The MathWorks Inc., Natick, Massachusetts, 2017.

[10] Keith W Miller, Jeffrey Voas, and George F Hurlburt. Byod: Security and privacy considerations. *IT Professional*, 14(5):53–55, 2012.

[11] Minghui Min, Liang Xiao, Caixia Xie, Mohammad Hajimirsadeghi, and Narayan B Mandayam. Defense against advanced persistent threats in dynamic cloud storage: A colonel blotto game approach. *IEEE Internet of Things J*, 5(6):4250–4261, 2018.

[12] Mohammad A Noureddine, Ahmed Fawaz, William H Sanders, and Tamer Başar. A game-theoretic approach to respond to attacker lateral movement. In *Decision and Game Theory for Security*, pages 294–313. Springer, 2016.

[13] United States Department of Defense. *Department of Defense Strategy for Operating in Cyberspace*. DIANE Publishing, 2012.

[14] Jeffrey Pawlick and Quanyan Zhu. Deception by design: Evidence-based signaling games for network defense. In *Workshop on the Economics of Inform. Security and Privacy*, Delft, The Netherlands, 2015.

[15] Frank J. Stech, Kristin E. Heckman, and Blake E. Strom. Integrating cyber-D&D into adversary modeling for active cyber defense. In *Cyber Deception*, pages 169–201. Springer, 2016.

[16] Chris Stokel-Walker. Hunting the DNC hackers: how Crowdstrike found proof Russia hacked the Democrats. *WIRED*, 2017. [Online] Available: http://www.wired.co.uk/.

[17] Heinrich Von Stackelberg. *Marktform und gleichgewicht*. J. Springer, 1934.

[18] Lingyu Wang, Sushil Jajodia, Anoop Singhal, Pengsu Cheng, and Steven Noel. k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *IEEE Trans. Dependable and Secure Computing*, 11(1):30–44, 2014.

[19] Liang Xiao, Dongjin Xu, Narayan B Mandayam, and H Vincent Poor. Attacker-centric view of a detection game against advanced persistent threats. *IEEE Trans. Mobile Computing*, 17(11):2512–2523, 2018.

[20] Liang Xiao, Dongjin Xu, Caixia Xie, Narayan B Mandayam, and H Vincent Poor. Cloud storage defense against advanced persistent threats: A prospect theoretic study. *IEEE J Selected Areas in Commun.*, 35(3):534–544, 2017.

[21] J. Zhuang, V. M. Bier, and O. Alagoz. Modeling secrecy and deception in a multiple-period attacker–defender signaling game. *European J Operational Res.*, 203(2):409–418, 2010.