Game-Theoretic Approach to Feedback-Driven Multi-stage Moving Target Defense

Quanyan Zhu[1] and Tamer Başar[2,⋆]

[1] Department of Electrical Engineering,
Princeton University, NJ, USA, 08544
quanyanz@princeton.edu
[2] Coordinated Science Laboratory and
Department of Electrical and Computer Engineering,
University of Illinois at Urbana Champaign,
1308 W. Main St., Urbana, IL, USA, 61801
basar1@illinois.edu

Abstract. The static nature of computer networks allows malicious attackers to easily gather useful information about the network using network scanning and packet sniffing. The employment of secure perimeter firewalls and intrusion detection systems cannot fully protect the network from sophisticated attacks. As an alternative to the expensive and imperfect detection of attacks, it is possible to improve network security by manipulating the *attack surface* of the network in order to create a *moving target defense*. In this paper, we introduce a proactive defense scheme that dynamically alters the attack surface of the network to make it difficult for attackers to gather system information by increasing complexity and reducing its signatures. We use concepts from systems and control literature to design an optimal and efficient multi-stage defense mechanism based on a feedback information structure. The change of attack surface involves a reconfiguration cost and a utility gain resulting from risk reduction. We use information- and control-theoretic tools to provide closed-form optimal randomization strategies. The results are corroborated by a case study and several numerical examples.

1 Introduction

The static nature of computing systems facilitates an attacker's capability of gathering information and executing attacks. Given sufficient amount of time, an attacker can map out the system, gain access to a node and spread to other hosts and services within the system [1]. Although heavily secured perimeter firewalls and intrusion detection systems are deployed to protect the network from outside attackers, in practice they are not effective for zero-day vulnerabilities or virus, and can be avoided by skilled attackers. In addition, while the attackers have only

⋆ This research was partially supported by an NSERC Postdoctoral Fellowship (PDF) and partially by AFOSR MURI Grant FA9550-10-1-0573, and an NSA grant through the Information Trust Institute of the University of Illinois.

S.K. Das, C. Nita-Rotaru, and M. Kantarcioglu (Eds.): GameSec 2013, LNCS 8252, pp. 246–263, 2013.

to exploit one vulnerability to be successful, a firewall has to process millions of packets every minute and perform a sophisticated and timely analysis in order to detect software that exploits a previously unknown attack vector. Clearly, an attacker has an advantage over the defender, and the sole reliance on these technologies is not sufficient for assuring security.

As an alternative to the insufficient and expensive detection of attackers, the network security can be improved by changing the appearance of the system and creating a *moving target*. The availability of services will be time-varying under different system configurations, and the system can block dangerous network behaviors if an attacker does not follow the network dynamics. In addition, in order for an attack to succeed, an attacker has to spend a significant amount of resources to carefully guide his attacks.

In this paper, we introduce stochastic dynamics into multiple layers of computing systems for securing the system by shifting its attack surface. One challenge of achieving moving target defense is to understand the tradeoff between security and usability. A complete security could be achieved by frequently changing the network and thus making it completely unusable. Hence it is essential to take into account the reconfiguration cost of shifting the surface, and the attacker's cost for learning and changing its attack vector. Moreover, the interactions between an attacker and a defender can be seen as a game [14, 19] in which the system creates a moving target for minimizing its risk and maintaining its usability, while an attacker dynamically explores and exploits a vulnerability for causing maximum damage on the system.

With this motivation, we formulate a two-person zero-sum game to model the conflict of goals, and develop a feedback learning framework to implement a moving target defense based on real-time data and observations made by the system. To achieve this goal, we first decompose the system into multiple layers as it is often composed of multiple network zones, such as those in enterprise IT network [2] and cyber-physical networks [3]. An attacker has to launch a multi-stage attack starting from network scanning and packet sniffing to illegitimate authentication and service interruption. The infamous Stuxnet virus, for example, follows a sequence of attacks depicted in Fig. 1 before compromising the supervisory control and data acquisition (SCADA) system that controls and monitors uranium enrichment infrastructure [4, 5]. We build a multi-layer game model to capture the fact that the attack is carried out through multiple stages, and the defense mechanism is developed at each layer of the system. Built into the game model, the notion of *attack surface* is defined as the set of vulnerabilities exhibited by the system that can potentially be exploited by the attacker. The essential goal of moving target strategies is to find an optimal configuration policy for the defender to shift the attack surface that minimizes its risk and the damage inflicted by an attacker.

A natural solution for the zero-sum game is mixed strategy saddle-point equilibrium (SPE), i.e., the system defender randomizes its configuration and the associated attack surface, while an attacker randomizes over the set of vulnerabilities that he can exploit. The SPE mixed-strategy pair naturally leads to a

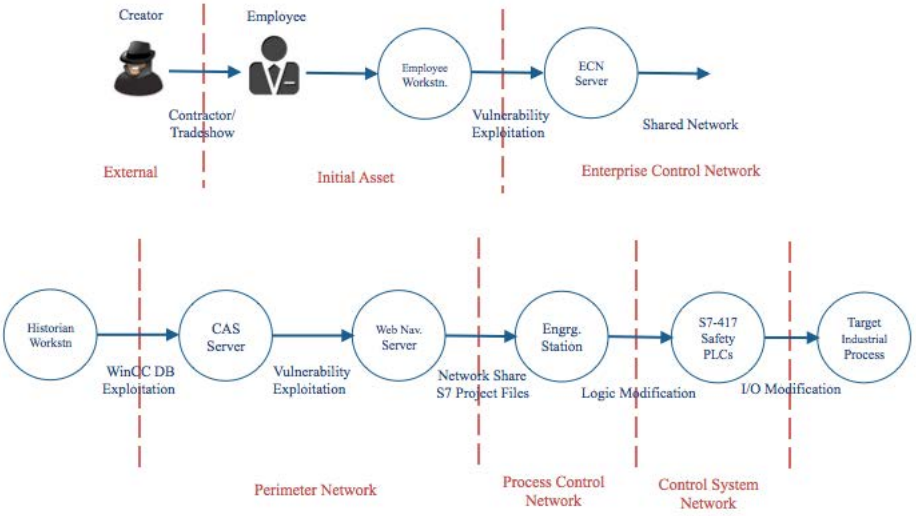

Fig. 1. Illustration of a sequence of attacks of Stuxnet: It is spread first using infected USB drive passed onto the employee. The virus exploits vulnerabilities of printer servers, WinCC database, service servers, and Siemens S7 project files at multiple stages, and propagates from employee workstation to the control system network through intermediate networks.

way of implementing moving target defense. However, it is an equilibrium solution concept which describes the steady-state outcome of a game of complete information after a sufficiently large number of repeated plays. Due to the uncertainties in the environment and limited knowledge of the players' own risk functions, it is pivotal for the system to implement a dynamic real-time defense mechanism driven by the data observed by the players. Hence we develop a feedback multi-stage defense strategy that enables the system to update its mixed strategy based on the risk it estimates online. In this case, the moving target defense is adaptive to the exogenous environment and less vulnerable than a static equilibrium strategy, which could be known to a resourceful attacker.

The contribution of this work can be summarized as follows.

(i) We formally develop a metric for quantifying attack surface, and establish a game-theoretic model for providing formal analysis of security strategies and guiding the design of moving target defense.
(ii) We develop a feedback system framework for strategically shifting attack surface based on observation.
(iii) We introduce reconfiguration and learning cost for the defender and attacker, respectively, and analyze the joint dynamics of strategy update and risk estimation.

The paper is organized as follows. Section 2 presents related works. Section 3 describes the two-person zero-sum game model and Section 4 provides a system framework for moving target defense based on learning mechanism of the game. In Section 5, we analyze the learning dynamics presented in Section 5. Section 6 illustrates the defense mechanism using numerical examples, and we conclude in Section 7.

2 Related Work

Moving target defense (MTD) is a broad class of proactive defense mechanisms, in which the defending system creates security strategies that change over time to limit the exposure of vulnerabilities and increase complexity and costs for attacks [7] and [8]. Many techniques have recently been developed for achieving MTD, including instruction set, and address space layout randomization [12], deceptive routing [23], and software diversity [13]. However, very few work has studied quantitative tradeoffs of MTD for guiding the design and analysis of the defense mechanism. In this work, we develop a feedback learning mechanism for designing MTD based on a game-theoretic framework.

Our work is related to the following existing literature. In [6], formal methods have been used to provide an attack surface metric as an indicator of the system's security. It develops an approach to reduce attack surface, which complements the software industry's traditional code quality improvement approach for security risk mitigation. Motivated by the concept, we define the notion of attack surface based on the set of existing vulnerabilities, which can be conveniently incorporated into the MTD game-theoretic model.

As depicted in Fig. 1, attackers often launch a sequence of attacks which exploits vulnerabilities at multiple stages of the system. The goal of dividing the system into multiple layers is to capture this fact. In addition, this approach is well-aligned with the research on attack graphs for assessing the cause-consequence relationships between various network states [9,10]. Based on attack graphs and trees, the system can be divided into logical layers, which correspond to a set of nodes of the same depth on a tree.

Game theory provides an appropriate theoretical framework for modeling the win-lose situation between a defender and an attacker [14]. The notion of mixed-strategy equilibrium is a natural solution concept for many applications of MTD. Our work is related to some recent works that apply game theory to MTD. In [21], a game-theoretic framework has been used to study a defense mechanism that strategically randomizes over a set of cryptographic keys that authenticate the commands from a system operator to PLC of a power system network. In [22], deceptive routing game is used to design defense strategies that mislead a jammer from attacking legitimate routes by deploying fake routes in wireless communication networks.

The feedback MTD defense mechanism is related to learning algorithms for games. In particular, the joint learning dynamics are related to the distributed learning algorithm described in [15–17], which have studied a class of distributed payoff and strategy learning for games of incomplete information. Different from

best response dynamics and fictitious play algorithms [11], the strategy updates do not require players' knowledge of their own payoff functions and the observations of actions played by others. Strategy update in the MTD defense is related to a class of imitative learning dynamics in which a player imitates the strategy of other players [24,25]. Here, we develop imitative strategies for MTD through a cost on shifting attack surface for the defender and a cost for learning system vulnerabilities and changing attack vectors for the attacker.

3 System Model

In this section, we introduce a game-theoretic framework for moving target defense between an attacker and a system defender. Many networked computing systems nowadays can be decomposed into hierarchical layers, and there are defense mechanisms residing on each of these layers. An attacker has to launch a multi-stage attack, which exploits vulnerabilities of the system at different layers, in order to compromise its final target. Hence we partition the system to be defended into a finite number of layers, and let $l = 1, 2, \cdots, N$ be the index of system layers, and $\mathcal{V}_l := \{v_{l,1}, v_{l,2}, \cdots, v_{l,n_l}\}$ be the set of n_l existing system vulnerabilities at layer l. We assume that $\mathcal{V}_l$ is common knowledge to the attacker and the system. A vulnerability $v_{l,i}$ is a weakness of the system that an adversary can exploit and launch an attack to compromise the system. The system at layer l can be configured in different ways, and each configuration exhibits its own set of vulnerabilities. Let $\mathcal{C}_l : \{c_{l,1}, c_{l,2}, \cdots, c_{l,m_l}\}$ be the set of feasible configurations of the system at layer l, and $\pi_l : \mathcal{C}_i \to 2^{\mathcal{V}_l}$ be the vulnerability map which associates with each configuration a subset of vulnerabilities. We call $\pi_l(c_{l,j})$ the *attack surface* at stage l when the system is configured to $c_{l,j}$.

At each stage l, an attacker chooses a vulnerability in the set $\mathcal{V}_l$ to exploit and launch an attack $a_{l,j}$. Let $\mathcal{A}_l := \{a_{l,1}, a_{l,2}, \cdots, a_{l,n_l}\}$ be the set of n_l attacks at stage l, and $\gamma_l : \mathcal{V}_l \to \mathcal{A}_l$ be the attack map which associates vulnerability $v_{l,j} \in \mathcal{V}_l$ with attack $a_{l,j} \in \mathcal{A}_l$. The corresponding inverse map is denoted by $\gamma_l^{-1} : \mathcal{A}_l \to \mathcal{V}_l$. Here, without the loss of generality, we assume that there is a one-to-one correspondence between $\mathcal{A}_l$ and $\mathcal{V}_l$. An attack $a_{l,j}$ can successfully cause damage on the system at layer l if the exploited vulnerability $v_{l,j}$ resides in the configuration $c_{l,i}$, i.e., $v_{l,j} \in \pi_l(c_{l,i})$.

Denote by $r_l : \mathcal{A}_l \times \mathcal{C}_l \to \mathbb{R}_+$ the damage or cost caused by the attacker at stage l, given by

$$r_l(a_{l,j}, c_{l,i}) = \begin{cases} D_{ij}, & \gamma_l^{-1}(a_{l,j}) \in \pi_l(c_{l,j}) \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where $D_{ij} \in \mathbb{R}_+$ is the (bounded) damage or risk quantified in terms of monetary values.

The goal of the attacker is to penetrate and compromise the system while the system aims to choose configurations that minimize the damage or risk. Hence we use a two-person zero-sum game to model this conflict between an attacker A and a defender S. Let Ξ_l be the game at stage l described by the triplet

$\langle\{A, S\}, \{\mathcal{A}_l, \mathcal{C}_l\}, \{r_l\}\rangle$. Since vulnerabilities are inevitable in modern computing systems, one approach for the system is to adopt moving target defense which randomizes between different system configurations, making it difficult for the attacker to learn and locate the system vulnerabilities to exploit. This naturally leads to the mixed strategy equilibrium solution concept of the game, where the defender chooses a randomized strategy $\mathbf{f}_l := (f_{l,1}, f_{l,2}, \cdots, f_{l,m_l})$ over the set $\mathcal{C}_l$, and the attacker at layer l chooses a randomized strategy $\mathbf{g}_l := (g_{l,1}, g_{l,2}, \cdots, g_{l,n_l})$ over the set $\mathcal{A}_l$, i.e.,

$$\mathbf{f}_l \in \mathcal{F}_l := \left\{ \mathbf{f}_l \in \mathbb{R}_+^{m_l} : \sum_{h=1}^{m_l} f_{l,h} = 1 \right\}, \quad \mathbf{g}_l \in \mathcal{G}_l := \left\{ \mathbf{g}_l \in \mathbb{R}_+^{n_l} : \sum_{h=1}^{n_l} g_{l,h} = 1 \right\}.$$

The game Ξ_l is a finite zero-sum matrix game with a bounded cost function. Hence there exists a mixed strategy saddle-point equilibrium (SPE) $(\mathbf{f}_l^*, \mathbf{g}_l^*)$, which satisfies the following inequality for all $\mathbf{f}_l \in \mathcal{F}_l$ and $\mathbf{g}_l \in \mathcal{G}_l$,

$$\mathbb{r}_l(\mathbf{f}_l^*, \mathbf{g}_l) \leq \mathbb{r}_l(\mathbf{f}_l^*, \mathbf{g}_l^*) \leq \mathbb{r}_l(\mathbf{f}_l, \mathbf{g}_l^*), \tag{2}$$

where $\mathbb{r}_l$ is the expected cost given by

$$\mathbb{r}_l(\mathbf{f}_l, \mathbf{g}_l) := \mathbb{E}_{\mathbf{f}_l, \mathbf{g}_l} r_l = \sum_{k=1}^{n_l} \sum_{h=1}^{m_l} f_{l,h} g_{l,k} r_l(a_{l,k}, c_{l,h}).$$

The expect cost $\mathbb{r}_l$ at SPE $(\mathbf{f}_l^*, \mathbf{g}_l^*)$ is the value of the game Ξ_l, denoted as $\text{val}(\Xi_l) = \hat{r}_l(\mathbf{f}_l^*, \mathbf{g}_l^*)$ and it is unique for zero-sum games under mixed strategies. The static analysis of the game provides an insight into the system performance (value of the game) and its strategy against an attacker if the zero-sum game is played repeatedly.

Solving the game using (2) requires that each player have complete information of the game, including the knowledge of the cost function (1) and the strategy spaces of the players. In practice, the information available to the system can be limited and sometimes uncertain. Hence the players need to learn information online and adapt their defense strategies.

Example 1. *In Fig. 2, we illustrate the multi-stage moving target defense game. At each layer l, the system has a set of vulnerabilities: $\mathcal{V}_l = \{v_{l,1}, v_{l,2}, v_{l,3}\}$, $l = 1, 2, 3, 4$. At layer 1, two configurations are feasible, i.e., $\mathcal{C}_1 = \{c_{1,1}, c_{1,2}\}$. A configuration $c_{1,1}$ is chosen in Fig. 2 and it has an attack surface $\pi_1(c_{1,1}) = \{v_{1,1}, v_{1,2}\}$. In Fig. 3, a configuration $c_{1,2}$ is chosen and it has a corresponding attack surface $\pi_1(c_{1,2}) = \{v_{1,2}, v_{1,3}\}$. Likewise, in both Fig. 2 and Fig. 3, $\pi_2(c_{2,1}) = \{v_{2,1}, v_{2,2}\}$, $\pi_3(c_{3,1}) = \{v_{3,1}, v_{3,2}\}$, and $\pi_4(c_{4,2}) = \{v_{4,2}, v_{4,3}\}$. The static system configuration $\{c_{1,1}, c_{2,1}, c_{3,1}, c_{4,2}\}$ depicted in Fig. 2 allows an attacker to launch a sequence of attacks which exploit $v_{1,1} \to v_{2,2} \to v_{2,3} \to v_{4,3}$ if sufficient amount of time and resources are given to the attacker. Mixed strategies provide a mechanism for the system defender to randomize between different configurations so that the attack surface shifts at each layer of the system. Fig. 3 depicts a scenario where the system changes its configuration to $\{c_{1,2}, c_{2,1}, c_{3,1}, c_{4,2}\}$. Then, the original attack sequence will not succeed.*

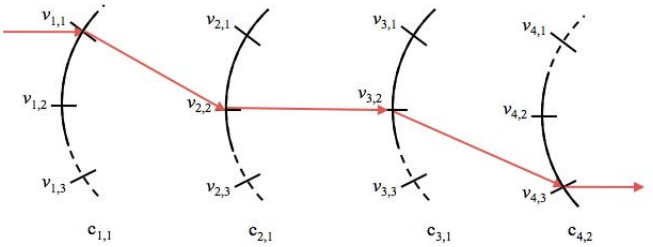

Fig. 2. A static configuration of attack surface that leads to a sequence of attacks on the physical system: An attacker can succeed in targeting the resources at the last layer by exploiting vulnerabilities $v_{1,1} \to v_{2,2}, \to v_{3,2} \to v_{4,2}$. Solid curves describe an attack surface containing existing vulnerabilities; dotted curves describe vulnerabilities circumvented by the current configuration; solid arrows refer to successful attacks.

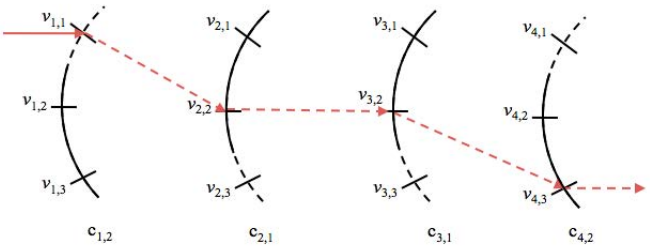

Fig. 3. Randomized configuration of attack surface that protects the system from attack exploitations at the first step. The attack is thwarted at layer 1. Solid curves describe an attack surface containing existing vulnerabilities; dotted curves describe vulnerabilities circumvented by the current configuration; dotted arrows refer to unsuccessful attacks.

Remark 1. *The attack surface at stage l for a given configuration $c_{l,h}$ is measured by $\pi_l(c_{l,h})$. We can further measure the level of vulnerability given $\mathbf{f}_l, \mathbf{g}_l$. Denote by $\eta_l = (\eta_{l,1}, \eta_{l,2}, \cdots, \eta_{l,n_l}) \in \mathcal{H}_l$ the mixed strategies for defending the set of vulnerabilities $\mathcal{V}_l$, where*

$$\mathcal{H}_l := \left\{ \eta_l \in \mathbb{R}_+^{n_l} : \sum_{h=1}^{n_l} \eta_{l,h} = 1 \right\}.$$

Suppose that vulnerabilities on the attack surface are defended with equal probabilities. Then the mixed strategy $\mathbf{f}_l$ on $\mathcal{C}_l$ leads to the following mixed strategy η_l on $\mathcal{V}_l$:

$$\eta_{l,h} = \sum_{k \in \mathcal{N}_h} \frac{f_{l,k}}{|\pi_l(c_{l,k})|}, \quad h = 1, 2, \cdots, n_l, \tag{3}$$

where $\mathcal{N}_h = \{h' \in \mathbb{Z}_+ : v_{l,h} \in \pi_l(c_{l,h'}), c_{l,h'} \in \mathcal{C}_l\}$. *The level of vulnerability* ψ *at layer* l *can be quantified by the Kullback-Leibler (K-L) divergence between two distributions* η_l, $\mathbf{g}_l$, *i.e.,*

$$\psi_l(\mathbf{f}_l, \mathbf{g}_l) := d_{KL}(\eta_l || \mathbf{g}_l) = \sum_{h=1}^{n_l} \eta_{l,h} \ln\left(\frac{\eta_{l,h}}{g_{l,h}}\right). \tag{4}$$

where d_{KL} *is the K-L divergence, and* $\eta_{l,h}$ *can be obtained from* $\mathbf{f}$ *using (3). Following Example 1, we let* $\mathbf{g}_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ *and* $\mathbf{f}_1 = (\frac{1}{2}, \frac{1}{2})$, *which randomizes between two configurations* $c_{1,1}$ *and* $c_{1,2}$. *We obtain* $\eta_1 = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$, *which leads to* $\psi = \ln\left(\frac{3\sqrt{2}}{4}\right)$. *It is clear that the system is less vulnerable under moving target defense when* ψ *is large.*

4 Moving Target Defense

Section 3 describes an ideal game model of complete information. It is common to see that the payoff function r_l can be subject to noise and disturbance, and the system/attacker cannot know each other's action spaces. This information needs to be learned over time and often there will be a cost associated with learning and adaptation. In this section, we introduce an adaptive moving target defense framework based on the system model in Section 3 in which the system dynamically updates its defense strategy through learning in an uncertain environment and without complete information. We use subscript t here to denote the strategy or cost at time t. At time t, each player independently chooses actions $\mathbb{c}_{l,t} \in \mathcal{C}_l$ and $\mathbb{a}_{l,t} \in \mathcal{A}_l$ according to strategies $\mathbf{f}_{l,t}$ and $\mathbf{g}_{l,t}$, respectively. The players cannot observe the action played by the other player but can observe the cost $r_{l,t}$ as an outcome of action pair $(\mathbb{c}_{l,t}, \mathbb{a}_{l,t})$ at time t. Based on the observed cost, the system and the attacker can estimate the average risk of the system $\hat{r}^S_{l,t} : \mathcal{C}_l \to \mathbb{R}_+$ and $\hat{r}^A_{l,t} : \mathcal{V}_l \to \mathbb{R}_+$, respectively, as follows:

$$\hat{r}^S_{l,t+1}(c_{l,h}) = \hat{r}^S_{l,t}(c_{l,h}) + \mu^S_t \mathbb{1}_{\{\mathbb{c}_{l,t}=c_{l,h}\}}(r_{l,t} - \hat{r}^S_{l,t}(c_{l,h})), \tag{5}$$

$$\hat{r}^A_{l,t+1}(a_{l,h}) = \hat{r}^A_{l,t}(a_{l,h}) + \mu^A_t \mathbb{1}_{\{\mathbb{a}_{l,t}=a_{l,h}\}}(r_{l,t} - \hat{r}^A_{l,t}(a_{l,h})). \tag{6}$$

In (5) and (6), μ^S_t and μ^A_t are payoff learning rates for the system and the attacker. Note that in the learning schemes above, the players do not know the actions played by the other players and each one estimates the average cost based on choices made in the past. The two updates are made in a distributed fashion; however, they are coupled through the fact that the observed cost depends on actions played by both players at time t.

The players can make use of the cost function learned online for updating the moving target defense strategies. The change of defense strategies from $\mathbf{f}_{l,t}$ to $\mathbf{f}_{l,t+1}$ involves a cost for the system to reconfigure by maneuvering its defense resources and altering its attack surface from $\pi_l(\mathbb{c}_{l,t})$ to $\pi_l(\mathbb{c}_{l,t+1})$, where $\mathbb{c}_{l,t}$ and $\mathbb{c}_{l,t+1}$ are selected according to distributions $\mathbf{f}_{l,t}$ and $\mathbf{f}_{l,t+1}$, respectively.

Hence we introduce the following switching cost for the system as the relative entropy between two strategies:

$$R^S_{l,t} = \epsilon^S_{l,t} \sum_{h=1}^{m_l} f_{l,h,t+1} \ln\left(\frac{f_{l,h,t+1}}{f_{l,h,t}}\right), \tag{7}$$

where $\epsilon^S_{l,t} > 0$. This cost is added onto the expected cost given by $\langle \mathbf{f}_{l,t+1}, \hat{\mathbf{r}}^S_{l,t}\rangle$, where $\langle\cdot,\cdot\rangle$ denotes the inner product between two vectors of appropriate dimensions, and $\hat{\mathbf{r}}^S_{l,t} = [\hat{r}^S_{l,t}(c_{l,1}), \hat{r}^S_{l,t}(c_{l,2}), \cdots, \hat{r}^S_{l,t}(c_{l,m_l})]'$. Hence at time t, with the learned cost vector $\hat{\mathbf{r}}^S_{l,t}$, the system solves the following system problem

$$\text{(SP)} \quad \sup_{\mathbf{f}_{l,t+1}\in\mathcal{F}_l} \langle \mathbf{f}_{l,t+1}, -\hat{\mathbf{r}}^S_{l,t}\rangle - \epsilon^S_{l,t} \sum_{h=1}^{m_l} f_{l,h,t+1} \ln\left(\frac{f_{l,h,t+1}}{f_{l,h,t}}\right). \tag{8}$$

Likewise, it takes an attacker resources (in terms of time and energy) to explore new vulnerabilities and exploit them. Hence we introduce a similar cost that capture the learning cost for the attacker

$$R^A_{l,t} = \epsilon^A_{l,t} \sum_{h=1}^{n_l} g_{l,h,t+1} \ln\left(\frac{g_{l,h,t+1}}{g_{l,h,t}}\right), \tag{9}$$

where $\epsilon^A_{l,t} > 0$. A similar problem for the attacker is

$$\text{(AP)} \quad \sup_{\mathbf{g}_{l,t+1}\in\mathcal{G}_l} \langle \mathbf{g}_{l,t+1}, \hat{\mathbf{r}}^A_{l,t}\rangle - \epsilon^A_{l,t} \sum_{h=1}^{n_l} g_{l,h,t+1} \ln\left(\frac{g_{l,h,t+1}}{g_{l,h,t}}\right). \tag{10}$$

Theorem 1. *The following statements hold for (SP) and (AP):*

(i) The following strategies $f_{l,h,t+1}$ and $g_{l,h,t+1}$ are optimal for (SP) and (AP), respectively.

$$f_{l,h,t+1} = \frac{f_{l,h,t} e^{-\frac{\hat{r}_{l,t}(c_{l,h})}{\epsilon^S_{l,t}}}}{\sum_{h'=1}^{m_l} f_{l,h',t} e^{-\frac{\hat{r}_{l,t}(c_{l,h'})}{\epsilon^S_{l,t}}}}, \tag{11}$$

$$g_{l,h,t+1} = \frac{g_{l,h,t} e^{\frac{\hat{r}_{l,t}(a_{l,h})}{\epsilon^A_{l,t}}}}{\sum_{h'=1}^{n_l} g_{l,h',t} e^{\frac{\hat{r}_{l,t}(a_{l,h'})}{\epsilon^A_{l,t}}}}. \tag{12}$$

(ii) The optimal values achieved at (11) and (12) for (SP) and (AP) are given by

$$W^S_{l,t} = \epsilon^S_{l,t} \ln\left(\sum_{h=1}^{m_l} f_{l,h,t} e^{\frac{-\hat{r}_{l,h,t}}{\epsilon^S_{l,t}}}\right), \tag{13}$$

$$W^A_{l,t} = \epsilon^A_{l,t} \ln\left(\sum_{h=1}^{n_l} g_{l,h,t} e^{\frac{\hat{r}_{l,h,t}}{\epsilon^A_{l,t}}}\right). \tag{14}$$

Sketch of Proof. The results are obtained by directly solving two concave constrained optimization problems. A complete proof can be found in [26]. □

The parameters $\epsilon^S_{l,t}$ and $\epsilon^A_{l,t}$ model the switching and learning costs for the system and the attacker respectively. The parameter values are also related to the degree of rationality of the players associated with the zero-sum game Ξ_l. When $\epsilon^S_{l,t}, \epsilon^A_{l,t}$ are close to zero, the players tend to be highly rational, whereas they are irrational when the parameters go to infinity. This observation is summarized in the following theorem.

Theorem 2. *The following statements hold for (SP) and (AP):*

(i) The players are of high rationality when $\epsilon^S_{l,t}$ and $\epsilon^A_{l,t}$ approach 0, i.e.,

$$\lim_{\epsilon^S_{l,t}\to 0} W^S_{l,t} = \min_{c_{l,h}\in\mathcal{N}_l} \hat{r}^S_{l,h,t} \tag{15}$$

$$\lim_{\epsilon^A_{l,t}\to 0} W^A_{l,t} = \max_{a_{l,h}\in\mathcal{A}_l} \hat{r}^A_{l,h,t} \tag{16}$$

(ii) The players are of low rationality when when $\epsilon^S_{l,t}$ and $\epsilon^A_{l,t}$ approach $+\infty$, i.e.,

$$\lim_{\epsilon^S_{l,t}\to\infty} W^S_{l,t} = \langle \mathbf{f}_{l,t}, -\hat{\mathbf{r}}^S_{l,t}\rangle \tag{17}$$

$$\lim_{\epsilon^A_{l,t}\to\infty} W^A_{l,t} = \langle \mathbf{g}_{l,t}, \hat{\mathbf{r}}^A_{l,t}\rangle \tag{18}$$

Sketch of Proof. The results follow directly from taking limits of the closed form solution of $W^S_{l,t}$ and $W^A_{l,t}$ in Theorem 1(ii). A complete proof can be found in [26]. □

The optimization problems (SP) and (AP) provide a mechanism for players to update their mixed strategies at time $t+1$ based on the information learned at time t. The defender will choose a new configuration $_{l,t+1}$ according to $\mathbf{f}_{l,t+1}$ to alter the attack surface from $\pi_l(\mathbb{c}_{l,t})$ to $\pi_l(\mathbb{c}_{l,t+1})$.

Fig. 4 summarizes the feedback-driven moving target defense from the defender's perspective. The system updates the mixed strategies for moving target defense by solving (SP) based on the risked learned online using (5). The defender reconfigures the system according to (11) and shifts the attack surface to minimize the damage and risk of the system. An intelligent attacker on the other hand can also follow the same procedure to explore and exploit existing vulnerabilities of the system.

Remark 2. *Note that there is a clear relation between the defense mechanism depicted in Fig. 4 and feedback control systems. The risk learning can be seen as a sensor which measures outputs of the system and estimates the system state. The "Shift Attack Surface" block in the diagram can be regarded as an actuator which sends input to command and control the system. The design of moving target defense is analogous to a feedback controller design. Acknowledging this connection allows to apply control-theoretic tools to analyze the system.*

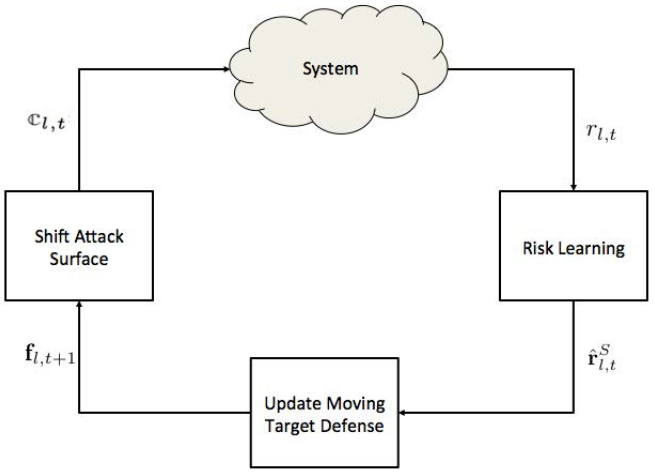

Fig. 4. System framework of the moving target defense: The defender learns online the risk of the system and updates the mixed strategy for moving target defense. The system shifts its attack surface according to the updated defense strategy.

Remark 3. *Compared to existing moving target defense which adopts a static distribution to randomize the attack surface, the defense mechanism described in Fig. 4 provides another layer of defense by changing the mixed strategy (or distribution) over time according to the environment. In this way, it creates more complexity and higher cost for an attacker to learn and gain system information.*

5 Learning Dynamics

In this section, we analyze the feedback moving target defense mechanism described in Section 4. The dynamics for mixed strategy update using (11) and (12) can be generalized by taking a convex combination of (11), (12) and the previous mixed strategy. They are described by

$$f_{l,h,t+1} = (1-\lambda^S_{l,t}) f_{l,h,t} + \lambda^S_{l,t} \left(\frac{f_{l,h,t} e^{-\frac{\hat{r}_{l,t}(c_{l,h})}{\epsilon^S_{l,t}}}}{\sum_{h'=1}^{m_l} f_{l,h',t} e^{-\frac{\hat{r}_{l,t}(c_{l,h'})}{\epsilon^S_{l,t}}}} \right), \tag{19}$$

$$g_{l,h,t+1} = (1-\lambda^A_{l,t}) f_{l,h,t} + \lambda^A_{l,t} \left(\frac{g_{l,h,t} e^{\frac{\hat{r}_{l,t}(a_{l,h})}{\epsilon^A_{l,t}}}}{\sum_{h'=1}^{n_l} g_{l,h',t} e^{\frac{\hat{r}_{l,t}(a_{l,h'})}{\epsilon^A_{l,t}}}} \right), \tag{20}$$

where $\lambda^S_{l,t}, \lambda^A_{l,t} \in [0,1]$ are learning rates. If $\lambda^S_{l,t} = \lambda^A_{l,t} = 1$, the dynamics correspond to the update procedure in Section 4. Note that the dynamics (19) and (20) are coupled with cost learning in (5) and (6).

The convergence of the coupled dynamics can be studied by its corresponding continuous-time dynamics. Let $e_{c_{l,h}} \in \mathcal{F}_l, e_{a_{l,h}} \in \mathcal{G}_l$ be a vector of proper dimension with $h-$th entry being 1 and others being 0, and all the variables as a continuous function of time. We let the learning rates satisfy the following conditions:

$$\sum_{t\geq 1} \lambda^S_{l,t} = +\infty, \quad \sum_{t\geq 1} (\lambda^S_{l,t})^2 < +\infty, \quad \sum_{t\geq 1} \lambda^A_{l,t} = +\infty, \quad \sum_{t\geq 1} (\lambda^A_{l,t})^2 < +\infty;$$

$$\sum_{t\geq 1} \mu^S_{l,t} = +\infty, \quad \sum_{t\geq 1} (\mu^S_{l,t})^2 < +\infty, \quad \sum_{t\geq 1} \mu^A_{l,t} = +\infty, \quad \sum_{t\geq 1} (\mu^A_{l,t})^2 < +\infty.$$

Theorem 3. *The joint cost and strategy learning algorithms (5) and (19), (6) and (20) converge to the following set of ordinary differential equations (ODEs).*

(i) System defender's dynamics

$$\begin{cases} \frac{d}{dt} f_{l,h,t} = f_{l,h,t} \left(\frac{e^{\frac{\hat{r}_{l,t}(c_{l,h})}{\epsilon^S_{l,t}}}}{\sum_{h'=1}^{m_l} f_{l,h',t} e^{\frac{\hat{r}_{l,t}(c_{l,h'})}{\epsilon^S_{l,t}}}} - 1 \right), \quad h = 1, 2, \cdots, m_l, \\ \frac{d}{dt} \hat{r}^S_{l,t}(c_{l,h}) = -\mathbb{r}_{l,t}(e_{c_{l,h}}, \mathbf{g}_{l,t}) - \hat{r}^S_{l,t+1}(c_{l,h}), \quad c_{l,h} \in \mathcal{C}_l \end{cases} \tag{21}$$

(ii) Attacker's dynamics

$$\begin{cases} \frac{d}{dt} g_{l,h,t} = g_{l,h,t} \left(\frac{e^{\frac{\hat{r}_{l,t}(a_{l,h})}{\epsilon^A_{l,t}}}}{\sum_{h'=1}^{n_l} g_{l,h',t} e^{\frac{\hat{r}_{l,t}(a_{l,h'})}{\epsilon^A_{l,t}}}} - 1 \right), \quad h = 1, 2, \cdots, n_l, \\ \frac{d}{dt} \hat{r}^A_{l,t}(a_{l,h}) = \mathbb{r}_{l,t}(\mathbf{f}_{l,t}, e_{a_{l,h}}) - \hat{r}^A_{l,t+1}(a_{l,h}), \quad a_{l,h} \in \mathcal{A}_l \end{cases} \tag{22}$$

Sketch of Proof. Under the assumptions of the learning rates, the results can be shown using stochastic approximation techniques, [27]. □

Theorem 4. *Let $\epsilon^S_{l,t} = \epsilon^A_{l,t} = \epsilon_l$ for all t. The following statements hold.*

(i) SPE of the game Ξ_l are steady states of the dynamics (21) and (22).
(ii) The interior stationary points of the dynamics are SPE of the game Ξ_l.

Sketch of Proof. The results follow from the properties of imitative Boltzmann-Gibbs dynamics ([25] and the references therein). A complete proof can be found in [26]. □

Remark 4. *The adaptive moving target defense illustrated in Fig. 4 can be employed at each layer of the system independently. This is reasonable if attackers can explore and exploit vulnerability at each layer of the system simultaneously. For the case where the attacker has to launch a stage-by-stage attack, the game-theoretic model in Section 3 can be extended to a stochastic dynamic game framework to capture the fact that the outcome of the game at layer l leads to the game at layer $l+1$. In the stochastic game model, the transition probabilities can be taken to be attacker's success probabilities at each stage, and the systematic risk can be taken to be the aggregate potential damage on the system across all the layers.*

6 Numerical Example

In this section, we illustrate the feedback-driven moving target defense within the context of Example 1. We let $\mathcal{V}_1 = \{v_{1,1}, v_{1,2}, v_{1,3}\}$ be the set of three vulnerabilities at layer 1. They lead to a low (L), medium (M), and high (H) level of damage on the system, respectively, if it is compromised. The attacker can choose to exploit each of these vulnerabilities and launch an attack. The set of attacks is given by $\mathcal{A}_1 = \{a_{1,1}, a_{1,2}, a_{1,3}\}$ with attack $a_{l,h}$ corresponding to vulnerability $v_{l,h}$, $h = 1, 2, 3$. The success of an attack $a_{1,h}$ will result costs of 1p.u., 2p.u., 3p.u. of damage, respectively. Hence the game matrix can be written as follows:

Ξ_1 :

	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$c_{1,1}$	1	2	0
$c_{1,2}$	0	2	3

The column player is the attacker (minimizer) and the row player is the defender (maximizer). The game has a pure Nash equilibrium where the attacker chooses $a_{1,2}$ while the defender chooses $c_{1,2}$, which results in the value 2 for the game. We set $\epsilon^S_{l,t} = \epsilon^A_{l,t} = \frac{1}{30}$. Fig. 5 and Fig. 6 illustrate the strategy and cost update dynamics (21) for the system defender. Fig. 7 and Fig. 8 illustrate the strategy and cost update dynamics (22) for an intelligent attacker. We see that the dynamics converge to a Nash equilibrium of the matrix game Ξ_1.

The numerical examples above have illustrated the convergence properties of the learning algorithm when the system and the attacker both adopt the same learning mechanism. In practice, two players can follow different dynamics, and the observations made by the system are noisy. Hence, the equilibrium of the game may not be attained as in Figs. 5 and 7. The distributed nature of the feedback system in Section 4 provides nevertheless a convenient framework for a defender to respond to its own observations. We let the mixed strategy of the attacker $\mathbf{g}_t$ be an i.i.d. random process, where $\mathbf{g}_t$ is uniformly chosen from $\mathcal{G}_1$. Assume that the payoff matrix Ξ_1 is subject to an additive noise v_1, which is uniformly distributed on $[0, 1/2]$. In Fig. 9, we show for different values of ϵ, the evolution of mixed strategy generated by (19) of the system when the attacker behaves randomly, while the system optimally switches between two configurations. When ϵ is the large, it is more costly for the system to change its

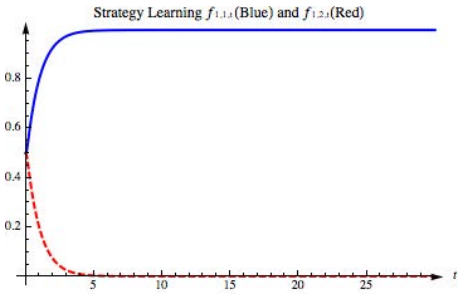

Fig. 5. System's strategy learning: Continuous-time evolution of defender's mixed-strategies $f_{1,1,t}$ and $f_{1,2,t}$

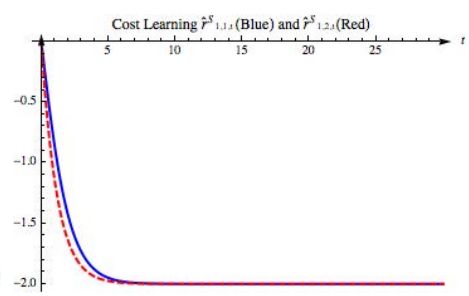

Fig. 6. System's payoff learning: Continuous-time evolution of defender's estimated cost $\hat{r}^S_{1,1,t}$ and $\hat{r}^S_{1,2,t}$

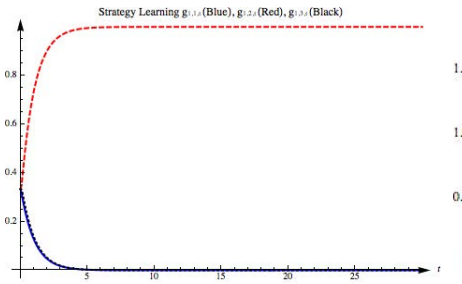

Fig. 7. Attacker's payoff learning: Continuous-time evolution of attacker's mixed strategies $g_{1,1,t}$, $g_{1,2,t}$, and $g_{1,3,t}$

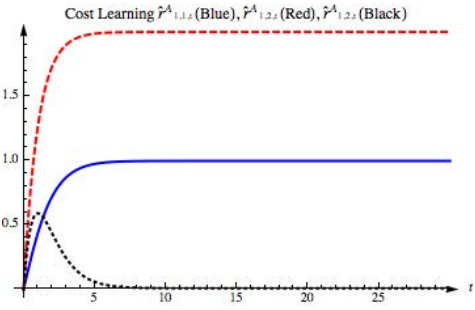

Fig. 8. Attacker's payoff learning: Continuous-time evolution of attacker's estimated payoff $\hat{r}^A_{1,1,t}$, $\hat{r}^A_{1,2,t}$, and $\hat{r}^S_{1,3,t}$

attack surface regularly. Hence, the evolution of mixed strategy $f_{1,1,t}$ is smoother than the one with smaller ϵ.

In Fig. 10, we show the risk measured by the defender, which depends on attacker's action and defender's attack surface. From $t = 99$ to $t = 115$, the system sees an unusual peak of risk under $c_{1,2}$. This exogenous input is used to model unexpected malicious events or alerts that have been detected by or alerted to the system due to the potential risk imposed by $v_{1,1}$. In Fig. 9, we can see that the randomized strategy $f_{1,1}$ reacts to the surge of risk at $t = 99$ and the probability of choosing $c_{1,1}$ starts to increase until the alert is over. The mixed strategy at steady state is found to be $\mathbf{f}_1 = (0.61, 0.39)$. The feedback mechanism allows the defense system to adapt to unanticipated events and enable emergency response that enhances the resiliency of the system.

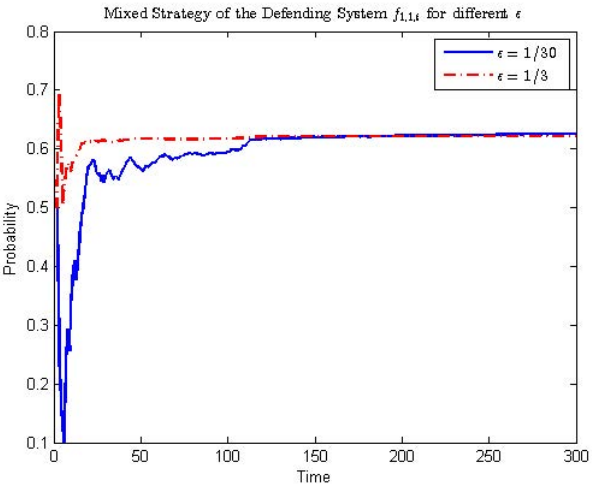

Fig. 9. Mixed strategy of the defending system for different values of ϵ: $f_{1,1,t}$ is the probability of choosing configuration $c_{1,1}$, and $f_{1,2,t} = 1 - f_{1,1,t}$ is the probability of choosing $c_{1,2}$

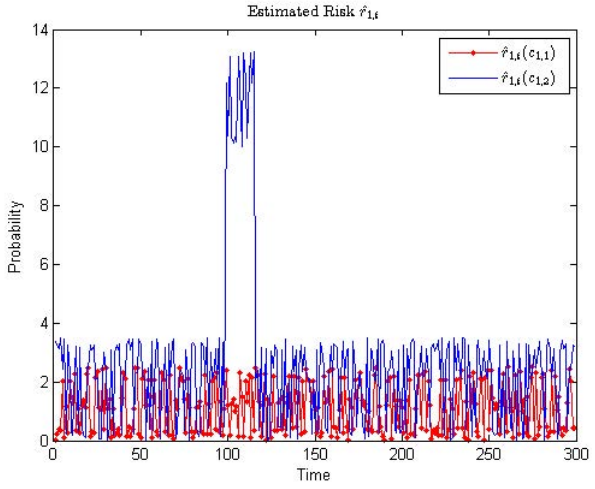

Fig. 10. Risk estimated by the defender for employing configurations $c_{1,1}$ and $c_{1,2}$, respectively. An unexpected event is detected during the period from $t = 99$ to $t = 115$.

In Fig. 11, the vulnerability metric ψ is computed at each time instant. We observe that the moving target defense outperforms a static randomized strategy $(1/3, 1/3, 1/3)$ as its ψ value is constantly higher than the one under the static strategy. The feedback mechanism provides a way to misalign the vulnerability the attack intend to exploit and the vulnerabilities on system's attack surface.

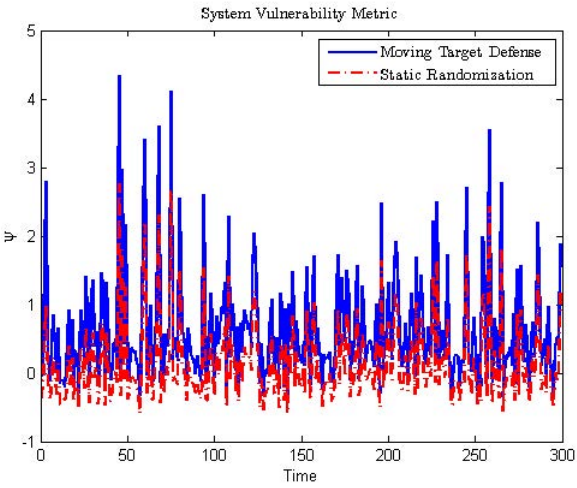

Fig. 11. System vulnerability metric ψ measures the level of vulnerability when the system uses $\mathbf{f}_1$ to defend against an attacker who adopts $\mathbf{g}_1$. A higher value of ψ indicates a lower level of vulnerability. In comparison to a static randomized strategy $\mathbf{f}_1 = (1/3, 1/3, 1/3)$, the moving target defense yields a better performance.

7 Conclusions

Moving target defense is an alternative solution to the current expensive and imperfect detection of an intelligent attacker. It is a defense mechanism that dynamically varies the attack surface of the system being defended, and provides probabilistic protections despite exposed vulnerabilities. In this paper, we have developed a game-theoretic framework for guiding the quantitative design of moving target defense as a tradeoff between security and usability. Based on the model, we have proposed a feedback mechanism that allows the system to monitor its current system state and update its randomized strategy based on its observation. We have analyzed the equilibrium stochastic joint strategy and payoff dynamics by studying its associated continuous-time dynamics. As discussed in Remark 4, this work could be further extended to a stochastic game framework where transition probabilities between games capture strategy interdependencies across the layers. Instead of finding equilibrium for games at each layer, we would be interested in a security policy that minimizes the overall risk of the multi-layer system.

References

1. Bowers, K.D., van Dijk, M., Griffin, R., Juels, A., Oprea, A., Rivest, R.L., Triandopoulos, N.: Defending against the unknown enemy: Applying FlipIt to system security. In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 248–263. Springer, Heidelberg (2012)

2. Department of Energy, "Control systems cyber security: defense in depth strategies," External Report # INL/EXT-06-11478, http://energy.gov/sites/prod/files/oeprod/DocumentsandMedia/Defense_in_Depth_Strategies.pdf
3. Zhu, Q., Başar, T.: A hierarchical security architecture for the smart grid. In: Hossain, E., Han, Z., Poor, H.V. (eds.) Smart Grid Communications and Networking, Cambridge University Press (2012)
4. Byres, E., Ginter, A., Langill, J.: "How Stuxnet spreads – A study of infection paths in best practice systems," White Paper, Tofino Security (February 22, 2011)
5. Falliere, N., Murchu, L.O., Chien, E.: "W32. Stuxnet Dossier," Symantec Reports (February 2011)
6. Manadhata, P.K., Wing, J.M.: An attack surface metric. IEEE Trans. on Software Engineering 37(3), 371–386 (2011)
7. Jajodia, S., Ghosh, A.K., Swarup, V., Wang, C., Wang, X.S.: Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats. In: Advances in Information Security. Springer (2011)
8. Jajodia, S., Ghosh, S.K., Subrahmanian, V.S., Swarup, V., Wang, C., Wang, X.S.: Moving Target Defense II: Application of Game Theory and Adversarial Modeling. In: Advances in Information Security. Springer (2012)
9. Poolsappasit, N., Dewri, R., Ray, I.: Dynamic Security Risk Management Using Bayesian Attack Graphs. IEEE Transactions on Dependable and Secure Computing 9(1), 61–74 (2012)
10. Ten, C.-W., Liu, C.-C., Manimaran, G.: Vulnerability assessment of cybersecurity for SCADA systems using attack trees. In: Proc. IEEE Power Eng. Soc. Gen. Meeting, Tampa, FL, June 24-28, pp. 1–8 (2007)
11. Fudenberg, D., Levine, D.K.: The Theory of Learning in Games. The MIT Press (1998)
12. Kc, G.S., Keromytis, A.D., Prevelakis, V.: Countering code-injection attacks with instruction-set randomization. In: Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003), New York, NY, USA, pp. 272–280 (2003)
13. Neti, S., Somayaji, A., Locasto, M.E.: Software diversity: security, entropy and game theory. In: Proceedings of the 7th USENIX Conference on Hot Topics in Security, HotSec 2012 (2012)
14. Manshaei, M.H., Zhu, Q., Alpcan, T., Başar, T., Hubaux, J.-P.: Game theory meets network security and privacy. ACM Computing Survey 45(3), 25:1–25:39 (2013)
15. Zhu, Q., Tembine, H., Başar, T.: Hybrid learning in stochastic games and its applications in network security. In: Lewis, F., Liu, D. (eds.) Reinforcement Learning and Approximate Dynamic Programming for Feedback Control, ch. 14. Computational Intelligence Series, pp. 305–329. IEEE Press, Wiley (2013)
16. Zhu, Q., Tembine, H., Başar, T.: Distributed strategic learning with application to network security. In: Proc. 2011 American Control Conference (ACC 2011), San Francisco, CA, June 29-July 1, pp. 4057–4062 (2011)
17. Zhu, Q., Tembine, H., Başar, T.: Heterogeneous learning in zero-sum stochastic games with incomplete information. In: Proc. 49th IEEE Conference on Decision and Control (CDC 2010), Atlanta, Georgia, December 15-17, pp. 219–224 (2010)
18. Zhu, Q., Clark, A., Poovendran, R., Başar, T.: Deceptive routing games. In: Proc. 51st IEEE Conference on Decision and Control (CDC 2012), Maui, Hawaii, December 10-13 (2012)
19. Başar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory. SIAM Series in Classics in Applied Mathematics (January 1999)

20. Zhu, Q., Başar, T.: Dynamic policy-based IDS configuration. In: Proc. 48th IEEE Conference on Decision and Control (CDC 2009), Shanghai, China, December 16-18 (2009)
21. Clark, A., Zhu, Q., Poovendran, R., Başar, T.: An impact-aware defense against Stuxnet. In: Proc. 2013 American Control Conference (ACC 2013), Washington, DC, June 17-19, pp. 4146–4153 (2013)
22. Zhu, Q., Clark, A., Poovendran, R., Başar, T.: Deceptive routing games. In: Proc. 51st IEEE Conference on Decision and Control (CDC 2012), Maui, Hawaii, December 10-13, pp. 2704–2711 (2012)
23. Clark, A., Zhu, Q., Poovendran, R., Başar, T.: Deceptive routing in relay networks. In: Grossklags, J., Walrand, J. (eds.) GameSec 2012. LNCS, vol. 7638, pp. 171–185. Springer, Heidelberg (2012)
24. Sandholm, W.H.: Excess payoff dynamics and other well-behaved evolutionary dynamics. Journal of Economic Theory 124(2), 149–170 (2005)
25. Weibull, J.W.: Evolutionary game theory. MIT Press (1997)
26. Zhu, Q., Başar, T.: "Feedback-Driven Multi-Stage Moving Target Defense", CSL Technical Report
27. Borkar, V.S.: Stochastic approximation: A dynamical systems viewpoint. Cambridge University Press (2008)
28. Franklin, G.F., Powell, D.J., Emami-Naeini, A.: Feedback Control of Dynamic Systems, 5th edn. Prentice Hall PTR, Upper Saddle River (2001)