Reinforcement Learning Algorithms for Adaptive Cyber Defense against Heartbleed

NSE ML+Security Reading Group

Kim Hammar

kimham@kth.se

Division of Network and Systems Engineering
KTH Royal Institute of Technology

October 22, 2021

The Context and Key Points of the Paper

- The paper proposes **two reinforcement learning algorithms** for *Adaptive Cyber Defense*
- Motivating use case: the Heartbleed vulnerability

Outline

- **Background**
  - Heartbleed
- **The Paper**
  - Approach & Contributions
  - System Model
  - Proposed Algorithms
  - Theoretical Analysis
- **Limitations of the paper and Discussion**
  - Limitations of the paper
  - Discussion about future work
- **Conclusions**

Background: Heartbleed

- **A security bug in the OpenSSL library**
  - Released 2012
  - Disclosed 2014
- **Affected software:** most implementations of TLS
- **How it works:**
  - A sender in OpenSSL can send a heartbeat msg with payload+length
  - The receiver allocates a memory buffer according to the length without verifying the length
  - The receiver writes the payload to the buffer
  - The receiver sends back the content of the buffer to the sender
  - Since the buffer size can be larger than the payload (it is not verified) the sender may send back more data than the original payload - possibly sensitive data.

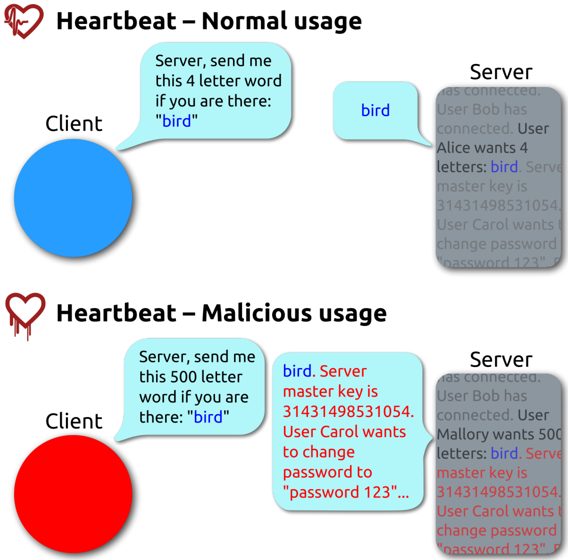
Background: Heartbleed

- **A security bug in the OpenSSL library**
  - Released 2012
  - Disclosed 2014
- **Affected software:** most implementations of TLS
- **How it works:**
  - A sender in OpenSSL can send a heartbeat msg with payload+length
  - The receiver allocates a memory buffer according to the length without verifying the length
  - The receiver writes the payload to the buffer
  - The receiver sends back the content of the buffer to the sender
  - Since the buffer size can be larger than the payload (it is not verified) the sender may send back more data than the original payload - possibly sensitive data.

Background: Heartbleed (CVE-2014-0160)

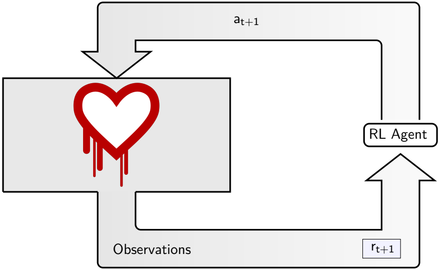
- **A security bug in the OpenSSL library**
  - Released 2012
  - Disclosed 2014
- **Affected software:** most implementations of TLS
- **How it works:**
  - A sender in OpenSSL can send a heartbeat msg with payload+length
  - The receiver allocates a memory buffer according to the length without verifying the length
  - The receiver writes the payload to the buffer
  - The receiver sends back the content of the buffer to the sender
  - Since the buffer size can be larger than the payload (it is not verified) the sender may send back more data than the original payload - possibly sensitive data.

Background: Heartbleed (CVE-2014-0160)



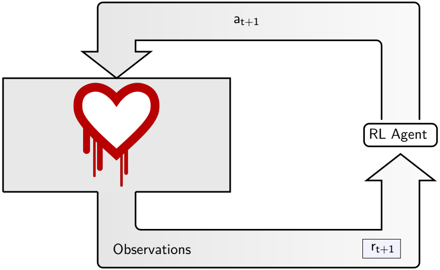
The Paper Approach and Contributions

- **Approach**:
  - Adaptive Cyber Defense (ACD)
  - Model ACD as a decision problem
  - Find defender strategies through reinforcement learning
- **Contributions**:
  - A generic system model of security problems (minor contribution)
  - Two custom reinforcement learning algorithms
    - One algorithm that only works against stable attackers
    - One "robust" algorithm that works against random attackers
  - Convergence proofs



The Paper Approach and Contributions

- **Approach**:
  - Adaptive Cyber Defense (ACD)
  - Model ACD as a decision problem
  - Find defender strategies through reinforcement learning
- **Contributions**:
  - A generic system model of security problems (minor contribution)
  - Two custom reinforcement learning algorithms
    - One algorithm that only works against stable attackers
    - One "robust" algorithm that works against random attackers
  - Convergence proofs



The System Model

- The defender has n defenses:
  - $\mathcal{D} \triangleq \{d_1, \ldots, d_n\}$
- The attacker has m attacks:
  - $\mathcal{A} \triangleq \{a_1, \ldots, a_m\}$
- **Utility function** U:
  - $U : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$

The System Model

- The defender has n defenses:
  - $\mathcal{D} \triangleq \{d_1, \ldots, d_n\}$
- The attacker has m attacks:
  - $\mathcal{A} \triangleq \{a_1, \ldots, a_m\}$
- **Utility function** U:
  - $U : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$

The System Model

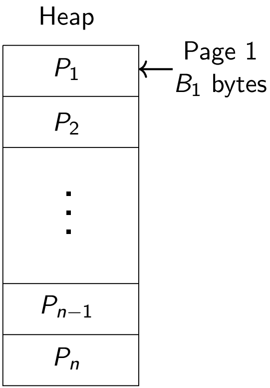
- The defender has n defenses:
  - $\mathcal{D} \triangleq \{d_1, \ldots, d_n\}$
- The attacker has m attacks:
  - $\mathcal{A} \triangleq \{a_1, \ldots, a_m\}$
- **Utility function** U:
  - $U : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$

The General System Model

- The defender has n defenses:
  - $\mathcal{D} \triangleq \{d_1, \ldots, d_n\}$
- The attacker has m attacks:
  - $\mathcal{A} \triangleq \{a_1, \ldots, a_m\}$
- **Utility function** U:
  - $U : \mathcal{D} \times \mathcal{A} \to \mathbb{R}$
- That's it!
  - No explicit states (you can consider previous actions as state)
  - No transition probabilities
  - No observation function
  - Not a sequential problem
  - Assume non-rational attacker

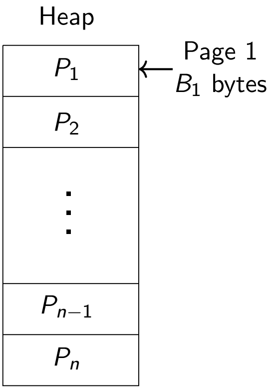
The System Model of Heartbleed

- The defender can defend pages P_i on a heap of n pages
  - $\mathcal{D} \triangleq \{P_1, \ldots, P_n\}$
  - Each page has B_i bytes of data
  - Defending a subset of pages: $d(t) = \{P_i, P_k, \ldots\} \subseteq \mathcal{D}$
    - monitor the page
    - detect unwanted read operations
- The attacker attacks by sending heartbeats:
  - $\mathcal{A} \triangleq \mathcal{D} \times \mathbb{N}$
  - $a(t) = (p(t), b(t))$
- **Utility function** U:
  - $U(a, d) = c(d) - I(a, d)$
  - $c(d)$ is the cost of defenses
  - $I(a, d)$ is the number of heartbeat requests



The System Model of Heartbleed

- The defender can defend pages P_i on a heap of n pages
  - $\mathcal{D} \triangleq \{P_1, \ldots, P_n\}$
  - Each page has B_i bytes of data
  - Defending a subset of pages: $d(t) = \{P_i, P_k, \ldots\} \subseteq \mathcal{D}$
    - monitor the page
    - detect unwanted read operations
- The attacker attacks by sending heartbeats:
  - $\mathcal{A} \triangleq \mathcal{D} \times \mathbb{N}$
  - $a(t) = (p(t), b(t))$
- **Utility function** U:
  - $U(a, d) = c(d) - I(a, d)$
  - $c(d)$ is the cost of defenses
  - $I(a, d)$ is the number of heartbeat requests

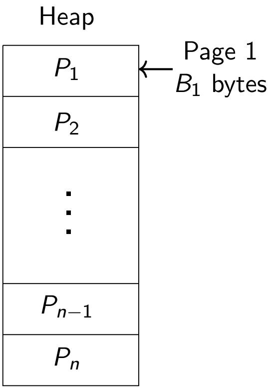


The System Model of Heartbleed

- The defender can defend pages P_i on a heap of n pages
  - $\mathcal{D} \triangleq \{P_1, \ldots, P_n\}$
  - Each page has B_i bytes of data
  - Defending a subset of pages: $d(t) = \{P_i, P_k, \ldots\} \subseteq \mathcal{D}$
    - monitor the page
    - detect unwanted read operations

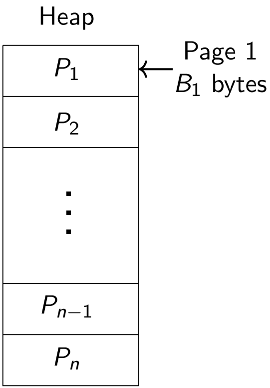
- The attacker attacks by sending heartbeats:
  - $\mathcal{A} \triangleq \mathcal{D} \times \mathbb{N}$
  - $a(t) = (p(t), b(t))$

- **Utility function** U:
  - $U(a, d) = c(d) - I(a, d)$
  - $c(d)$ is the cost of defenses
  - $I(a, d)$ is the number of heartbeat requests

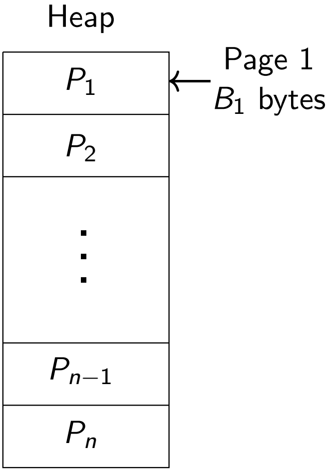


The System Model of Heartbleed

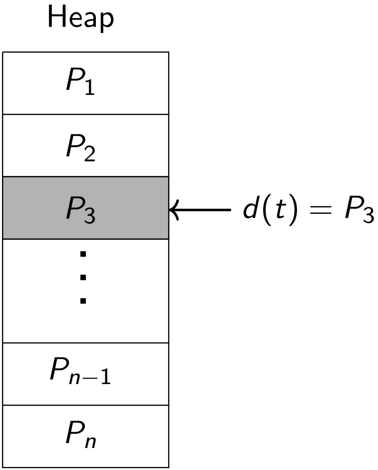
- The defender can defend pages P_i on a heap of n pages
  - $\mathcal{D} \triangleq \{P_1, \ldots, P_n\}$
  - Each page has B_i bytes of data
  - Defending a subset of pages: $d(t) = \{P_i, P_k, \ldots\} \subseteq \mathcal{D}$
    - monitor the page
    - detect unwanted read operations through segfaults
- The attacker attacks by sending heartbeats:
  - $\mathcal{A} \triangleq \mathcal{D} \times \mathbb{N}$
  - $a(t) = (p(t), b(t))$
- **Utility function** U:
  - $U(a, d) = c(d) - I(a, d)$
  - $c(d)$ is the cost of defenses
  - $I(a, d)$ is the number of heartbeat requests



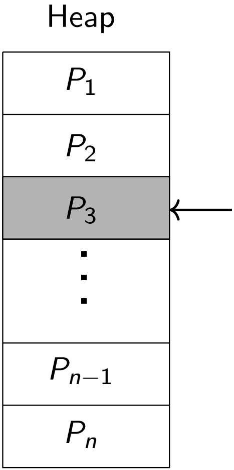
Defending the Heap from Heartbleed Attacks



Defending the Heap from Heartbleed Attacks



Defending the Heap from Heartbleed Attacks

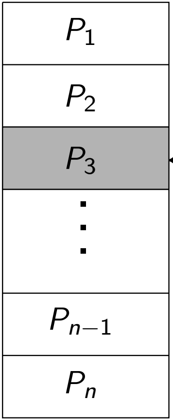


$\longleftarrow d(t) = P_3$

(+) protect P_3 from unwanted reads

Defending the Heap from Heartbleed Attacks

Heap

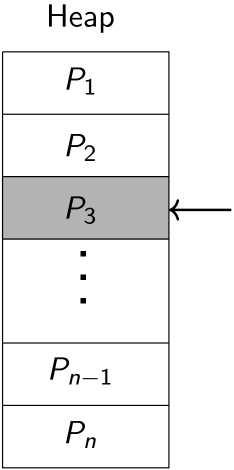


$d(t) = P_3$

(+) protect P_3 from unwanted reads

(-) increase system response time $c(d(t))$

Defending the Heap from Heartbleed Attacks



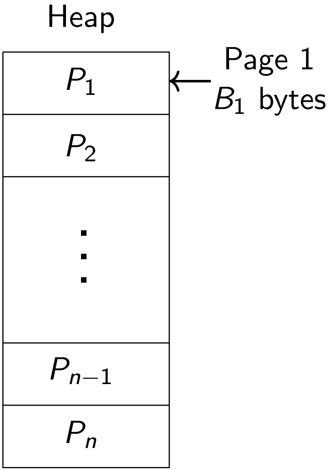
$d(t) = P_3$

(+) protect P_3 from unwanted reads

(-) increase system response time $c(d(t))$

(-) reduce system requests $I(a(t), d(t))$

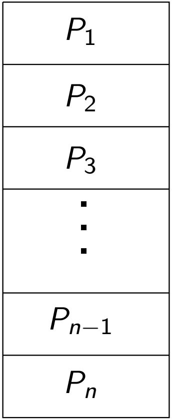
A Heartbleed Attack



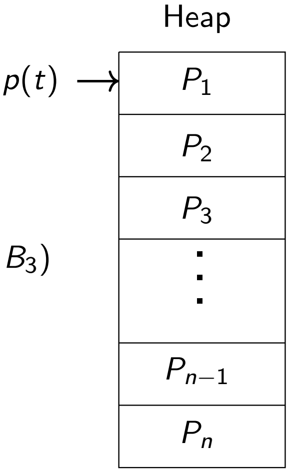
A Heartbleed Attack

$$
\begin{aligned}
a(t) &= (p(t), b(t)) \\
&= (P_1, B_1 + B_2 + B_3)
\end{aligned}
$$

Heap

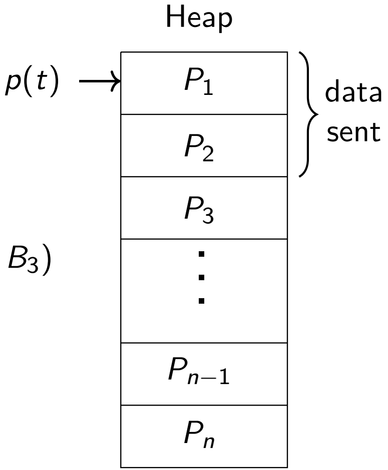


A Heartbleed Attack



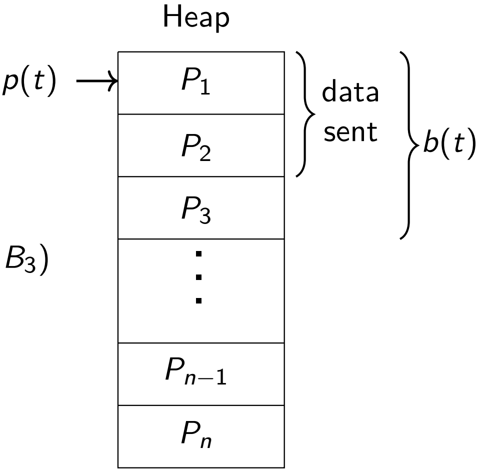
$$\begin{aligned} a(t) &= (p(t), b(t)) \\ &= (P_1, B_1 + B_2 + B_3) \end{aligned}$$

A Heartbleed Attack



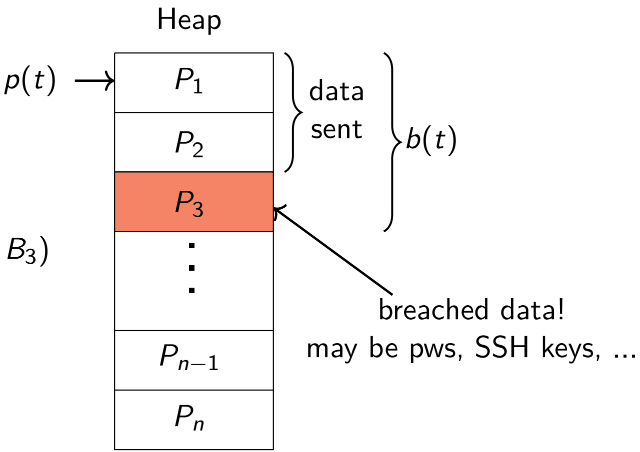
$$\begin{aligned} a(t) &= (p(t), b(t)) \\ &= (P_1, B_1 + B_2 + B_3) \end{aligned}$$

A Heartbleed Attack



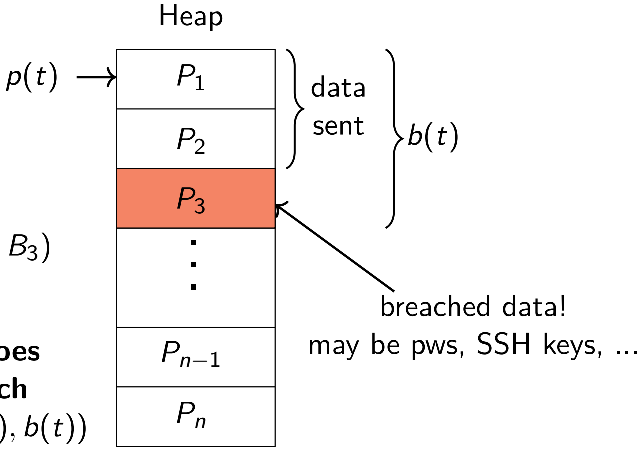
$$\begin{aligned} a(t) &= (p(t), b(t)) \\ &= (P_1, B_1 + B_2 + B_3) \end{aligned}$$

A Heartbleed Attack



$$\begin{aligned} a(t) &= (p(t), b(t)) \\ &= (P_1, B_1 + B_2 + B_3) \end{aligned}$$

A Heartbleed Attack

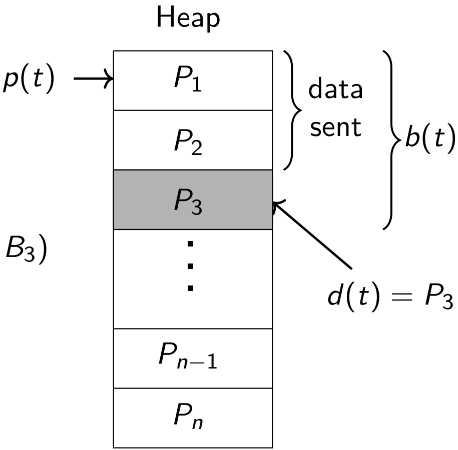


$$
\begin{aligned}
a(t) &= (p(t), b(t)) \\
&= (P_1, B_1 + B_2 + B_3)
\end{aligned}
$$

the attacker does not know which combinations $(p(t), b(t))$ will leak the sensitive data

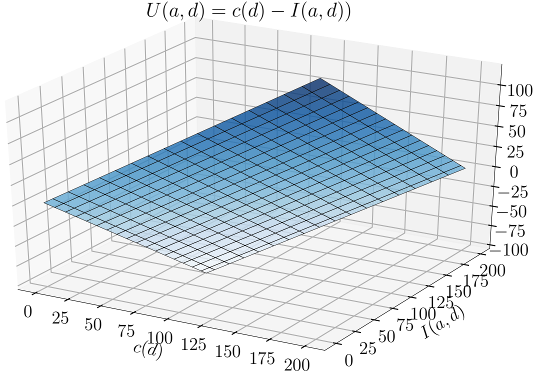
A Detected and Prevented Attack

$$
\begin{aligned}
a(t) &= (p(t), b(t)) \\
&= (P_1, B_1 + B_2 + B_3)
\end{aligned}
$$



The Utility Function

$$U(a,d) = \underbrace{c(d)}_{\text{response time}} - \underbrace{I(a,d)}_{\#\text{ requests}} \tag{1}$$



- The defender's goal is to minimize utility
- I.e. minimize response times and maximize requests between attacks

First Proposed Algorithm: "Adaptive RL Algorithm"

- **Assume attacker uses the same action w.p** $1 - \epsilon_{a(t)}$ and selects new action w.p $\epsilon_{a(t)}$ decided by ALG_A (which is unknown).
- Assume $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$

First Proposed Algorithm: "Adaptive RL Algorithm"

- **Assume attacker uses the same action w.p** $1 - \epsilon_{a(t)}$ and selects new action w.p $\epsilon_{a(t)}$ decided by ALG_A (which is unknown).
- Assume $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$

First Proposed Algorithm: “Adaptive RL Algorithm”

- **Assume attacker uses the same action w.p** $1 - \epsilon_{a(t)}$ and selects new action w.p $\epsilon_{a(t)}$ decided by ALG_A (which is unknown).
- Assume $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$

Algorithm 1: Adaptive reinforcement learning algorithm

$d(0) \leftarrow \text{sample}(\mathcal{D})$;
$a(0) \leftarrow \text{sample}(\mathcal{A})$;
$u(0) \leftarrow U(d(0), a(0))$;
$d(1) \leftarrow d(0)$;
$u(1) \leftarrow u(0)$;
while $t \geq 2$ **do**
  $d^{\text{tp}} \leftarrow \text{sample}(\mathcal{D} \setminus \{d(t), d(t-1)\})$ with prob. $\epsilon_d(t)$;
  if $u(t) < u(t-1)$ **then**
    $d^{\text{tp}} \leftarrow d(t)$ with prob. $(1 - \epsilon_d(t))$;
  else
    $d^{\text{tp}} \leftarrow d(t-1)$ with prob. $(1 - \epsilon_d(t))$;
  $d(t+1) \leftarrow d^{\text{tp}}$;
  $a^{\text{tp}} \leftarrow \text{ALG}_a([d(t) \;\; a(t)]^T)$ with prob. $\epsilon_a(t)$;
  $a^{\text{tp}} \leftarrow a(t)$ with prob. $1 - \epsilon_a(t)$;
  $a(t+1) \leftarrow a^{\text{tp}}$;
  $u(t+1) \leftarrow U(d(t+1), a(t+1))$;

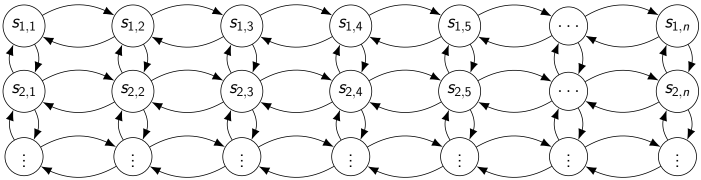
First Proposed Algorithm: "Adaptive RL Algorithm"

- **Assume attacker uses the same action w.p** $1 - \epsilon_{a(t)}$ and selects new action w.p $\epsilon_{a(t)}$ decided by ALG_A (which is unknown).
- Assume $\lim_{t \to \infty} \epsilon_{a(t)} = 0$ and $\lim_{t \to \infty} \epsilon_{d(t)} = 0$

- **In essence**:
  - if current defender action was better than the previous action, use same action w.p $1 - \epsilon_{d(t)}$
  - otherwise use previous action w.p $1 - \epsilon_{d(t)}$
  - Select random action w.p $\epsilon_{d(t)}$

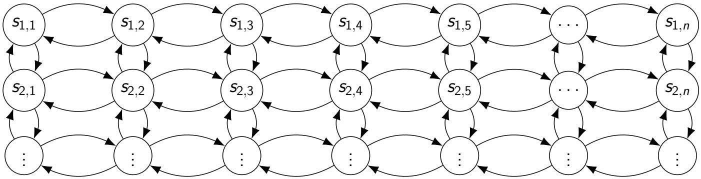
Theoretical Analysis of the First Algorithm: TLDR;

- Given the fixed defender policy & attacker policy, the sequence of actions **forms a Markov chain** $\mathcal{P}_t$
- The stationary distribution with high probability will consist of states that are optimal for the defender.



Theoretical Analysis of the First Algorithm: TLDR;

- Given the fixed defender policy & attacker policy, the sequence of actions **forms a Markov chain** $\mathcal{P}_t$
- The stationary distribution with high probability will consist of states that are optimal for the defender.



Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t \geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution

- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t \geq 1}$
- Since we assume $\lim_{t \to \infty} \epsilon_{a(t)} = 0$ and $\lim_{t \to \infty} \epsilon_{d(t)} = 0$, we have a limiting stationary distribution $\lim_{t \to \infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t \geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution

- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t \geq 1}$
- Since we assume $\lim_{t \to \infty} \epsilon_{a(t)} = 0$ and $\lim_{t \to \infty} \epsilon_{d(t)} = 0$, we have a unique limiting stationary distribution $\lim_{t \to \infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t \geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution

- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t \geq 1}$
- Since we assume $\lim_{t \to \infty} \epsilon_{a(t)} = 0$ and $\lim_{t \to \infty} \epsilon_{d(t)} = 0$, we have a unique limiting stationary distribution $\lim_{t \to \infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t\geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution
- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t\geq 1}$
- Since we assume $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$, we have a unique limiting stationary distribution $\lim_{t\to\infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t\geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution
- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t\geq 1}$
- Since we assume $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$, we have a unique limiting stationary distribution $\lim_{t\to\infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t\geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution
- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t\geq 1}$
- Since we assume $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$, we have a unique limiting stationary distribution $\lim_{t\to\infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t\geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution

- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t\geq 1}$
- Since we assume $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$, we have a unique limiting stationary distribution $\lim_{t\to\infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- Let the previous actions of each agent be the state $s = ((d(t), d(t-1)), (a(t), a(t-1)))$
- Fix the exploration rate $\epsilon_t = [\epsilon_{a(t)}, \epsilon_{d(t)}]$
- Then $(s_t)_{t \geq 1}$ **is a Markov chain** $\mathcal{P}_t$ (policies are fixed with ϵ fixed)
- Assume that the Markov chain is irreducible and aperiodic.
- Then, the Markov chain has a unique stationary distribution μ_t
- Playing the game will converge to this distribution

- Now let ϵ vary with t, then we get a sequence of stationary distributions $(\mu_t)_{t \geq 1}$
- Since we assume $\lim_{t \to \infty} \epsilon_{a(t)} = 0$ and $\lim_{t \to \infty} \epsilon_{d(t)} = 0$, we have a unique limiting stationary distribution $\lim_{t \to \infty} \mu_t = \mu^*$ (Lemma 4.1).

Theoretical Analysis of the First Algorithm

- **We want to show that** μ^* with high probability has the best defender response.
- Define the set of best responses: $S_{BR} = \{s = (d, a) \in \mathcal{S} | U(d, a) = \min_{d' \in \mathcal{D}} U(d', a)\}$.

Theoretical Analysis of the First Algorithm

- **We want to show that** μ^* with high probability has the best defender response.
- Define the set of best responses:
  $S_{BR} = \{s = (d, a) \in \mathcal{S} | U(d, a) = \min_{d' \in \mathcal{D}} U(d', a)\}$.

Theoretical Analysis of the First Algorithm

- **We want to show that** μ^* with high probability has the best defender response.
- Define the set of best responses:
  $S_{BR} = \{s = (d, a) \in \mathcal{S} | U(d, a) = \min_{d' \in \mathcal{D}} U(d', a)\}$.

Theorem

Consider the Markov chain $\mathcal{P}_t$ induced by the RL algorithm. Then,

$$\lim_{t \to \infty} \mathbb{P}[s_t \in S_{BR} \times S_{BR}] = 1 \tag{2}$$

Theoretical Analysis of the First Algorithm

- **We want to show that** μ^* with high probability has the best defender response.
- Define the set of best responses: $S_{BR} = \{s = (d, a) \in \mathcal{S} | U(d, a) = \min_{d' \in \mathcal{D}} U(d', a)\}$.

Theorem

Consider the Markov chain $\mathcal{P}_t$ induced by the RL algorithm. Then,

$$\lim_{t \to \infty} \mathbb{P}[s_t \in S_{BR} \times S_{BR}] = 1 \tag{3}$$

- The proof is based on the theory of resistance trees
- Based on the fact that exploration diminishes and defender always selects best action according to past

Second Proposed Algorithm: “Robust RL Algorithm”

- Drop assumption that $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$
- This means that the adaptive algorithm will not converge

- The “robust” algorithm keeps a history $h(t) = ((u(0), a(0), d(0), \ldots, (u(t), a(t), d(t))$.
- Define $D_{MM}(t)$ to be the set of minmax actions based on $h(t)$.
- $M(d, t) \triangleq \max_{0\leq s\leq t, d(s)=d} u(t)$,
  $D_{MM}(t) \triangleq \{d | M(d, t) \leq M(d', t) \quad \forall d' \in \mathcal{D}\}$
- **At each step t, w.p $1 - \epsilon_{d(t)}$ sample an action** $d(t) \sim D_{MM}(t)$
- w.p $\epsilon_{d(t)}$ sample a random new action, i.e. $d(t) \sim \mathcal{D} \setminus D_{MM}(t)$.

Second Proposed Algorithm: "Robust RL Algorithm"

- Drop assumption that $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$
- This means that the adaptive algorithm will not converge
- The "robust" algorithm keeps a history $h(t) = ((u(0), a(0), d(0), \ldots, (u(t), a(t), d(t)).$
- Define $D_{MM}(t)$ to be the set of minmax actions based on $h(t)$.
- $M(d,t) \triangleq \max_{0 \leq s \leq t, d(s)=d} u(t),$ $D_{MM}(t) \triangleq \{d | M(d,t) \leq M(d',t) \quad \forall d' \in \mathcal{D}\}$
- **At each step t, w.p $1-\epsilon_{d(t)}$ sample an action** $d(t) \sim D_{MM}(t)$
- w.p $\epsilon_{d(t)}$ sample a random new action, i.e. $d(t) \sim \mathcal{D} \setminus D_{MM}(t)$.

Second Proposed Algorithm: "Robust RL Algorithm"

- Drop assumption that $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$
- This means that the adaptive algorithm will not converge

- The "robust" algorithm keeps a history $h(t) = ((u(0), a(0), d(0), \ldots, (u(t), a(t), d(t)).$
- Define $D_{MM}(t)$ to be the set of minmax actions based on $h(t)$.
- $M(d,t) \triangleq \max_{0\le s\le t, d(s)=d} u(t)$,
  $D_{MM}(t) \triangleq \{d | M(d,t) \le M(d',t) \quad \forall d' \in \mathcal{D}\}$
- **At each step t, w.p $1-\epsilon_{d(t)}$ sample an action** $d(t) \sim D_{MM}(t)$
- w.p $\epsilon_{d(t)}$ sample a random new action, i.e. $d(t) \sim \mathcal{D} \setminus D_{MM}(t)$.

Second Proposed Algorithm: "Robust RL Algorithm"

- Drop assumption that $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$
- This means that the adaptive algorithm will not converge

- The "robust" algorithm keeps a history $h(t) = ((u(0), a(0), d(0), \ldots, (u(t), a(t), d(t))$.
- Define $D_{MM}(t)$ to be the set of minmax actions based on $h(t)$.
- $M(d,t) \triangleq \max_{0\leq s\leq t, d(s)=d} u(t)$,
  $D_{MM}(t) \triangleq \{d | M(d,t) \leq M(d',t) \quad \forall d' \in \mathcal{D}\}$
- **At each step t, w.p $1 - \epsilon_{d(t)}$ sample an action** $d(t) \sim D_{MM}(t)$
- w.p $\epsilon_{d(t)}$ sample a random new action, i.e. $d(t) \sim \mathcal{D} \setminus D_{MM}(t)$.

Second Proposed Algorithm: "Robust RL Algorithm"

- Drop assumption that $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$
- This means that the adaptive algorithm will not converge

- The "robust" algorithm keeps a history $h(t) = ((u(0), a(0), d(0), \ldots, (u(t), a(t), d(t))$.
- Define $D_{MM}(t)$ to be the set of minmax actions based on $h(t)$.
- $M(d,t) \triangleq \max_{0\leq s\leq t, d(s)=d} u(t)$,
  $D_{MM}(t) \triangleq \{d | M(d,t) \leq M(d',t) \quad \forall d' \in \mathcal{D}\}$
- **At each step t, w.p $1-\epsilon_{d(t)}$ sample an action** $d(t) \sim D_{MM}(t)$
- w.p $\epsilon_{d(t)}$ sample a random new action, i.e. $d(t) \sim \mathcal{D} \setminus D_{MM}(t)$.

Second Proposed Algorithm: "Robust RL Algorithm"

- Drop assumption that $\lim_{t\to\infty} \epsilon_{a(t)} = 0$ and $\lim_{t\to\infty} \epsilon_{d(t)} = 0$
- This means that the adaptive algorithm will not converge

- The "robust" algorithm keeps a history $h(t) = ((u(0), a(0), d(0), \ldots, (u(t), a(t), d(t)).$
- Define $D_{MM}(t)$ to be the set of minmax actions based on $h(t)$.
- $M(d,t) \triangleq \max_{0\le s\le t, d(s)=d} u(t)$,
  $D_{MM}(t) \triangleq \{d | M(d,t) \le M(d',t) \quad \forall d' \in \mathcal{D}\}$
- **At each step t, w.p $1-\epsilon_{d(t)}$ sample an action** $d(t) \sim D_{MM}(t)$
- w.p $\epsilon_{d(t)}$ sample a random new action, i.e. $d(t) \sim \mathcal{D} \setminus D_{MM}(t)$.

Theoretical Analysis of the Second Algorithm

- Due to the non-diminishing exploration, the best-response action may change
- Recall: with w.p $\epsilon_{d(t)}$ the defender always **selects a random action**
- Recall: with w.p $1 - \epsilon_{d(t)}$ the defender **selects an action greedily based on the set** $D_{MM}(t)$
- We want to show that $D_{MM}(t)$ converges to the mini-max set D_{MM},
  - i.e. w.p $1 - \epsilon_{d(t)}$ the defender selects an action that minimizes the utility against at least one attacker action.
- By definition of $D_{MM}(t)$, if all states of the Markov chain $\mathcal{P}_t$ have been visited, then $D_{MM}(t) = D_{MM}$
- **Hence it is sufficient to show that** $\mathcal{P}_t$ **visits** $\mathcal{S}$ **as** $t \to \infty$

Theoretical Analysis of the Second Algorithm

- Due to the non-diminishing exploration, the best-response action may change
- Recall: with w.p $\epsilon_{d(t)}$ the defender always **selects a random action**
- Recall: with w.p $1 - \epsilon_{d(t)}$ the defender **selects an action greedily based on the set $D_{MM}(t)$**
- We want to show that $D_{MM}(t)$ converges to the mini-max set D_{MM},
  - i.e. w.p $1 - \epsilon_{d(t)}$ the defender selects an action that minimizes the utility against at least one attacker action.
- By definition of $D_{MM}(t)$, if all states of the Markov chain $\mathcal{P}_t$ have been visited, then $D_{MM}(t) = D_{MM}$
- **Hence it is sufficient to show that $\mathcal{P}_t$ visits $\mathcal{S}$ as $t \to \infty$**

Theoretical Analysis of the Second Algorithm

- Due to the non-diminishing exploration, the best-response action may change
- Recall: with w.p $\epsilon_{d(t)}$ the defender always **selects a random action**
- Recall: with w.p $1 - \epsilon_{d(t)}$ the defender **selects an action greedily based on the set** $D_{MM}(t)$
- We want to show that $D_{MM}(t)$ converges to the mini-max set D_{MM},
  - i.e. w.p $1 - \epsilon_{d(t)}$ the defender selects an action that minimizes the utility against at least one attacker action.
- By definition of $D_{MM}(t)$, if all states of the Markov chain $\mathcal{P}_t$ have been visited, then $D_{MM}(t) = D_{MM}$
- **Hence it is sufficient to show that** $\mathcal{P}_t$ **visits** $\mathcal{S}$ **as** $t \to \infty$

Theoretical Analysis of the Second Algorithm

- Due to the non-diminishing exploration, the best-response action may change
- Recall: with w.p $\epsilon_{d(t)}$ the defender always **selects a random action**
- Recall: with w.p $1 - \epsilon_{d(t)}$ the defender **selects an action greedily based on the set** $D_{MM}(t)$
- We want to show that $D_{MM}(t)$ converges to the mini-max set D_{MM},
  - i.e. w.p $1 - \epsilon_{d(t)}$ the defender selects an action that minimizes the utility against at least one attacker action.
- By definition of $D_{MM}(t)$, if all states of the Markov chain $\mathcal{P}_t$ have been visited, then $D_{MM}(t) = D_{MM}$
- **Hence it is sufficient to show that $\mathcal{P}_t$ visits $\mathcal{S}$ as $t \to \infty$**

Theoretical Analysis of the Second Algorithm

- Let $\mathcal{P}_t$ be the Markov chain induced by the robust RL algorithm.
- Since $\mathcal{P}_t$ is irreducible and aperiodic by assumption, it will visit $\mathcal{S}$ as $t \to \infty$
- $\implies$ the robust RL algorithm converges to the minimax strategy

Theoretical Analysis of the Second Algorithm

- Let $\mathcal{P}_t$ be the Markov chain induced by the robust RL algorithm.
- Since $\mathcal{P}_t$ is irreducible and aperiodic by assumption, it will visit $\mathcal{S}$ as $t \to \infty$
- $\implies$ the robust RL algorithm converges to the minimax strategy

Theoretical Analysis of the Second Algorithm

- Let $\mathcal{P}_t$ be the Markov chain induced by the robust RL algorithm.
- Since $\mathcal{P}_t$ is irreducible and aperiodic by assumption, it will visit $\mathcal{S}$ as $t \to \infty$
- $\implies$ the robust RL algorithm converges to the minimax strategy

Strong points of the Paper

- **Real-world Use Case**
  - Easy to relate to the model by using well known vulnerability
- **The Formal Analysis**
  - Convergence proofs

Limitations Drawbacks of the Paper

- **A bit unorthodox approach**
  - Minimize utility instead of maximize
  - Apply RL to a non-sequential decision problem
  - Custom model, does not use existing frameworks (e.g. MDP, normal game)
- **Simplifying assumptions**
  - Non-rational/strategic attacker
  - Assume specific exploration rates
  - Assume static system
- **Abstract analysis only**
  - No attempt to evaluate in a realistic environment

Conclusions

- Adaptive Cyber Defense against Heartbleed attacks
- Custom model and very simple reinforcement learning algorithms
- Nice theoretical guarantees
- Abstract model and evaluation