Reinforcement Learning Algorithms for Adaptive Cyber Defense against Heartbleed

Minghui Zhu
Pennsylvania State University
University Park, PA, USA
16802
muz16@psu.edu

Zhisheng Hu
Pennsylvania State University
University Park, PA, USA
16802
hzsxiaoyi@gmail.com

Peng Liu
Pennsylvania State University
University Park, PA, USA
16802
pliu@ist.psu.edu *

ABSTRACT

In this paper, we investigate a model where a defender and an attacker simultaneously and repeatedly adjust the defenses and attacks. Under this model, we propose two iterative reinforcement learning algorithms which allow the defender to identify optimal defenses when the information about the attacker is limited. With probability one, the adaptive reinforcement learning algorithm converges to the best response with respect to the attacks when the attacker diminishingly explores the system. With a probability arbitrarily close to one, the robust reinforcement learning algorithm converges to the min-max strategy despite that the attacker persistently explores the system. The algorithm convergence is formally proven and the algorithm performance is verified via numerical simulations.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Algorithms, security

Keywords

Security, adaptive cyber defense, reinforcement learning

1. INTRODUCTION

Information technology (IT) systems are indispensable for the efficient operation of our society. Since IT systems are inherently vulnerable to malicious attacks, their security has been attracting significant attention in both academia and industry. At the system level, the security of IT systems can be viewed as the interactions between a defender and an attacker. The goals of defenders and attackers are completely opposite. By exploiting the vulnerabilities of IT systems, the attacker aims to intrude IT systems and achieve illegitimate goals; e.g., stealing financial reports. On the other hand, the defender aims to deploy suitable defenses in order to minimize the damages caused by malicious attacks and maintain the quality-of-service of IT systems.

Current cyber defenses are mostly static and demand a long process to reconfigure if any. So the attackers have a sufficient amount of time to explore and further exploit the vulnerabilities of IT systems. The asymmetry between the defender and attacker motivates recent study of adaptive cyber defense (ACD). In ACD, the defender dynamically reconfigures cyber defenses and increases complexity and costs for the attacker. Some typical ACD schemes include moving target defenses (MTD), artificial diversity defense and bio-inspired defenses [10, 11, 18].

ACD is a challenging problem. In adversarial environments, it is difficult for the defender to access to the private information of the attacker; e.g., his/her targets, resources, intentions, preferences and launched attacks. The limited information about the opponent prevents the defender from choosing optimal defenses. That is, the defender may spend substantial time and effort to protect an asset which may not be the target of the attacker.

In this paper, we investigate a model where a defender and an attacker simultaneously and repeatedly adjust the defenses and attacks. Time instants are referred to as the collection of events when the defender and attacker perform the adjustments. The defender is accessible to the deployed defenses, the classes the attackers belong to and the system utilities jointly affected by the defenses and attacks. However, the defender is unaware of launched attacks and attacking policies. This model can characterize several classes of attacks; e.g., buffer over-read attacks (e.g., Heartbleed), buffer over-run attacks and code reuse attacks (e.g., ROP).

Contributions. Under the above model, we propose two iterative reinforcement learning algorithms which allow the defender to identify optimal defenses.

- We first come up with the adaptive reinforcement learning algorithm. At each time instant, the defender, on one hand, reinforces the successful defense in the last two time instants with a certain exploitation rate, and on the other hand, randomly chooses any feasible defense. By using stochastic stability of Markov chains, we formally prove that, with probability one,

*M. Zhu and Z. Hu are partially supported by ARO W911NF-13-1-0421 (MURI). P. Liu is partially supported by ARO W911NF-13-1-0421 (MURI), AFOSR W911NF1210055 and NSF CNS-1422594.

MTD'14, November 3, 2014, Scottsdale, Arizona, USA.

http://dx.doi.org/10.1145/2663474.2663481.

the defense converges to the best response of the attacks when the exploration rate of the attacker is diminishing.

- We then propose the robust reinforcement learning algorithm for the case when the exploration rate of the attacker is persistent. At each time instant, the defender, on one hand, reinforces the defense which minimizes the maximal utility in the whole observed history with a certain exploitation rate, and on the other hand, randomly chooses any feasible defense. We formally prove that, with a probability arbitrarily close to one, the defense converges to the min-max strategy despite that the attacker could persistently explore the system.

2. RELATED WORK

Non-cooperative game theory has been widely used to model the interdependency of defenders and attackers in; e.g., [12, 13, 14, 15, 16, 19, 21]. A number of algorithms have been proposed to compute Nash equilibrium and further infer the actions of attackers in; e.g., [2]. Security under limited information is considered in; e.g., [12, 26]. For the papers aforementioned, a key assumption is that attackers are rational and strategic decision makers and willing to play the strategies corresponding to Nash equilibrium. In addition, in non-cooperative games, pure Nash equilibrium may not exist and mixed Nash equilibrium is difficult to implement in practical IT systems. In contrast, we do not have specific attacker models; e.g., utility functions, and do not formulate the problem as a non-cooperative game. In this regard, our problem formulation is similar to [5, 6, 8].

The algorithms proposed in this paper are based on reinforcement learning [4, 20]. In particular, we utilize the idea of exploration and exploitation to reinforce the defenses which are successful in the observed history. Reinforcement learning is particularly well suited to problems where decision makers interact with unknown environments but can evaluate their actions via induced outcomes. Reinforcement learning has been applied successfully to various problems, including robot control, elevator scheduling, telecommunications, checkers and target assignment [20]. Recently, reinforcement learning has been extended to compute Nash equilibrium of non-cooperative games; e.g., [17, 23, 24, 25, 26], with applications to sensor coverage and network security. Our work is also related to multi-armed bandit problems [3, 7] in reinforcement learning where the decision-maker aims to maximize the sum of rewards earned through a sequence of decisions when the rewards are randomly generated according to unknown distributions. This is the most fundamental example of sequential decision problems with an exploration-exploitation trade-off. In contrast, our problem is not a sequential decision problem and our goal is to identify the optimal defenses instead of minimizing the regret over a finite-horizon or infinite-horizon decision process.

3. PROBLEM FORMULATION

3.1 The system model

In this section, we present an abstract model of the IT system. In Section 3.3, we will use the Heartbleed to exemplify the model. In particular, the defender has a set of available defenses which is denoted by $\mathcal{D} \triangleq \{d_1, \cdots, d_n\}$. And the attacker has a set of available attacks denoted by $\mathcal{A} \triangleq \{a_1, \cdots, a_m\}$. They jointly influence the IT system. Given any pair of $(d, a) \in \mathcal{D} \times \mathcal{A}$, the scalar $U(d, a)$ represents the utility received by the defender.

In the real world, the defender and attacker make decisions and take actions asynchronously instead of simultaneously or alternatively. The asynchronous decision-making process is depicted in Figure 1 and described as follows. Once receiving an IDS alert, the defender performs meta-analysis on the attacks occurred since the last defense deployment, computes the corresponding utility and then deploys a new defense if necessary. This is referred to as a defense cycle. On the other hand, the attacker evaluates the consequences of previous attacks and launches a new one if needed. This is referred to as an attack cycle. The frequency of attack cycles is usually larger than that of defense cycles, and thus the defense-attack course presents two time-scales. In the paper, we investigate the security from the point of view of the defender, and focus on defense deployments in contrast to attack launches. For the ease of presentation, we make the following assumptions on the model:

- The time for each defense and attack cycle is very little and negligible.
- The time horizon is divided according to the defense cycles. That is, time instant t represents the t^{th} defense cycle. At each time instant t, the decision-making of the defender is based on the meta-analysis of attacks between $t-1$ and t. This is shown in Figure 2.
- The decision-making of the defender and attacker is synchronized.
- There is not intrusion detection delays.

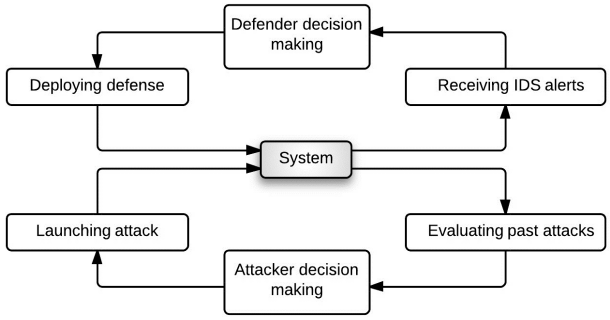

Figure 1: The interactions of the defender and attacker.

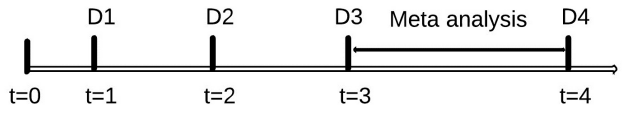

Figure 2: The time horizon of defense deployment.

3.2 Adaptive cyber defense

In practice, it is difficult for the defender to access to the private information of the attacker; e.g., his/her targets, resources, intentions, preferences and launched attacks. That

is, it is difficult for the defender to establish a precise model of the attacker. The limited information about the opponent model prevents the defender from choosing proper defenses to minimize the utilities. The defender may spend substantial time and effort to protect an asset which may not be the target of the attacker. Our paper focus is to synthesize and analyze the algorithms for the defender to deploy optimal defenses where the defender is only accessible to the deployed defenses and the classes the attackers belong to.

Here we would like mention that, in the real world, the attacker may follow certain strategies to explore the IT system and launch attacks. So there could be an underlining model of the attacker. However, the defender just may not be able to know it. In order to broaden the applicability of our algorithms, we do not restrict that the attacker has to follow certain strategies and instead treat attacking strategies as black boxes.

3.3 A motivating example

The system model in Section 3.1 can characterize several classes of attacks; e.g., buffer over-read attacks (e.g., Heartbleed), buffer over-run attacks and code reuse attacks (e.g., ROP). In what follows, we will use the Heartbleed [1] to further exemplify the model. The Transport Layer Security (TLS) protocol maintains communication links between a client and a server by allowing the client to send a Heartbeat request to the server. The Heartbeat request consists of a message type, a payload's length, a payload and a padding. The affected versions of OpenSSL allocate a memory buffer for the message to be returned based on the payload length in the client request, regardless the actual payload length in the request. Because of the failure to do proper bounds checking, the message returned by the server consists of the payload, possibly followed by whatever else in the memory buffer.

Let us consider a heap in Figure 3. The heap consists of n pages and each page P_i contains B_i bytes data. A data object consists of a subset of pages. In particular, when a data object is born, function malloc() assigns several pages in freelist to the data object. When a data object dies, function free() returns its pages to freelist. At each time instant t, the malicious client (attacker) sends a heartbeat request $a(t) = (p(t), b(t))$ to the server (defender) where the request is characterized by the number $p(t)$ of the page he/she wants to visit and the bytes $b(t)$ of data he/she requests the server to return where the returned data starts from the beginning of page $p(t)$. By incident, the attacker may send some request $(p(t), b(t))$ such that the serves returns the data which does not belong to the allowable data object. Figure 3 is an example of a successful Heartbleed attack. The attacker is unaware which combinations of $p(t)$ and $b(t)$ may succeed. So the attacker may try different combinations and adjust his/her attacking strategy on basis of the data returned from the server. Note that a large $b(t)$ increases the likelihood that the attacker is detected. So, choosing the maximum bit $64K$ for $b(t)$ may not be the best decision for the attacker.

On the other hand, the server can choose to guard a subset of pages to prevent the Heartbleed attacks. The defense at time instant t is denoted by $d(t) \subseteq \{P_1, \cdots, P_n\}$. Figure 4 shows a defense example. When page P_3 is guarded, its subsequent pages cannot be visited when the request starts from page P_1 or P_2. If guarded pages are touched by Heartbleed requests, intrusion detection system (IDS) is triggered through a segmentation fault. Then IDS performs meta analysis, checks system log and determines how many heartbleed requests have been sent to the server since last intrusion detection. The number of heartbleed requests depends on both of attack and defense and is denoted by $I(t) = I(a(t), d(t))$. If the defender guards more pages, then the number of heartbleed requests is smaller. However, if more pages are guarded, the response time of the IT system becomes longer. The response time penalty induced by $d(t)$ is denoted by $c(d(t))$. The tradeoff between detection and response time is modeled by the utility function $U(a, d) = c(d) - I(a, d)$. Based on the observed utilities, the defender dynamically reconfigures the defenses and aims to identify one to minimize the utility.

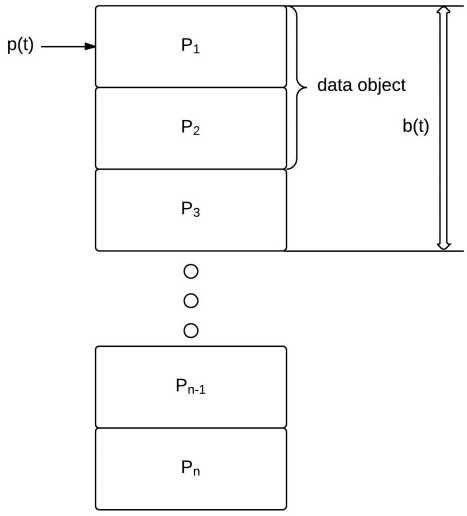

Figure 3: An illustration of successful Heartbleed attacks.

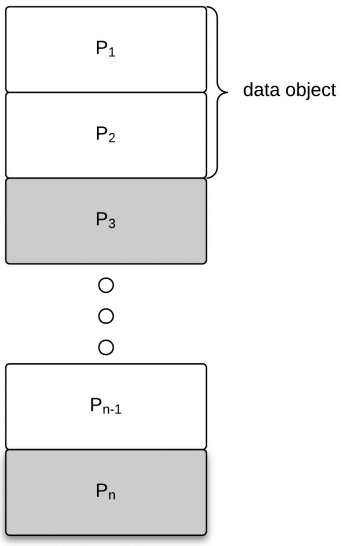

Figure 4: An illustration of defenses where the gray pages are guarded.

4. ADAPTIVE REINFORCEMENT LEARNING ALGORITHM

As discussed in Section 3.3, during the attacking course, the attacker may try different attacking strategies, explore the vulnerabilities of the IT system, learn from previous attacks and correspondingly adjust the attacking strategy. In this section, we study a class of attackers where their exploration is diminishing. In particular, the attacker repeatedly adjusts his/her actions in such a way that, at each time instant t, with probability $1-\epsilon_a(t)$, the attacker sticks to the previous action and, with probability $\epsilon_a(t)$, he/she switches to a new one via following the algorithm ALG_a. Figure 5 presents an example of the attacker's decision-making where the transition edges excluding self-loops are determined by ALG_a. Here, we do not specify the algorithm ALG_a the attacker follows and the algorithm is unknown to the defender. The exploration of the attacker is non-persistent and the exploration rate $\epsilon_a(t)$ is diminishing. In reality, non-persistent attackers could be the case when they can gather more information about the vulnerabilities of the IT system and thus their attacking strategies become more stable over the time.

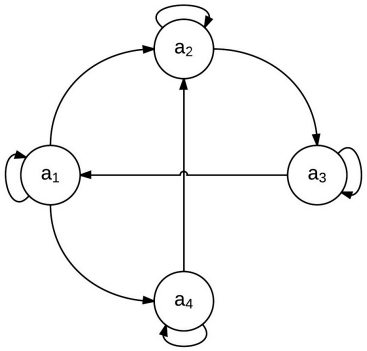

Figure 5: An illustration of the decision-making of the attacker where the transition probability of self-loops is $1-\epsilon_d(t)$.

4.1 Algorithm

In order to deal with non-persistent attackers, we propose the adaptive reinforcement learning algorithm. More specifically, at each time instant t, the defender measures the utility $u(t)$. Note that $u(t)=U(d(t),a(t))$ and but the defender is unaware of the structure of U and the attacker action $a(t)$. Then the defender updates his/her action on the basis of the values $u(t-1)$ and $u(t)$. With probability $1-\epsilon_d(t)$, the defender chooses the new action $d(t+1)$ as the one which generates a smaller values between $u(t-1)$ and $u(t)$ (Lines 8 and 9). This represents the exploitation of the defender where he/she reinforces the recent successful action. With probability $\epsilon_d(t)$, the defender explores the set $\mathcal{D}\setminus\{d(t),d(t-1)\}$, and chooses the new action $d(t+1)$ uniformly from the set $\mathcal{D}\setminus\{d(t),d(t-1)\}$ (Line 11). The adaptive reinforcement learning algorithm is formally stated in Algorithm 1 and Figure 6 describes the interactions between the defender and attacker. In Algorithm 1, sample(M) uniformly samples the set M. Here, ALG_a is a mapping from $\mathcal{D}\times\mathcal{A}$ to the simplex $\Delta^{|\mathcal{A}|}$ on $\mathcal{A}$. Here, we do not specify ALG_a the attacker uses to adjust the actions. It worthy to mention that, although the domain of ALG_a is $\mathcal{D}\times\mathcal{A}$, the attacker is unnecessarily aware of the defense $d(t)$. For example, in the Heartbleed example of Section 3.3, the attacker can observe how much private information he/she gets but is unaware which pages are guarded.

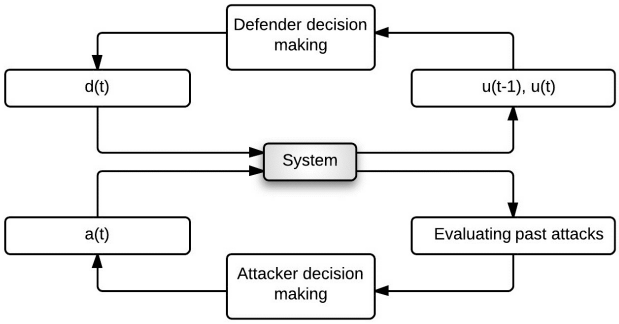

Figure 6: The interactions of the defender and attacker in the adaptive reinforcement learning algorithm.

Algorithm 1: Adaptive reinforcement learning algorithm

```
1  d(0) ← sample(D);
2  a(0) ← sample(A);
3  u(0) ← U(d(0), a(0));
4  d(1) ← d(0);
5  u(1) ← u(0);
6  while t ≥ 2 do
7      d^tp ← sample(D \ {d(t), d(t−1)}) with prob. ε_d(t);
8      if u(t) < u(t−1) then
9          d^tp ← d(t) with prob. (1 − ε_d(t));
10     else
11         d^tp ← d(t−1) with prob. (1 − ε_d(t));
12     d(t+1) ← d^tp;
13     a^tp ← ALG_a([d(t)  a(t)]^T) with prob. ε_a(t);
14     a^tp ← a(t) with prob. 1 − ε_a(t);
15     a(t+1) ← a^tp;
16     u(t+1) ← U(d(t+1), a(t+1));
```

4.2 Analysis

In this section, we provide a set of analysis on the adaptive reinforcement learning algorithm. Before doing that, let us introduce a set of notations. Denote $S\triangleq\mathcal{D}\times\mathcal{A}$, $S_{BR}\triangleq\{s=(d,a)\in\mathcal{S}\mid U(d,a)=\min_{d'\in\mathcal{D}}U(d',a)\}$, $s(t)\triangleq[d(t)\ \ a(t)]^T$, $z(t)\triangleq(s(t-1),s(t))$, $\epsilon\triangleq(\epsilon_d,\epsilon_a)$.

Given any ϵ, and consider the Markov chain $\{\mathcal{P}^\epsilon\}_{t\geq 0}$. Given any two distinct states z and z', consider all paths which start from z and end at z'. Denote by $p_{zz'}$ the large probability among all such paths. Now define a complete directed graph $\mathcal{G}$ where there is one vertex z for each state z, and the probability on the edge (z,z') is $p_{zz'}$. A *z-tree* on $\mathcal{G}$ is a spanning tree such that from every vertex $z'\neq z$, there is a unique path from z' to z. Denote by $G(z)$ the set of all *z-trees* on $\mathcal{G}$. The total probability of a *z-tree* is the product of the probabilities of its edges. The *stochastic potential* of the state z is the large total probability among

all *z-trees* in $G(z)$. Let $\Lambda(\epsilon)$ be the set of states which have maximum stochastic potential. Any of such trees is denoted by $T_{\max}(\epsilon)$.

Remark 4.1. *The above notions are inspired by the theory of resistance trees [22]. However, the above notions are defined for any $\epsilon \in (0,1)$ in contrast to $\epsilon \to 0$ in the theory of resistance trees. This allows us to characterize the transient performance of our algorithms.* •

Assumption 4.1. *The exploration rates satisfy:*

(A1) $\lim_{t\to+\infty} \epsilon_d(t) = \epsilon_d^* = 0$ *and* $\lim_{t\to+\infty} \epsilon_a(t) = \epsilon_a^* = 0$*;*

(A2) Either $\lim_{t\to+\infty} \frac{\epsilon_d(t)}{\epsilon_a(t)}$ *or* $\lim_{t\to+\infty} \frac{\epsilon_a(t)}{\epsilon_d(t)}$ *exists;*

(A3) The sequences $\{\epsilon_d(t)\}$ *and* $\{\epsilon_a(t)\}$ *are not summable;*

(A4) There exists $T \geq 0$ *such that* $\epsilon_a(t) < \epsilon_d(t)(1-\epsilon_d(t))(1-\epsilon_a(t))$ *for all* $t \geq T$.

Assumption 4.2. *The Markov chain induced by* ALG_a *is irreducible and aperiodic.*

The following lemma characterizes the convergence of the sequence of limiting distributions $\mu^*(\epsilon(t))$.

Lemma 4.1. *If (A1) and (A2) in Assumption 4.1 hold, then the limiting distribution* $\mu^* \triangleq \lim_{t\to+\infty} \mu^*(\epsilon(t))$ *exists and its support is contained in* $\Lambda(\epsilon^*)$*. Furthermore, the convergence rate can be characterized as follows:*

$$\|\mu^*(\epsilon(t)) - \mu^*\| \leq O(\|\epsilon(t)\|).$$

Proof: We define the numbers

$$\sigma_z(\epsilon(t)) \triangleq \sum_{T\in G(z)} P_T^{\epsilon(t)}, \quad P_T^{\epsilon(t)} \triangleq \prod_{(x,y)\in T} P_{xy}^{\epsilon(t)}$$

$$\mu_z^*(\epsilon(t)) \triangleq \frac{\sigma_z(\epsilon(t))}{\sum_{x\in X}\sigma_x(\epsilon(t))}, \quad \sigma(\epsilon(t)) \triangleq \sum_{x\in X}\sigma_x(\epsilon(t)).$$

Note that $\{\mathcal{P}^{\epsilon(t)}\}_{t\geq 0}$ is irreducible and thus $\sigma_z(\epsilon(t)) > 0$. As Lemma 3.1 of Chapter 6 in [9], one can show that $\mu^*(\epsilon(t))$ is the unique stationary distribution of $\{\mathcal{P}^{\epsilon(t)}\}_{t\geq 0}$; i.e., $(\mu^*(\epsilon(t)))^T P^{\epsilon(t)} = (\mu^*(\epsilon(t)))^T$.

The probability $P_{xy}^{\epsilon(t)}$ is a linear combination of $\epsilon_d(t)$, $\epsilon_a(t)$, $\epsilon_d(t)\epsilon_a(t)$, $1-\epsilon_d(t)$, $1-\epsilon_a(t)$ and $(1-\epsilon_d(t))(1-\epsilon_a(t))$. Hence, $\sigma_z(\epsilon(t))$ is a polynomial of $\epsilon_d(t)$ and $\epsilon_a(t)$. So is $\sigma(\epsilon(t))$.

Case 1: $\lim_{t\to+\infty} \frac{\epsilon_d(t)}{\epsilon_a(t)} = \gamma \geq 0$.

Let $\epsilon_d(t) = \gamma(t)\epsilon_a(t)$. and replace $\epsilon_d(t)$ by $\gamma(t)\epsilon_a(t)$ in σ_z and σ. We rewrite the polynomials as

$$\sigma_z(\epsilon(t)) = c_l^z(t)\epsilon_a(t)^l + c_{l+1}^z(t)\epsilon_a(t)^{l+1} + \cdots + c_m^z(t)\epsilon_a(t)^m,$$

$$\sigma(\epsilon(t)) = c_l(t)\epsilon_a(t)^l + c_{l+1}(t)\epsilon_a(t)^{l+1} + \cdots + c_m(t)\epsilon_a(t)^m,$$

where $0 \leq l$. Since $\gamma(t)$ converges, so do the coefficients in $\sigma_z(\epsilon(t))$ and $\sigma(\epsilon(t))$. Without loss of any generality, we assume that the limit of $c_l(t)$ is non-zero. If the limit of $c_l^z(t)$ is non-zero, then $z \in \Lambda(\epsilon^*)$ and

$$\lim_{t\to+\infty} \mu_z^*(\epsilon(t)) = \frac{\lim_{t\to+\infty} c_l^z(t)}{\lim_{t\to+\infty} c_l(t)} = \mu_z^*.$$

If the limit of $c_l^z(t)$ is zero, $z \notin \Lambda(\epsilon^*)$ and thus we have $\lim_{t\to+\infty} \mu_z^*(\epsilon(t)) = 0$.

We now proceed to further investigate the convergence rate to μ^*. For each $z \in \Lambda(\epsilon^*)$, we have

$$\begin{aligned}
&|\mu_z^*(\epsilon(t)) - \mu_z^*| \\
&= \Big|\frac{c_l^z(t)\epsilon_a(t)^l + c_{l+1}^z(t)\epsilon_a(t)^{l+1} + \cdots + c_m^z(t)\epsilon_a(t)^m}{c_l(t)\epsilon_a(t)^l + c_{l+1}(t)\epsilon_a(t)^{l+1} + \cdots + c_m(t)\epsilon_a(t)^m} \\
&- \frac{\lim_{t\to+\infty} c_l^z(t)}{\lim_{t\to+\infty} c_l(t)}\Big| \\
&= \epsilon_a(t)\Big|\frac{L_z(1,\epsilon_a(t),\cdots,\epsilon_a(t)^{m-l}) + \beta_z(t)}{L_z'(1,\epsilon_a(t),\cdots,\epsilon_a(t)^{m-l})}\Big|,
\end{aligned}$$

where L_z and L_z' are linear functions, the constant term of L_z' is non-zero and the sequence of $\beta_z(t)$ diminishes. Notice that the sequence $L_z(1,\epsilon_a(t),\cdots,\epsilon_a(t)^{m-l})$, $L_z'(1,\epsilon_a(t),\cdots,\epsilon_a(t)^{m-l})$ and $\beta_z(t)$ converge as $t \to +\infty$. So the sequence $\big|\frac{L_z(1,\epsilon_a(t),\cdots,\epsilon_a(t)^{m-l})+\beta_z(t)}{L_z'(1,\epsilon_a(t),\cdots,\epsilon_a(t)^{m-l})}\big|$ is uniformly bounded. Hence, it establishes the following

$$|\mu_z^*(\epsilon(t)) - \mu_z^*| \leq O(\|\epsilon_a(t)\|) \leq O(\|\epsilon(t)\|).$$

Case 2: $\lim_{t\to+\infty} \frac{\epsilon_a(t)}{\epsilon_d(t)} = \gamma \geq 0$.

The proof is analogous to Case 1. •

Remark 4.2. *Lemma 4.1 resembles to stochastic potential in [22].* •

The following lemma characterizes the set of $\Lambda(\epsilon)$ when ϵ is small enough.

Lemma 4.2. *Suppose (A1) in Assumption 4.1 hold and* $\epsilon_a < \epsilon_d(1-\epsilon_d)(1-\epsilon_a)$*. Then the roots of* $T_{\max}(\epsilon)$ *belong to* $\mathrm{diag}(S_{BR}\times S_{BR})$*. In addition,* $\Lambda(\epsilon^*) \in \mathrm{diag}(S_{BR}\times S_{BR})$.

Proof: The proof is divided into seven claims.

Claim 1: For any pair of $(s,s) \in \mathrm{diag}(\mathcal{S}\times\mathcal{S})\setminus\mathrm{diag}(S_{BR}\times S_{BR})$ and $(s^*,s^*) \in \mathrm{diag}(S_{BR}\times S_{BR})$, there is a finite sequence of transitions from (s,s) to (s^*,s^*):

$$\mathcal{L} \triangleq (s,s) \overset{\epsilon_a}{\to} (s,s') \overset{\epsilon_d}{\to} (s',s^*) \overset{(1-\epsilon_d)(1-\epsilon_a)}{\to} (s^*,s^*).$$

Proof: Let $s = (d,a)$, $s' = (d,a^*)$ and $s^* = (d^*,a^*)$. The transition from s to s' needs the exploration of the attacker. The transition from s' to s^* needs the exploration of the defender. With this, one can easily verify the path. •

Claim 2: For any $(s^0,s^1) \in \mathcal{S}\times\mathcal{S}\setminus\mathrm{diag}(\mathcal{S}\times\mathcal{S})$, there is a finite path such that

$$(s^0,s^1) \overset{(1-\epsilon_d)(1-\epsilon_a)}{\to} (s^1,s^1).$$

Proof: The path is the case when none of the defender and attacker performs the exploration. •

Claim 3: Pick any $s^* = (d^*,a^*) \in S_{BR}$ and consider $z \triangleq (s^*,s^*)$, $z' \triangleq (s',s^*)$ where $s' = (d^*,a') \notin S_{BR}$ with $a' \neq a^*$. There is a transition such that

$$z^* \overset{\epsilon_a}{\to} z'.$$

Proof: The path is the case when the defender does not explore but the attacker does. •

Claim 4: $\Lambda(\epsilon) \subseteq \mathrm{diag}(\mathcal{S}\times\mathcal{S})$.

Proof: Assume that there is $(s^0,s^1) \in \Lambda(\epsilon)$ but $s^1 \neq s^0$. Since $(s^0,s^1) \in \Lambda(\epsilon)$, there is a tree, say $T_{\max}$, rooted at

(s^0, s^1) such that the tree has the maximum probability. Construct a new tree T' by adding the feasible edge from (s^1, s^0) to (s^1, s^1) and deleting the edge leaving (s^1, s^1). Note that the probability of the added edge is $(1-\epsilon_d)(1-\epsilon_a)$ by Claim 2 and that of the deleted edge is at most $\max\{\epsilon_d, \epsilon_a\}$ where at least one player explores. Since $(1-\epsilon_d)(1-\epsilon_a) > \max\{\epsilon_d, \epsilon_a\}$, so the total probability of T' is strictly larger that that of $T_{\max}$. We reach a contradiction. •

Claim 5: Pick any $s^* \in S_{BR}$ and consider $z \triangleq (s^*, s^*)$, $z' \triangleq (s', s^*)$ where $s' \notin S_{BR}$. Pick any maximum stochastic potential tree $T_{\max}(\epsilon)$. If $(z, z') \in T_{\max}(\epsilon)$, then the probability of the edge (z, z') is either $\epsilon_d\epsilon_a$ or ϵ_a.

Proof: Suppose the deviator in the transition $z \to z'$ is unique, say the defender. So $z' = (s^*, s')$ where $s' = (d', a^*)$ and $d' \neq d^*$. Notice that the corresponding transition probability is ϵ_d. Since $s^* \in S_{BR}$, we have that $U(d, a^*) \geq U(d^*, a^*)$ for any $d \in \mathcal{D}$.

Since $z' \notin \text{diag}(\mathcal{S} \times \mathcal{S})$, it follows from Claim 4 that z' can not be the root of $T_{\max}(\epsilon)$ and thus has a successor $z'' = (s', s'')$ with $s'' = (d'', a^*)$ on $T_{\max}(\epsilon)$. By Claim 2, the probability of the transition $z' \to z''$ is $(1-\epsilon_d)(1-\epsilon_a)$. Since $U(d', a^*) \geq U(d^*, a^*)$, the defender must perform the exploitation; i.e., $d'' = d^*$ and thus $z'' = (s', s^*)$. By one more exploitation, the state z'' must have a successor z''' and $z''' = z = (s^*, s^*)$. We then obtain a loop in $T_{\max}(\epsilon)$ which contradicts that $T_{\max}(\epsilon)$ is a tree.

It must be one of the cases: (1) both of the defender and attacker explore in the transition $z \to z'$; (2) the attacker is the only deviator. Thus the probability of the edge (z, z') is $\epsilon_d\epsilon_a$ or ϵ_a. •

Claim 6: The roots of $T_{\max}(\epsilon)$ belongs to $\text{diag}(S_{BR} \times S_{BR})$.

Proof: Assume there is (s, s) where $s \neq S_{BR}$. Let $T_{\max}(\epsilon)$ be a maximum stochastic potential tree at (s, s). Let $s = (d, a)$ and choose (s^*, s^*) with $s^* = (d^*, a) \in S_{BR}$. So there is a finite sequence of transitions from (s, s) to (s^*, s^*):

$$\mathcal{L} \triangleq (s, s) \overset{\epsilon_d}{\to} (s, s^*) \overset{(1-\epsilon_d)(1-\epsilon_a)}{\to} (s^*, s^*),$$

where in the first transition the defender is only deviator and in the second transition, none of the players deviates. Construct a new tree T' by adding the path $\mathcal{L}$ from (s, s) to (s^*, s^*) and deleting the edges leaving those four states. The total probability of the added edges is $\epsilon_d(1-\epsilon_d)(1-\epsilon_a)$ by Claim 1 and that of the deleted edge is at most ϵ_a. Recall that $\epsilon_a < \epsilon_d(1-\epsilon_d)(1-\epsilon_a)$. So the total probability of T' is strictly larger that that of $T_{\max}(\epsilon)$. We reach a contradiction. •

Claim 7: $\Lambda(\epsilon^*) \subseteq \text{diag}(S_{BR} \times S_{BR})$.

Proof: Since Claim 6 hold for any sufficiently small $\epsilon > 0$, we then reach the desired result. •

The following theorem characterizes that the adaptive reinforcement learning algorithm asymptotically converges to the best response set S_{BR} with probability one.

Theorem 4.1. *Let (A1) to (A4) in Assumption 4.1 and Assumption 4.2 hold. Consider the Markov chain $\{\mathcal{P}_t\}$ induced by the adaptive reinforcement learning algorithm. Then it holds that*

$$\lim_{t \to +\infty} \mathbb{P}((s(t), s(t+1)) \in \text{diag}(S_{BR} \times S_{BR})) = 1.$$

Proof: Notice that

$$\begin{aligned}&\|\mu(t) - \mu^*(\epsilon(t))\| \\ &\leq \|\mu(t) - \mu^*(\epsilon(t-1))\| + \|\mu^*(\epsilon(t)) - \mu^*(\epsilon(t-1))\|. \quad (1)\end{aligned}$$

Let us consider the term $\|\mu(t) - \mu^*(\epsilon(t-1))\|$ in (1) as follows.

$$\begin{aligned}&\|\mu(t) - \mu^*(\epsilon(t-1))\| \\ &\leq \|(P^{\epsilon(t-1)})^T\mu(t-1) - (P^{\epsilon(t-1)})^T\mu^*(\epsilon(t-1))\| \\ &\leq \|P^{\epsilon(t-1)}\|\|\mu(t-1) - \mu^*(\epsilon(t-1))\| \\ &\leq \beta(t-1)\|\mu(t-1) - \mu^*(\epsilon(t-1))\|, \quad (2)\end{aligned}$$

where $\beta(t-1) \triangleq \max\{1-\epsilon_d(t-1), 1-\epsilon_a(t-1)\}$.

Let $x(t) \triangleq \|\mu(t) - \mu^*(\epsilon(t))\|$ and $y(t-1) \triangleq \|\mu^*(\epsilon(t)) - \mu^*(\epsilon(t-1))\|$. Then it follows from (2) that

$$x(t) \leq \beta(t-1)x(t-1) + y(t-1). \quad (3)$$

Similar to Claim 6 on Page 13 of [25], we have $\{y(t)\}$ is summable. Since $\beta(t) \in [0, 1]$, it follows from (3) that

$$\begin{aligned}x(t) &\leq \prod_{s=k}^{t-1} \beta(s)x(k) + \sum_{s=k}^{t-1} \Theta_{\ell=s}^{t-2} y(s) \\ &\leq \prod_{s=k}^{t-1} \beta(s)x(k) + \sum_{s=k}^{t-1} y(s), \quad (4)\end{aligned}$$

where $\Theta_{\ell=s}^{t} \triangleq \prod_{\ell=s}^{t} \beta(\ell+1)$ and $\prod_{\ell=t-1}^{t-2} \beta(\ell+1) = 1$. Notice that $\log(1-x) < -x$ for all $x \in (0, 1)$. Then we have

$$\begin{aligned}\Theta_{\ell=s}^{t} &= \prod_{\ell=s}^{t} \beta(\ell+1) \\ &= \prod_{\ell=s}^{t} \max\{1-\epsilon_d(\ell+1), 1-\epsilon_a(\ell+1)\} \\ &\leq e^{-\sum_{\ell=s}^{t} \min\{\epsilon_d(\ell+1), \epsilon_a(\ell+1)\}}. \quad (5)\end{aligned}$$

By (5), (A3) and the summability of $\{y(t)\}$, we take the limits on t and then k on both sides of (4), we reach the convergence of $x(t)$ to zero. Hence, we have $\mu(t) - \mu^*(\epsilon(t-1)) \to 0$. Since $\mu^*(\epsilon(t)) \to \mu^*$ and $\Lambda(\epsilon^*) \subseteq \text{diag}(S_{BR} \times S_{BR})$, we reach the desired result. •

5. ROBUST REINFORCEMENT LEARNING ALGORITHM

In Section 4, we study non-persistent attackers whose exploration rate is diminishing. In this section, we study a complementary class of persistent attackers. The behavior of persistent attackers is similar to that of non-persistent attackers. The distinction is that the exploration rate $\epsilon(t)$ may not be diminishing. The persistent exploration of the attacker may require the defender to constantly change defenses, challenging the defense deployment in practice. In this section, we propose the robust reinforcement learning algorithm which is insensitive to the persistent variations of the attacks.

5.1 Algorithm

The robust reinforcement learning algorithm is informally stated as follows. At each time instant t, the defender measures the system performance $u(t)$. Then the defender compares the values of $u(0), \cdots, u(t)$ and determines the set,

say $D_{MM}(t)$, where the actions in $D_{MM}(t)$ can minimize the maximum utilities in the history. With probability $1-\epsilon_d(t)$, the defender chooses any defense in $D_{MM}(t)$ to be the new action $d(t+1)$. This represents the exploitation of the defender where he/she reinforces the successful action in the whole history. With probability $\epsilon_d(t)$, the defender explores the set $\mathcal{D} \setminus \{D_{MM}(t)\}$, and chooses the new action $d(t+1)$ uniformly from the set.

Denote by $M(d,t) \triangleq \max_{0\le s\le t, d(s)=d} u(t)$ the maximum utility generated by action d up to time t. And denote by $D_{MM}(t) \triangleq \{d \mid M(d,t) \le M(d',t), \quad \forall d' \in \mathcal{D}\}$ the set of actions whose $M(d,t)$ is smallest among all the actions. The robust reinforcement learning algorithm is formally stated in Algorithm 2.

Algorithm 2: Robust reinforcement learning algorithm

```
d(0) ← sample(D);
a(0) ← sample(A);
u(0) ← U(d(0), a(0));
d(1) ← d(0);
u(1) ← u(0);
while t ≥ 2 do
    d^tp ← sample(D \ D_MM(t)) with prob. ε_d(t);
    d^tp ← sample(D_MM(t)) with prob. (1 − ε_d(t));
    d(t+1) ← d^tp;
    a^tp ← ALG_a([d(t)  a(t)]^T) with prob. ε_a(t);
    a^tp ← a(t−1) with prob. 1 − ε_a(t);
    a(t+1) ← a^tp;
    u(t+1) ← U(d(t+1), a(t+1));
```

5.2 Analysis

Denote by $M(d) \triangleq \max_{a\in\mathcal{A}} u(d,a)$ the maximum utility generated by action d. And denote by $D_{MM} \triangleq \{d \mid M(d) \le M(d'), \quad \forall d' \in \mathcal{D}\}$. So the set D_{MM} the collections of the defenses which minimize the maximal utilities. The following theorem characterizes that the robust reinforcement learning algorithm converges to the min-max set D_{MM} with probability at least $1-\epsilon_d^*$.

Theorem 5.1. *Assume that $\epsilon_a(t) \ge \epsilon_a^* > 0$ and $\epsilon_d(t) \ge \epsilon_d^* > 0$. Consider the Markov chain $\{\mathcal{P}_t\}$ induced by the robust reinforcement learning algorithm. Then it holds that*

$$\lim_{t\to+\infty} \mathbb{P}(d(t) \in D_{MM}) \ge 1-\epsilon_d^*.$$

Proof: We define the hitting time $T(s(0))$ as the first time the Markov chain $\{\mathcal{P}_t\}$ such that $\Theta(s(t)) = \mathcal{S}$ when $\{\mathcal{P}_t\}$ starts from the initial state $s(0)$. Notice that the hitting time is a random variable. Since $\mathcal{S}$ and $\{\mathcal{P}_t\}$ is irreducible and aperiodic, $\mathbb{E}[T(s(0))]$ is bounded by, say $\tilde{T}$. It follows from the Markov inequality that

$$\mathbb{P}(T(s(0)) > 2\tilde{T} | \Theta(s(t)) \ne \mathcal{S}) \le \frac{\mathbb{E}[T(s(0))]}{\tilde{T}} = \frac{1}{2}.$$

That is, the probability that $\Theta(s(t)) \ne \mathcal{S}$ after $2\tilde{T}$ time instants is smaller than $\frac{1}{2}$. Starting from $2\tilde{T}$, let us consider the posterior evolution of $\{\mathcal{P}_t\}$ for the next $2\tilde{T}$ time instants. We have

$$\mathbb{P}(T(s(2\tilde{T})) > 2\tilde{T} | \Theta(s(2\tilde{T})) \ne \mathcal{S}) \le \frac{\mathbb{E}[T(s(0))]}{\tilde{T}} = \frac{1}{2}.$$

That is, the probability that $\Theta(s(t)) \ne \mathcal{S}$ after $4\tilde{T}$ time instants is smaller than $(\frac{1}{2})^2$. One can follow the induction to show that the probability that $\Theta(s(t)) \ne \mathcal{S}$ after $2n\tilde{T}$ time instants is smaller than $(\frac{1}{2})^n$. We have

$$\lim_{t\to+\infty} \mathbb{P}(\Theta(s(t)) = \mathcal{S}) = 1,$$

and thus

$$\lim_{t\to+\infty} \mathbb{P}(D_{MM}(t) = D_{MM}) = 1. \tag{6}$$

With prob. $\epsilon_d(t)$, it holds that

$$d(t+1) \in D_{MM}(t). \tag{7}$$

The combination of (6) and (7) renders the desired result.•

6. SIMULATIONS

In what follows, we will use numerical simulations to demonstrate the performance of two proposed algorithms.

6.1 Adaptive reinforcement learning algorithm

The defender has 10 defenses and the attacker has 10 attacks. We choose the exploration rates $\epsilon_a(t) = \frac{1}{t^c}$, $\epsilon_d(t) = \frac{3}{t^c}$ with $c = 0.5$ and $\text{ALG}_a([d(t)\ \ a(t)]^T) = \text{sample}(\mathcal{A} \setminus \{a(t)\})$. We generate five 10×10 utility tables for the defender where the entries $U(d,a)$ of each table is randomly chosen from $[0,1]$. For each table, we repeat the simulations 25 times. Figure 7 shows the distribution of convergence times of 125 simulations. For 125 simulations, the mean of the convergence time is 78.55 time instants and the standard deviation is 17.99 time instants. In practice, the Heartbleed has to try a large number of attacks, ranging from 1 million to 2 millions. The simulation results suggests that the defender reconfigures the defense every $\frac{1{,}000{,}000}{78.55} \approx 12730$ attacks on average.

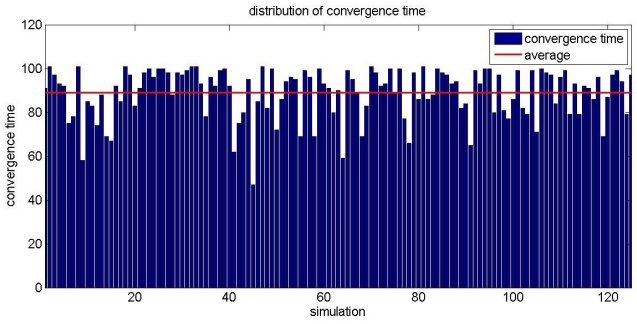

Figure 7: The distribution of convergence times of the adaptive reinforcement learning algorithm.

6.2 Robust reinforcement learning algorithm

Choose $\epsilon_a(t) = \frac{1}{2}$, $\epsilon_d(t) = \max\{\frac{1}{t^c}, \frac{1}{100}\}$ with $c = 0.5$ and $\text{ALG}_a([d(t)\ \ a(t)]^T) = \text{sample}(\mathcal{A} \setminus \{a(t)\})$. We use the same utility tables as last section. For each table, we repeat the simulations five times. Figure 8 shows the distribution of convergence times of 125 simulations. The mean of the convergence time is 77.96 time instants and the standard deviation is 18.54 time instants.

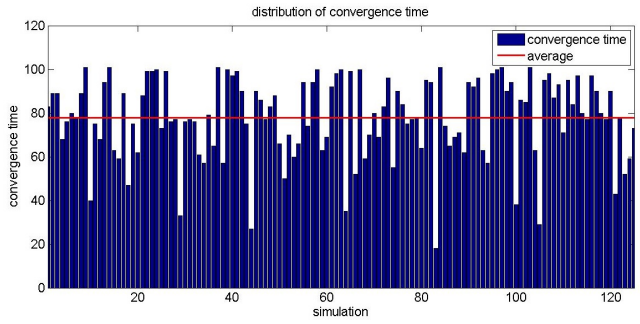

Figure 8: The distribution of convergence times of the robust reinforcement learning algorithm.

7. DISCUSSION AND CONCLUSIONS

We have proposed two iterative reinforcement learning algorithms for adaptive cyber defense. Their asymptotic convergence has been formally analyzed and their performance has been verified via numerical simulations. There is a key difference between two proposed algorithms. In the adaptive reinforcement learning algorithm, the defender myopically responds to the most recent history. Hence, the defenses asymptotically converge to the best response of the attacks. Instead, in the robust reinforcement learning algorithm, the defender reacts to the whole history. So the defenses asymptotically converge to the min-max strategy over the defense-attack domain.

8. REFERENCES

[1] `http://heartbleed.com/`.

[2] T. Alpcan and T. Basar. *Network Security: A Decision and Game Theoretic Approach*. Cambridge University Press, 2011.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[4] D.P. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[5] R. Boome. Security metrics and security investment models. In *Proceedings of the 5th International Conference on Advances in Information and Computer Security*, pages 10–24, 2010.

[6] R. Boome and T. Moore. The iterated weakest link - A model of adaptive security investment. In *Workshop on Economics of Information Security*, pages 2406–2411, 2009.

[7] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[8] L. Demetz and D. Bachlechner. To invest or not to invest? Assessing the economic viability of a policy and security configuration management tool. *The Economics of Information Security and Privacy*, pages 25–47, 2013.

[9] M. Freidlin and A. Wentzell. *Random perturbations of dynamical systems*. New York: Springer Verlag, 1984.

[10] S. Jajodia, A. Ghosh, V. Swarup, C. Wang, and X. Wang. *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*. Springer, 2011.

[11] S. Jajodia, A. Ghosh, V. Swarup, C. Wang, and X. Wang. *Moving Target Defense II: Application of Game Theory and Adversarial Modeling*. Springer, 2013.

[12] J. Lin, P. Liu, and J. Jing. Using signaling games to model the multi-step attack-defense scenarios on confidentiality. In *GameSec*, pages 118–137, 2012.

[13] P. Liu, W. Zang, and M. Yu. Incentive-based modeling and inference of attacker intent, objectives, and strategies. *ACM Transactions on Information and System Security*, 8(1):1094–9224, 2005.

[14] Y. Luo, F. Szidarovszky, Y. Al-Nashif, and S. Hariri. Game theory based network security. *Journal of Information Security*, 1(1):41–44, 2010.

[15] K. Lye and J. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1):71–86, 2005.

[16] M.H. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J.P. Hubaux. Game theory meets network security and privacy. *ACM Computing Surveys*, 45(3):25–39, 2013.

[17] J.R. Marden, H.P. Young, G. Arslan, and J.S. Shamma. Payoff based dynamics for multi-player weakly acyclic games. 48(1):373–396, February 2009.

[18] H. Okhravi, M. Rabe, T. Mayberry, W. Leonard, T. Hobson, D. Bigelow, and W. Streilein. Survey of cyber moving target techniques. Technical report, Lincoln Lab, Massachusetts Institute of Technology, 2013.

[19] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. pages 1–10, Hawaii, USA, 2010.

[20] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[21] G. Theodorakopoulos and J. Baras. Game theoretic modeling of malicious users in collaborative networks. *IEEE Journal on Selected Areas in Communications*, 26(7):1317–1327, 2008.

[22] H.P. Young. The evolution of conventions. *Econometrica*, 61:57–84, January 1993.

[23] M. Zhu and S. Martínez. Distributed coverage games for mobile visual sensors (I): Reaching the set of Nash equilibria. In *IEEE International Conference on Decision and Control*, pages 169–174, Shanghai, China, Dec 2009.

[24] M. Zhu and S. Martínez. Distributed coverage games for mobile visual sensors (II): Reaching the set of global optima. In *IEEE International Conference on Decision and Control*, pages 175–180, Shanghai, China, Dec 2009.

[25] M. Zhu and S. Martínez. Distributed coverage games for energy-aware mobile sensor networks. *SIAM Journal on Control and Optimization*, 51(1):1–27, 2013.

[26] Q. Zhu, H. Tembine, and T. Basar. Hybrid learning in stochastic games and its applications in network security. *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, pages 305–329, 2013.