Learning Security Strategies through Game Play and Optimal Stopping

ICML 22', Baltimore, 17/7-23/7 2022
Machine Learning for Cyber Security Workshop

Kim Hammar & Rolf Stadler

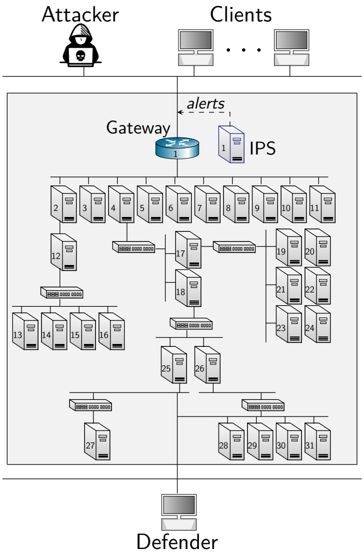
kimham@kth.se & *stadler@kth.se*

Division of Network and Systems Engineering
KTH Royal Institute of Technology

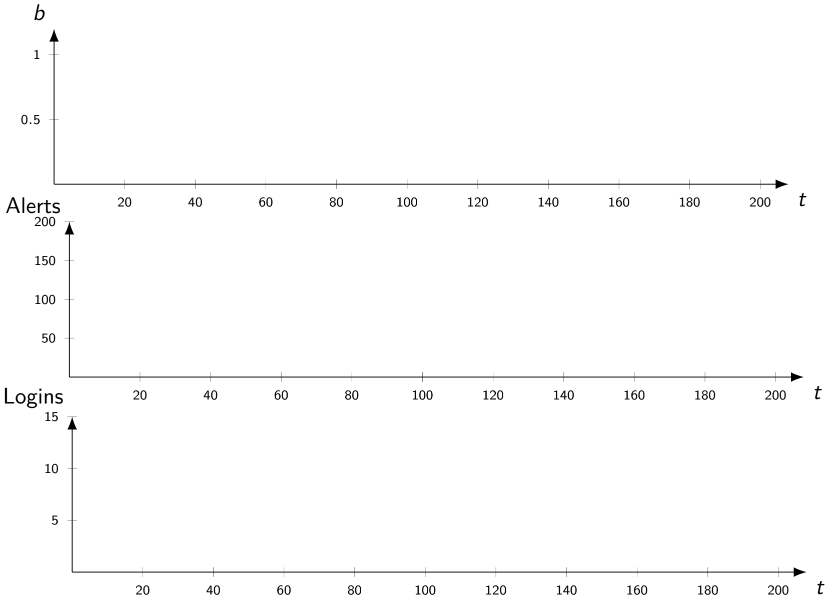
Mar 18, 2022

Use Case: Intrusion Prevention

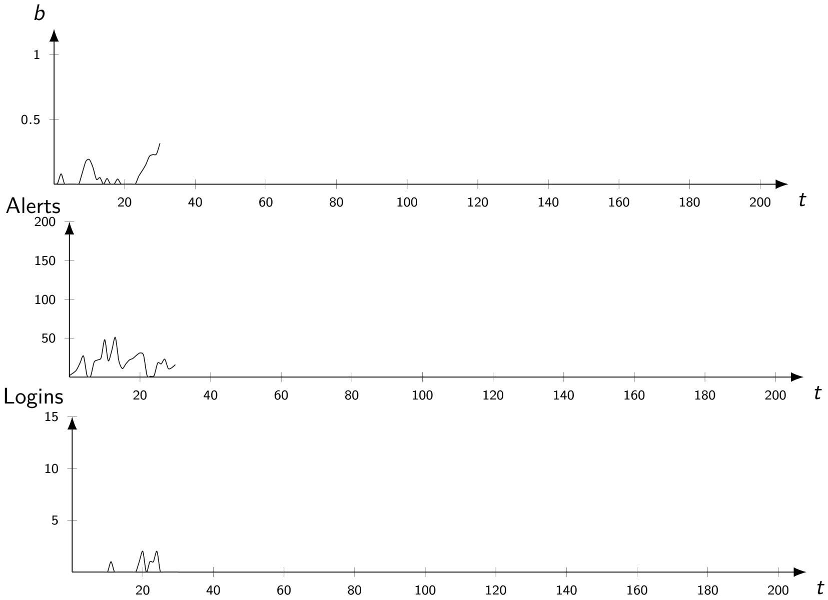
- A **Defender** owns an infrastructure
  - Consists of connected components
  - Components run network services
  - Defender defends the infrastructure by monitoring and active defense
  - Has partial observability
- An **Attacker** seeks to intrude on the infrastructure
  - Has a partial view of the infrastructure
  - Wants to compromise specific components
  - Attacks by reconnaissance, exploitation and pivoting



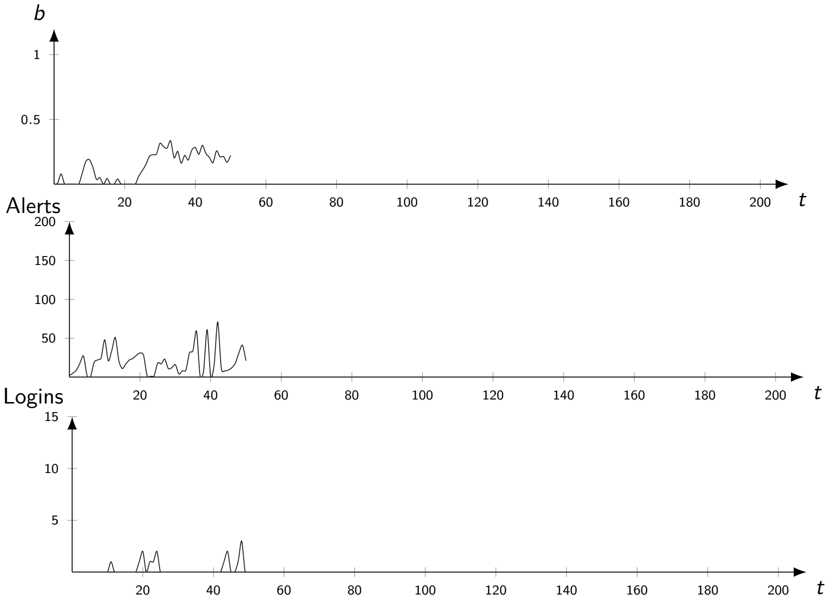
The Intrusion Prevention Problem



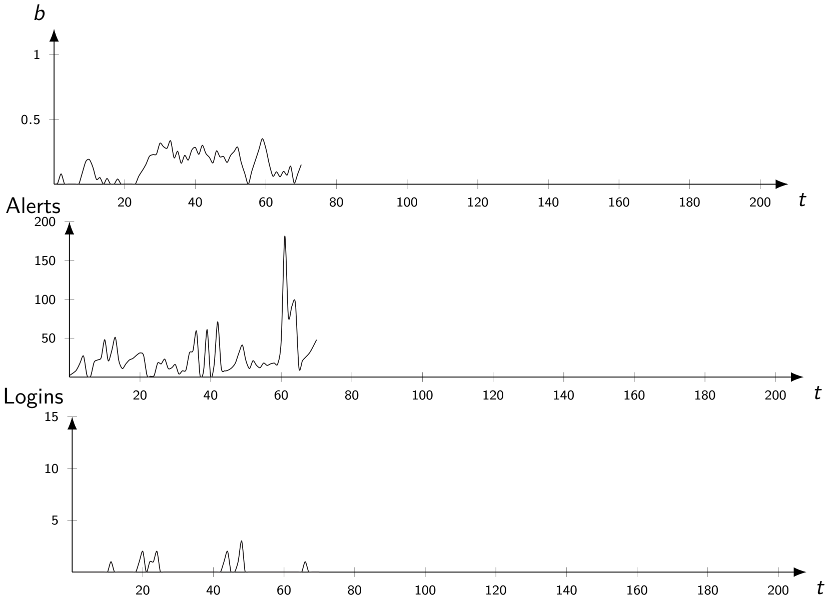
The Intrusion Prevention Problem



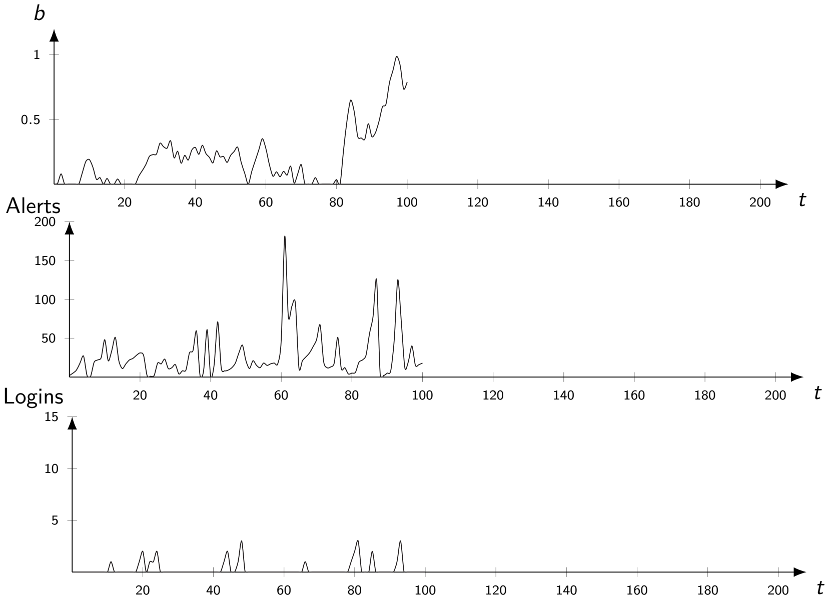
The Intrusion Prevention Problem



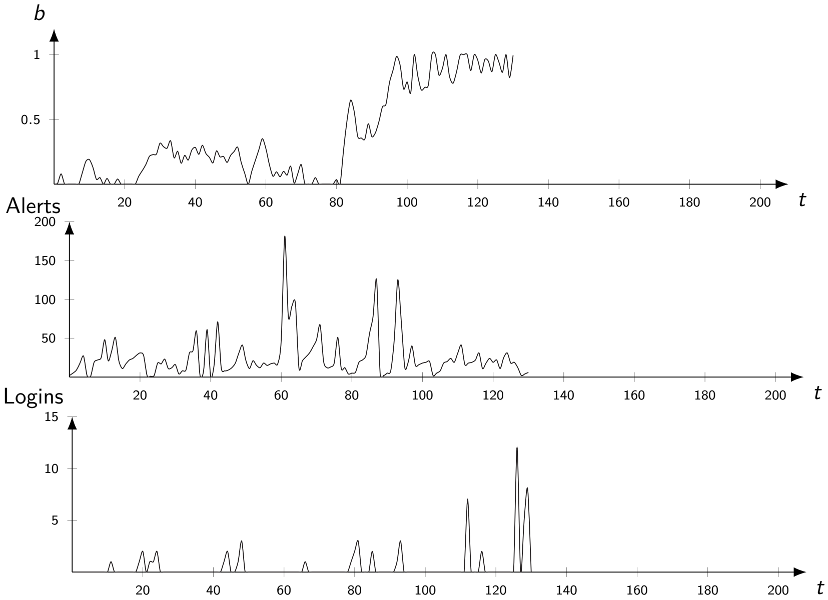
The Intrusion Prevention Problem



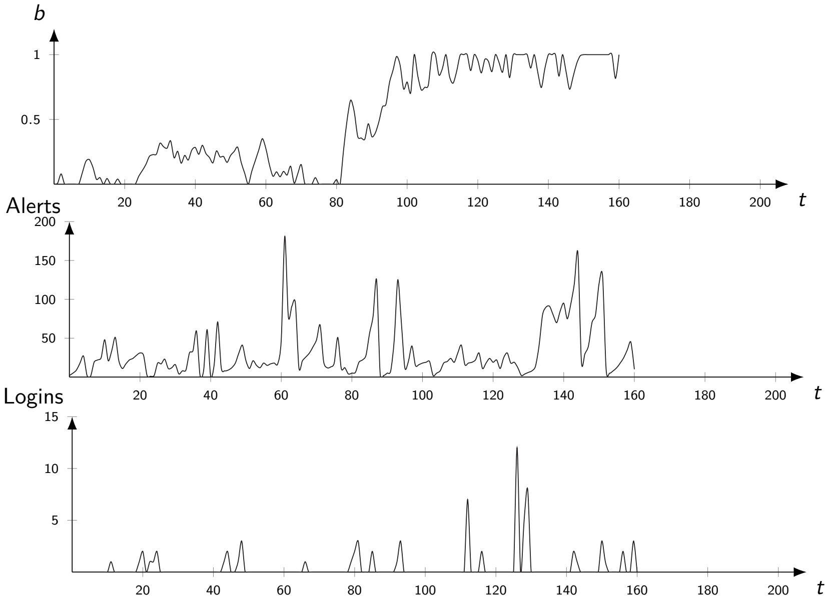
The Intrusion Prevention Problem



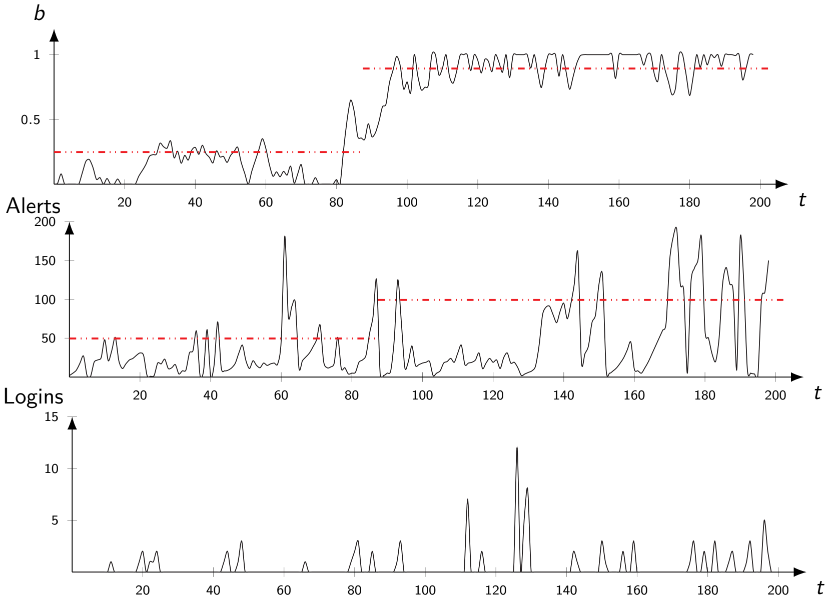
The Intrusion Prevention Problem



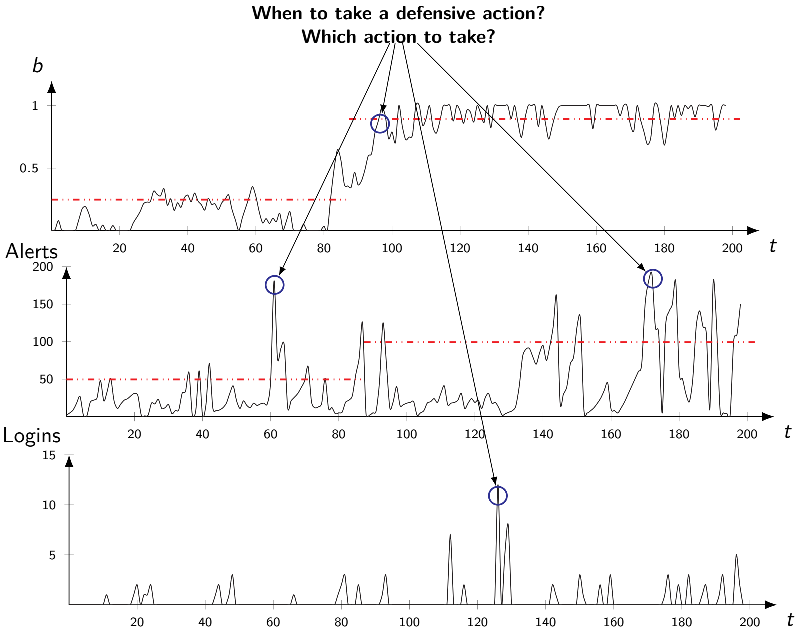
The Intrusion Prevention Problem



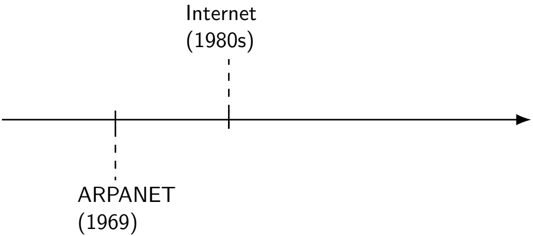
The Intrusion Prevention Problem



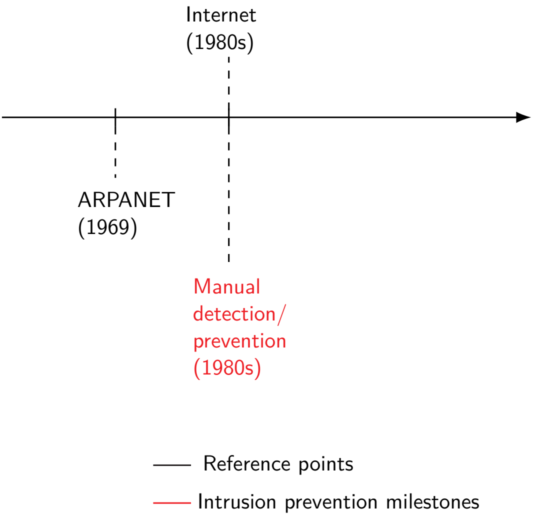
The Intrusion Prevention Problem



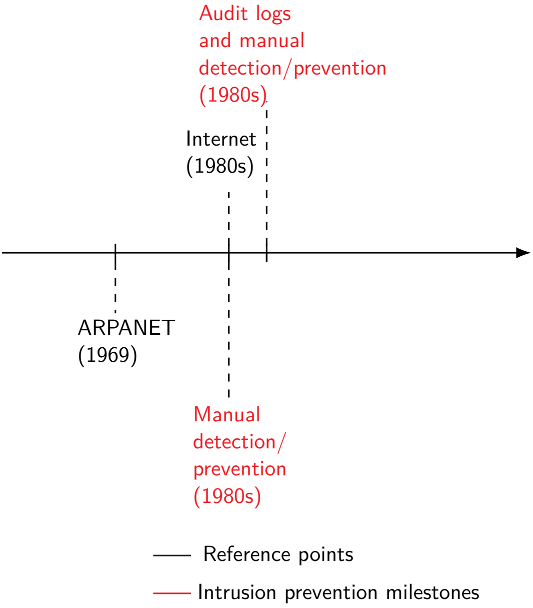
A Brief History of Intrusion Prevention



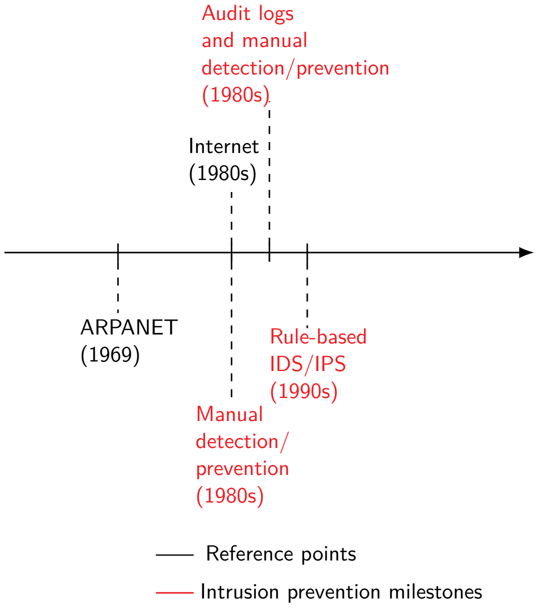
A Brief History of Intrusion Prevention



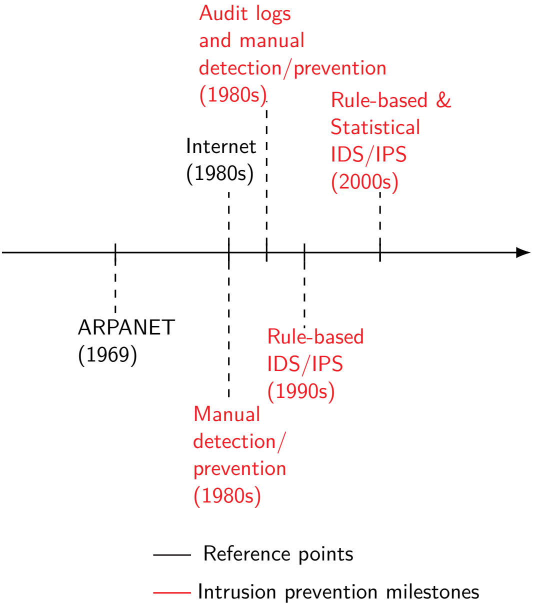
A Brief History of Intrusion Prevention



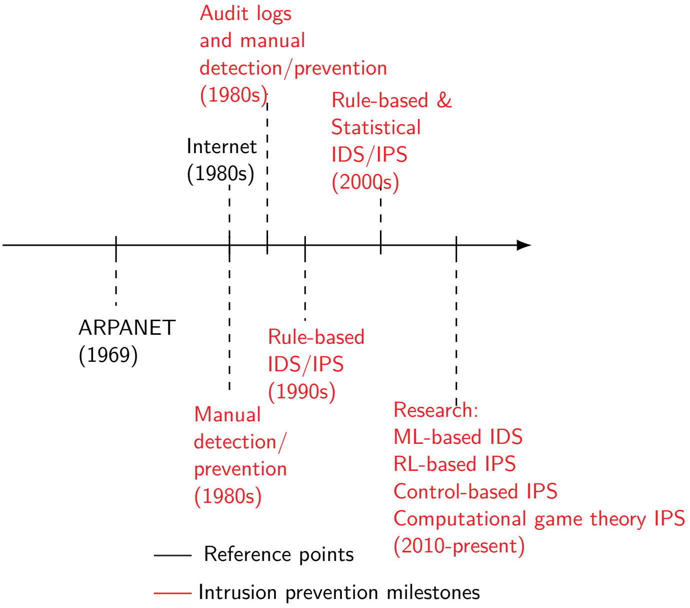
A Brief History of Intrusion Prevention



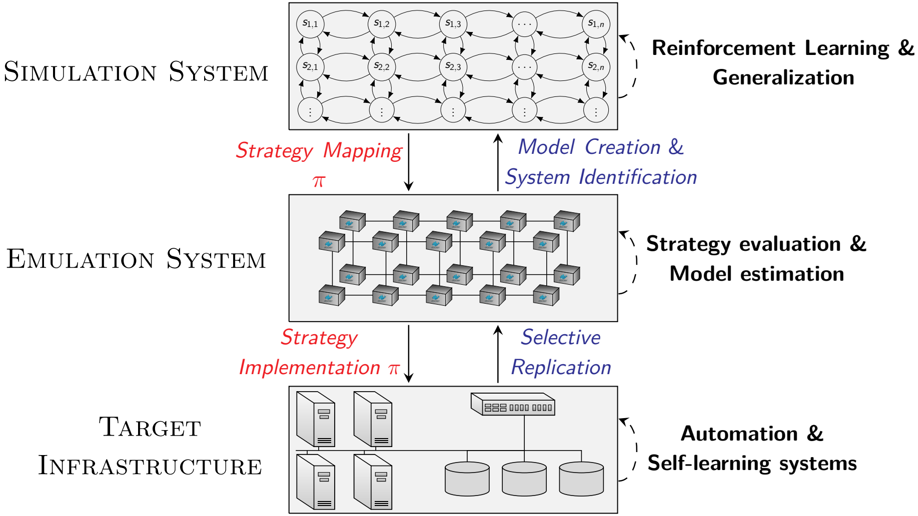
A Brief History of Intrusion Prevention



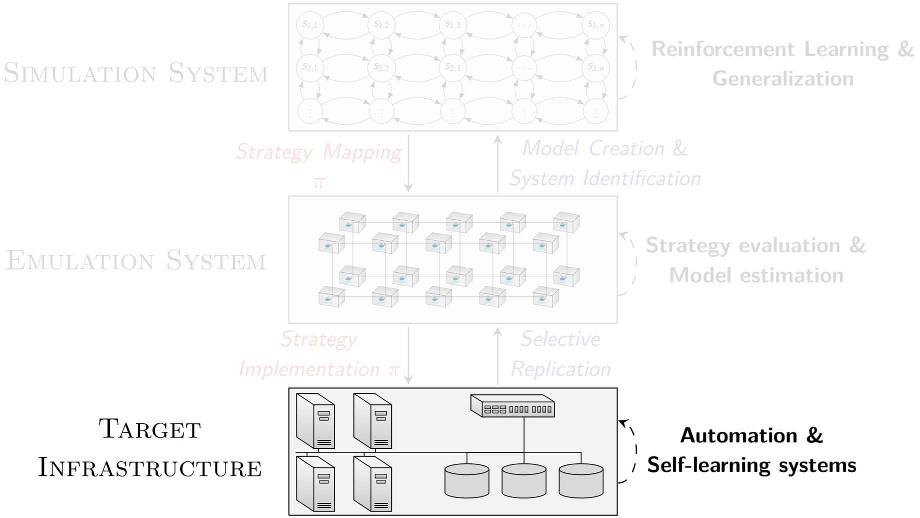
A Brief History of Intrusion Prevention



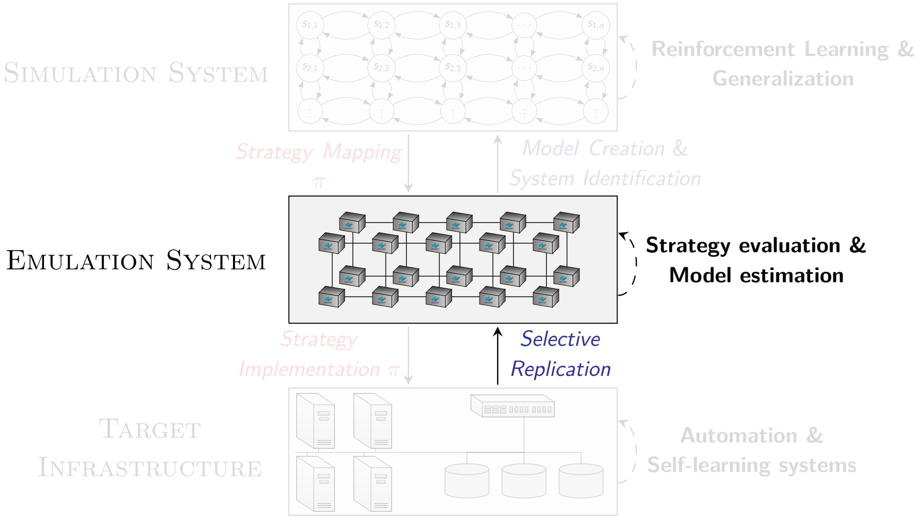
Our Approach for Learning Effective Security Strategies



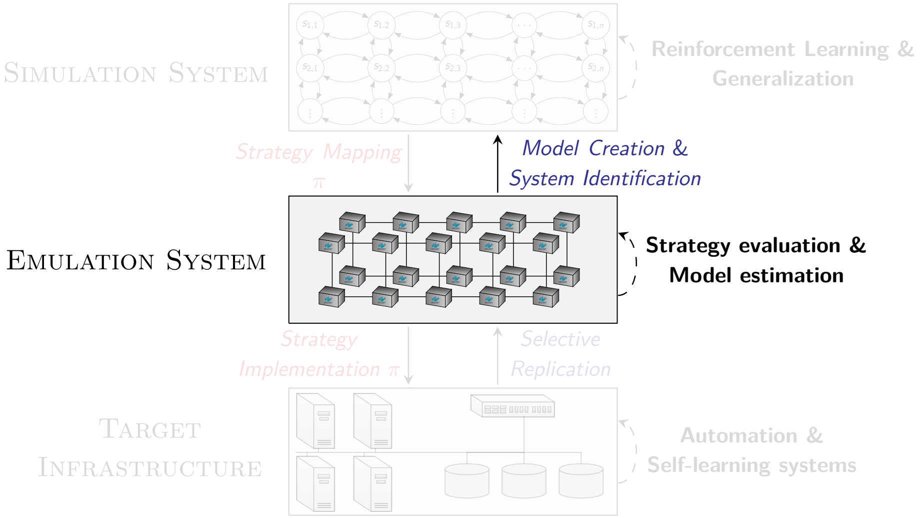
Our Approach for Finding Effective Security Strategies



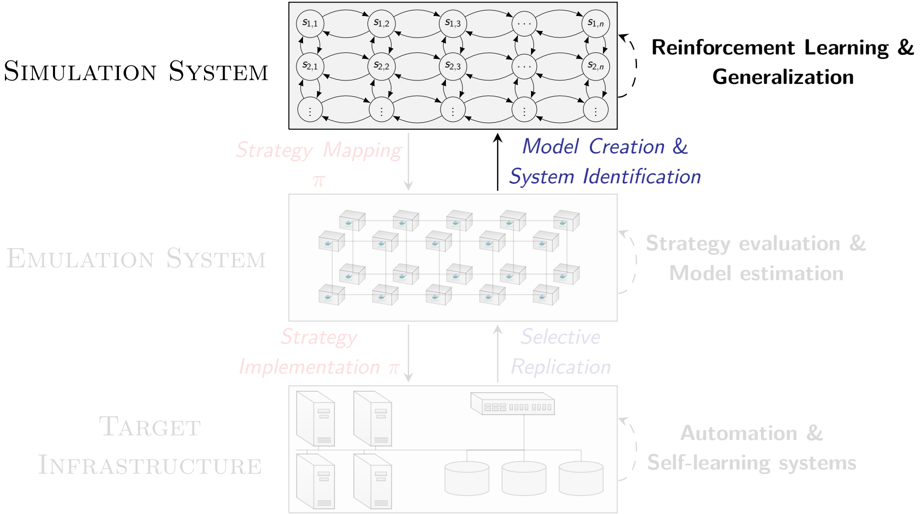
Our Approach for Finding Effective Security Strategies



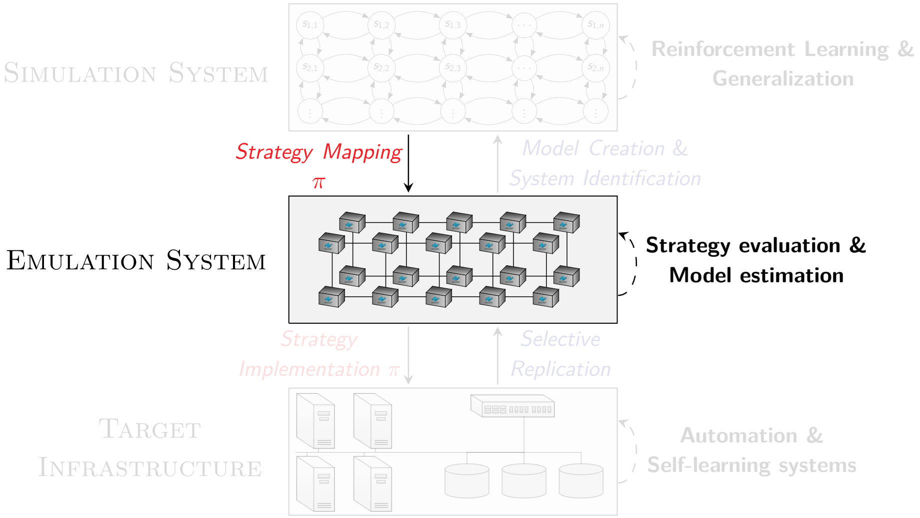
Our Approach for Finding Effective Security Strategies



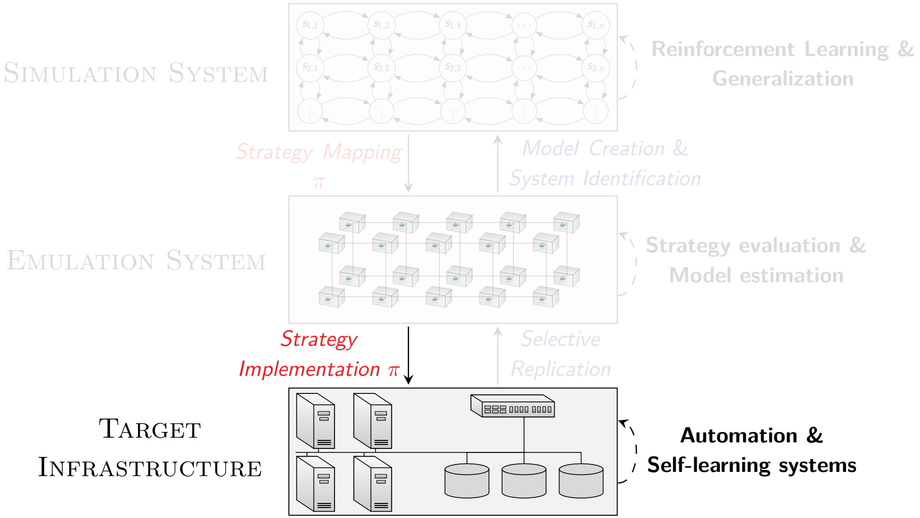
Our Approach for Finding Effective Security Strategies



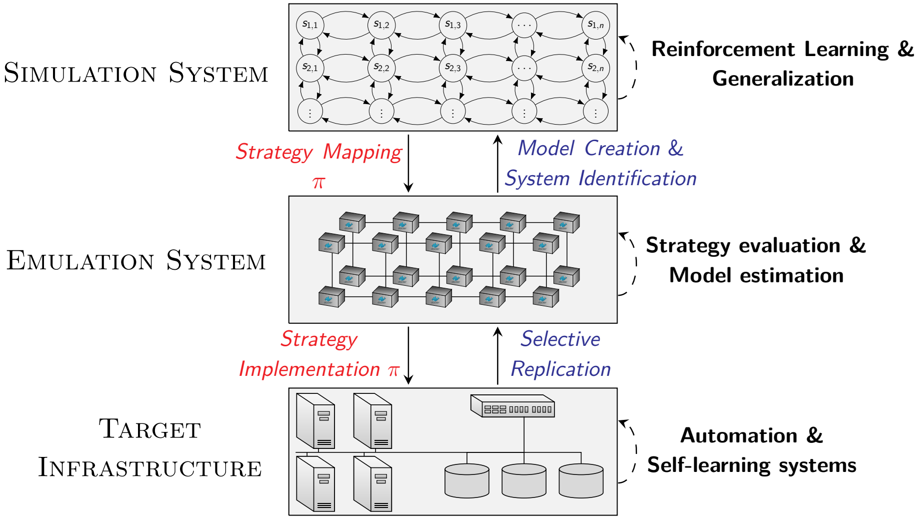
Our Approach for Finding Effective Security Strategies



Our Approach for Finding Effective Security Strategies



Our Approach for Finding Effective Security Strategies



Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: $\mathrm{T\text{-}FP}$
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Outline

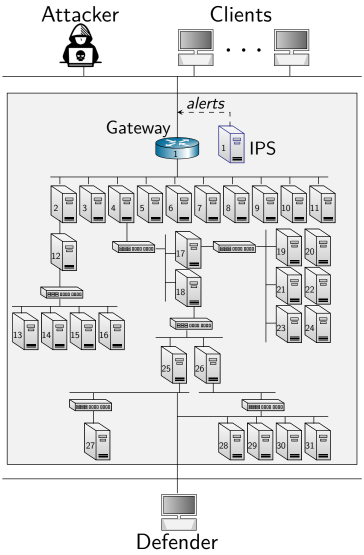
- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of** $(\tilde{\pi}_1, \tilde{\pi}_2)$
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

The Optimal Stopping Game

- **Defender**:
  - Has a pre-defined ordered list of defensive measures:
    1. Revoke user certificates
    2. Blacklist IPs
    3. Drop traffic that generates IPS alerts of priority 1 — 4
    4. Block gateway
  - Defender's strategy decides when to take each action
- **Attacker**:
  - Has a pre-defined randomized intrusion sequence of reconnaissance and exploit commands:
    1. TCP-SYN scan
    2. CVE-2017-7494
    3. CVE-2015-3306
    4. CVE-2015-5602
    5. SSH brute-force
    6. . . .
  - Attacker's strategy decides when to start/stop an intrusion

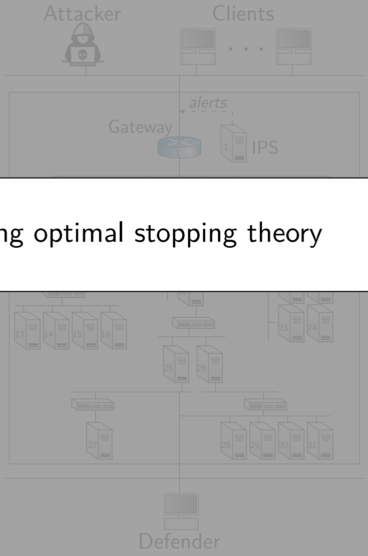


The Optimal Stopping Game

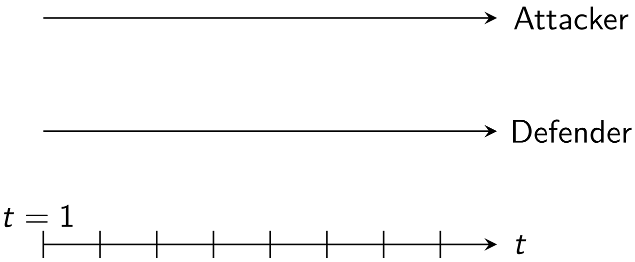
- **Defender**:
  - Has a pre-defined ordered list of defensive measures:
    1. Revoke user certificates
    2. Blacklist IPs
    3. Drop traffic that generates IPS alerts of priority 1 — 4
    4. Block gateway

We analyze attacker/defender strategies using optimal stopping theory

  - Has a pre-defined randomized intrusion sequence of reconnaissance and exploit commands:
    1. TCP-SYN scan
    2. CVE-2017-7494
    3. CVE-2015-3306
    4. CVE-2015-5602
    5. SSH brute-force
    6. . . .
  - Attacker's stratregy decides when to start/stop an intrusion

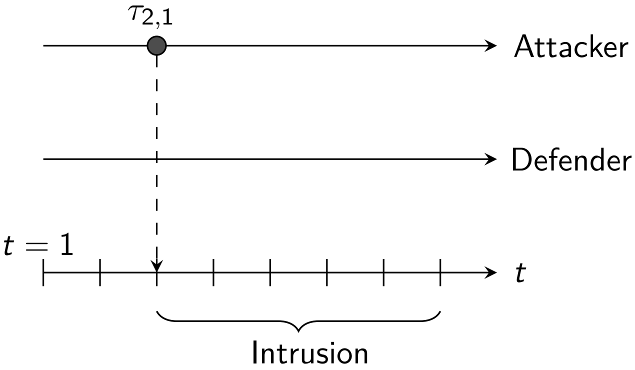


Optimal Stopping Formulation of Intrusion Prevention



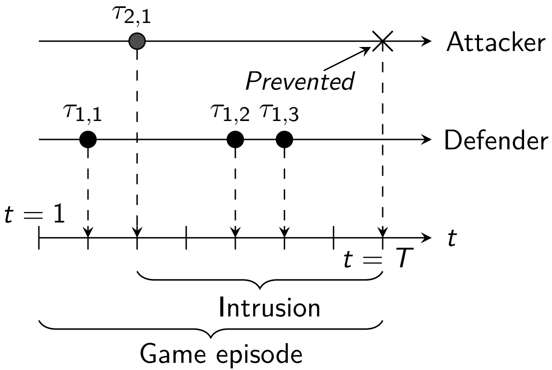
- The attacker's stopping times $\tau_{2,1}$ and $\tau_{2,2}$ determine the times to start/stop the intrusion
  - During the intrusion, the attacker follows a fixed intrusion strategy
- The defender's stopping times $\tau_{1,L}, \tau_{1,L-1}, \ldots$ determine the times to take defensive actions

Optimal Stopping Formulation of Intrusion Prevention



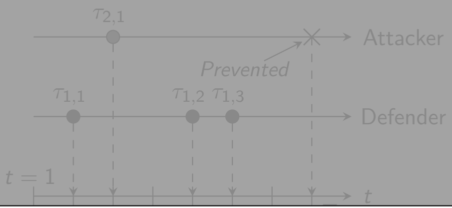
- The attacker's stopping times $\tau_{2,1}$ and $\tau_{2,2}$ determine the times to start/stop the intrusion
  - During the intrusion, the attacker follows a fixed intrusion strategy
- The defender's stopping times $\tau_{1,L}, \tau_{1,L-1}, \ldots$ determine the times to take defensive actions

Optimal Stopping Formulation of Intrusion Prevention



- The attacker's stopping times $\tau_{2,1}$ and $\tau_{2,2}$ determine the times to start/stop the intrusion
  - During the intrusion, the attacker follows a fixed intrusion strategy
- The defender's stopping times $\tau_{1,L}, \tau_{1,L-1}, \ldots$ determine the times to take defensive actions

Optimal Stopping Formulation of Intrusion Prevention

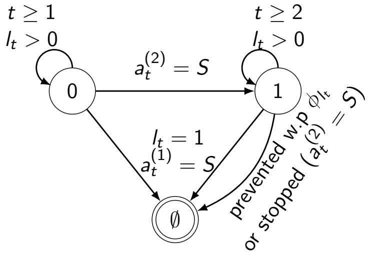


We model this game as a **zero-sum** partially observed stochastic game

- The attacker's stopping times $\tau_{2,1}, \tau_{2,2}, \ldots$ determine the times to start/stop the intrusion
  - During the intrusion, the attacker follows a fixed intrusion strategy
- The defender's stopping times $\tau_{1,1}, \tau_{1,2}, \ldots$ determine the times to update the IPS configuration

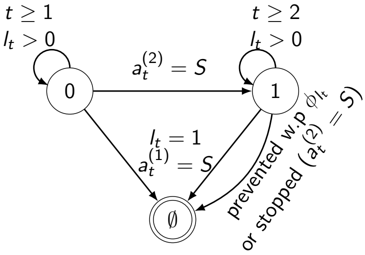
Partially Observed Stochastic Game

- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - Number of IPS Alerts $o_t \in \mathcal{O}$, defender stops remaining $l_t \in \{1, .., L\}$, o_t is drawn from r.v. $O \sim f_O(\cdot|s_t)$
- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞



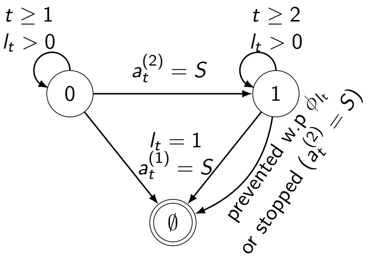
Partially Observed Stochastic Game

- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - Number of IPS Alerts $o_t \in \mathcal{O}$, defender stops remaining $l_t \in \{1, .., L\}$, o_t is drawn from r.v. $O \sim f_O(\cdot|s_t)$
- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞



Partially Observed Stochastic Game

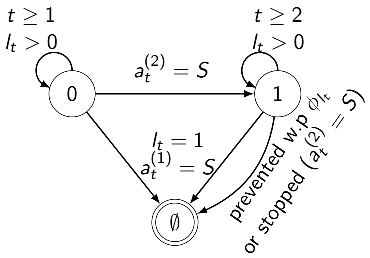
- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - Number of IPS Alerts $o_t \in \mathcal{O}$, defender stops remaining $l_t \in \{1, .., L\}$, o_t is drawn from r.v. $O \sim f_O(\cdot|s_t)$



- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞

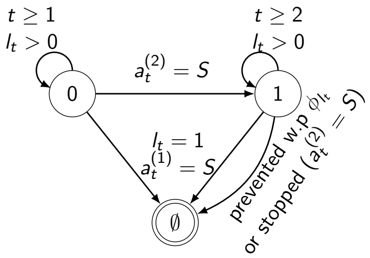
Partially Observed Stochastic Game

- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - Number of IPS Alerts $o_t \in \mathcal{O}$, defender stops remaining $l_t \in \{1, .., L\}$, o_t is drawn from r.v. $O \sim f_O(\cdot|s_t)$
- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞



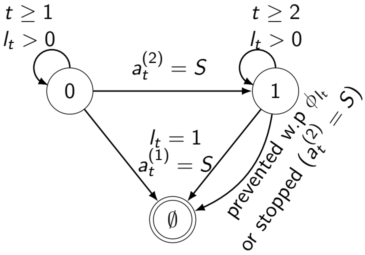
Partially Observed Stochastic Game

- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - Number of IPS Alerts $o_t \in \mathcal{O}$, defender stops remaining $l_t \in \{1, .., L\}$, o_t is drawn from r.v. $O \sim f_O(\cdot|s_t)$
- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞



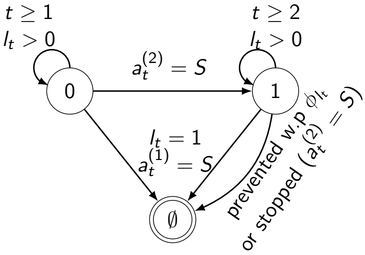
Partially Observed Stochastic Game

- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - Number of IPS Alerts $o_t \in \mathcal{O}$, defender stops remaining $l_t \in \{1, .., L\}$, o_t is drawn from r.v. $O \sim f_O(\cdot|s_t)$
- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞



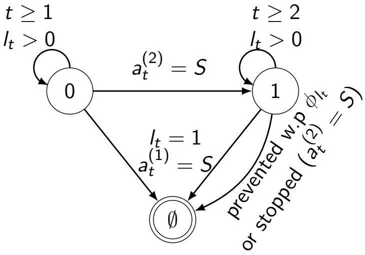
Partially Observed Stochastic Game

- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - Number of IPS Alerts $o_t \in \mathcal{O}$, defender stops remaining $l_t \in \{1, .., L\}$, o_t is drawn from r.v. $O \sim f_O(\cdot|s_t)$
- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞



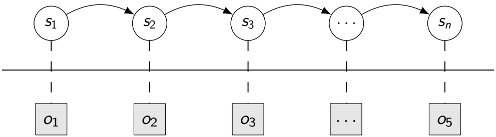
Partially Observed Stochastic Game

- **Players:** $\mathcal{N} = \{1, 2\}$ (Defender=1)
- **States:** Intrusion $s_t \in \{0, 1\}$, terminal $\emptyset$.
- **Observations:**
  - IPS Alerts $\Delta x_{1,t}, \Delta x_{2,t}, \ldots, \Delta x_{M,t}$, defender stops remaining $l_t \in \{1, .., L\}$, $f_X(\Delta x_1, \ldots, \Delta x_M | s_t)$
- **Actions:**
  - $\mathcal{A}_1 = \mathcal{A}_2 = \{S, C\}$
- **Rewards:**
  - Defender reward: security and service.
  - Attacker reward: negative of defender reward.
- **Transition probabilities:**
  - Follows from game dynamics.
- **Horizon:**
  - ∞

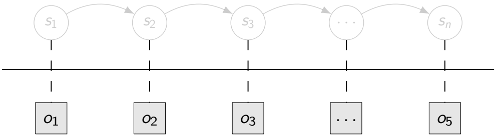


One-Sided Partial Observability

- We assume that the **attacker has perfect information**. Only the **defender has partial information**.
- The attacker's view:



- The defender's view:



One-Sided Partial Observability

- We assume that the **attacker has perfect information**. Only the **defender has partial information**.
- The attacker's view:

- The defender's view:

- Makes it tractable to compute the defender's belief $b_t^{(1)}(s_t) = \mathbb{P}[s_t | h_t]$ (avoid nested beliefs)

Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Game Analysis

- Defender strategy is of the form: $\pi_{1,l} : \mathcal{B} \rightarrow \Delta(\mathcal{A}_1)$
- Attacker strategy is of the form: $\pi_{2,l} : \mathcal{S} \times \mathcal{B} \rightarrow \Delta(\mathcal{A}_2)$

- **Defender and attacker objectives**:

$$J_1(\pi_{1,l}, \pi_{2,l}) = \mathbb{E}_{(\pi_{1,l}, \pi_{2,l})} \left[\sum_{t=1}^{\infty} \gamma^{t-1} \mathcal{R}_l(s_t, \boldsymbol{a}_t) \right]$$

$$J_2(\pi_{1,l}, \pi_{2,l}) = -J_1$$

- **Best response correspondences**:

$$B_1(\pi_{2,l}) = \arg\max_{\pi_{1,l} \in \Pi_1} J_1(\pi_{1,l}, \pi_{2,l})$$

$$B_2(\pi_{1,l}) = \arg\max_{\pi_{2,l} \in \Pi_2} J_2(\pi_{1,l}, \pi_{2,l})$$

- **Nash equilibrium** $(\pi^*_{1,l}, \pi^*_{2,l})$:

$$\pi^*_{1,l} \in B_1(\pi^*_{2,l}) \text{ and } \pi^*_{2,l} \in B_2(\pi^*_{1,l}) \quad (1)$$

Game Analysis

- Defender strategy is of the form: $\pi_{1,l} : \mathcal{B} \rightarrow \Delta(\mathcal{A}_1)$
- Attacker strategy is of the form: $\pi_{2,l} : \mathcal{S} \times \mathcal{B} \rightarrow \Delta(\mathcal{A}_2)$

- **Defender and attacker objectives**:

$$J_1(\pi_{1,l}, \pi_{2,l}) = \mathbb{E}_{(\pi_{1,l}, \pi_{2,l})}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \mathcal{R}_l(s_t, \boldsymbol{a}_t)\right]$$

$$J_2(\pi_{1,l}, \pi_{2,l}) = -J_1$$

- **Best response correspondences**:

$$B_1(\pi_{2,l}) = \arg\max_{\pi_{1,l} \in \Pi_1} J_1(\pi_{1,l}, \pi_{2,l})$$

$$B_2(\pi_{1,l}) = \arg\max_{\pi_{2,l} \in \Pi_2} J_2(\pi_{1,l}, \pi_{2,l})$$

- **Nash equilibrium** $(\pi^*_{1,l}, \pi^*_{2,l})$:

$$\pi^*_{1,l} \in B_1(\pi^*_{2,l}) \text{ and } \pi^*_{2,l} \in B_2(\pi^*_{1,l}) \tag{2}$$

Game Analysis

- Defender strategy is of the form: $\pi_{1,l} : \mathcal{B} \rightarrow \Delta(\mathcal{A}_1)$
- Attacker strategy is of the form: $\pi_{2,l} : \mathcal{S} \times \mathcal{B} \rightarrow \Delta(\mathcal{A}_2)$

- **Defender and attacker objectives**:

$$J_1(\pi_{1,l}, \pi_{2,l}) = \mathbb{E}_{(\pi_{1,l}, \pi_{2,l})} \left[\sum_{t=1}^{\infty} \gamma^{t-1} \mathcal{R}_l(s_t, \boldsymbol{a}_t) \right]$$

$$J_2(\pi_{1,l}, \pi_{2,l}) = -J_1$$

- **Best response correspondences**:

$$B_1(\pi_{2,l}) = \arg\max_{\pi_{1,l} \in \Pi_1} J_1(\pi_{1,l}, \pi_{2,l})$$

$$B_2(\pi_{1,l}) = \arg\max_{\pi_{2,l} \in \Pi_2} J_2(\pi_{1,l}, \pi_{2,l})$$

- **Nash equilibrium** $(\pi_{1,l}^*, \pi_{2,l}^*)$:

$$\pi_{1,l}^* \in B_1(\pi_{2,l}^*) \text{ and } \pi_{2,l}^* \in B_2(\pi_{1,l}^*) \quad (3)$$

Game Analysis

- Defender strategy is of the form: $\pi_{1,l} : \mathcal{B} \rightarrow \Delta(\mathcal{A}_1)$
- Attacker strategy is of the form: $\pi_{2,l} : \mathcal{S} \times \mathcal{B} \rightarrow \Delta(\mathcal{A}_2)$

- **Defender and attacker objectives**:

$$J_1(\pi_{1,l}, \pi_{2,l}) = \mathbb{E}_{(\pi_{1,l}, \pi_{2,l})}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \mathcal{R}_l(s_t, \boldsymbol{a}_t)\right]$$

$$J_2(\pi_{1,l}, \pi_{2,l}) = -J_1$$

- **Best response correspondences**:

$$B_1(\pi_{2,l}) = \arg\max_{\pi_{1,l} \in \Pi_1} J_1(\pi_{1,l}, \pi_{2,l})$$

$$B_2(\pi_{1,l}) = \arg\max_{\pi_{2,l} \in \Pi_2} J_2(\pi_{1,l}, \pi_{2,l})$$

- **Nash equilibrium** $(\pi_{1,l}^*, \pi_{2,l}^*)$:

$$\pi_{1,l}^* \in B_1(\pi_{2,l}^*) \text{ and } \pi_{2,l}^* \in B_2(\pi_{1,l}^*)$$

Game Analysis

Theorem

Given the one-sided POSG Γ with $L \geq 1$, the following holds.

(A) *Γ has a mixed Nash equilibrium. Further, Γ has a pure Nash equilibrium when $s = 0 \iff b(1) = 0$.*

(B) *Given any attacker strategy $\pi_{2,l} \in \Pi_2$, if $f_{O|s}$ is totally positive of order 2, there exist values $\tilde{\alpha}_1 \geq \tilde{\alpha}_2 \geq \ldots \geq \tilde{\alpha}_L \in [0,1]$ and a defender best response strategy $\tilde{\pi}_{1,l} \in B_1(\pi_{2,l})$ that satisfies:*

$$\tilde{\pi}_{1,l}(b(1)) = S \iff b(1) \geq \tilde{\alpha}_l \qquad l \in 1, \ldots, L \tag{4}$$

(C) *Given a defender strategy $\pi_{1,l} \in \Pi_1$, where $\pi_{1,l}(S|b(1))$ is non-decreasing in $b(1)$ and $\pi_{1,l}(S|1) = 1$, there exist values $\tilde{\beta}_{0,1}, \tilde{\beta}_{1,1}, \ldots, \tilde{\beta}_{0,L}, \tilde{\beta}_{1,L} \in [0,1]$ and a best response strategy $\tilde{\pi}_{2,l} \in B_2(\pi_{1,l})$ of the attacker that satisfies:*

$$\tilde{\pi}_{2,l}(0, b(1)) = C \iff \pi_{1,l}(S|b(1)) \geq \tilde{\beta}_{0,l} \tag{5}$$

$$\tilde{\pi}_{2,l}(1, b(1)) = S \iff \pi_{1,l}(S|b(1)) \geq \tilde{\beta}_{1,l} \tag{6}$$

Game Analysis

Theorem

Given the one-sided POSG Γ with $L \geq 1$, the following holds.

(A) *Γ has a mixed Nash equilibrium. Further, Γ has a pure Nash equilibrium when $s = 0 \iff b(1) = 0$.*

(B) *Given any attacker strategy $\pi_{2,l} \in \Pi_2$, if $f_{O|s}$ is totally positive of order 2, there exist values $\tilde{\alpha}_1 \geq \tilde{\alpha}_2 \geq \ldots \geq \tilde{\alpha}_L \in [0,1]$ and a defender best response strategy $\tilde{\pi}_{1,l} \in B_1(\pi_{2,l})$ that satisfies:*

$$\tilde{\pi}_{1,l}(b(1)) = S \iff b(1) \geq \tilde{\alpha}_l \qquad l \in 1, \ldots, L \tag{7}$$

(C) *Given a defender strategy $\pi_{1,l} \in \Pi_1$, where $\pi_{1,l}(S|b(1))$ is non-decreasing in $b(1)$ and $\pi_{1,l}(S|1) = 1$, there exist values $\tilde{\beta}_{0,1}, \tilde{\beta}_{1,1}, \ldots, \tilde{\beta}_{0,L}, \tilde{\beta}_{1,L} \in [0,1]$ and a best response strategy $\tilde{\pi}_{2,l} \in B_2(\pi_{1,l})$ of the attacker that satisfies:*

$$\tilde{\pi}_{2,l}(0, b(1)) = C \iff \pi_{1,l}(S|b(1)) \geq \tilde{\beta}_{0,l} \tag{8}$$

$$\tilde{\pi}_{2,l}(1, b(1)) = S \iff \pi_{1,l}(S|b(1)) \geq \tilde{\beta}_{1,l} \tag{9}$$

Game Analysis

Theorem

Given the one-sided POSG Γ with $L \geq 1$, the following holds.

(A) *Γ has a mixed Nash equilibrium. Further, Γ has a pure Nash equilibrium when $s = 0 \iff b(1) = 0$.*

(B) *Given any attacker strategy $\pi_{2,l} \in \Pi_2$, if $f_{O|s}$ is totally positive of order 2, there exist values $\tilde{\alpha}_1 \geq \tilde{\alpha}_2 \geq \ldots \geq \tilde{\alpha}_L \in [0,1]$ and a defender best response strategy $\tilde{\pi}_{1,l} \in B_1(\pi_{2,l})$ that satisfies:*

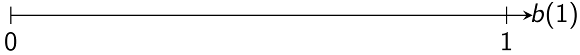
$$\tilde{\pi}_{1,l}(b(1)) = S \iff b(1) \geq \tilde{\alpha}_l \qquad l \in 1, \ldots, L \tag{10}$$

(C) *Given a defender strategy $\pi_{1,l} \in \Pi_1$, where $\pi_{1,l}(S|b(1))$ is non-decreasing in $b(1)$ and $\pi_{1,l}(S|1) = 1$, there exist values $\tilde{\beta}_{0,1}$, $\tilde{\beta}_{1,1}$, ..., $\tilde{\beta}_{0,L}$, $\tilde{\beta}_{1,L} \in [0,1]$ and a best response strategy $\tilde{\pi}_{2,l} \in B_2(\pi_{1,l})$ of the attacker that satisfies:*

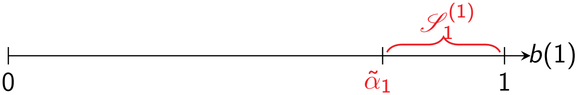
$$\tilde{\pi}_{2,l}(0, b(1)) = C \iff \pi_{1,l}(S|b(1)) \geq \tilde{\beta}_{0,l} \tag{11}$$

$$\tilde{\pi}_{2,l}(1, b(1)) = S \iff \pi_{1,l}(S|b(1)) \geq \tilde{\beta}_{1,l} \tag{12}$$

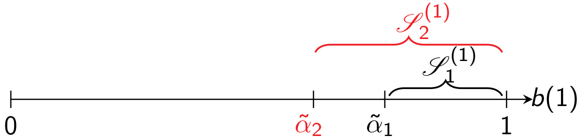
Structure of Best Response Strategies



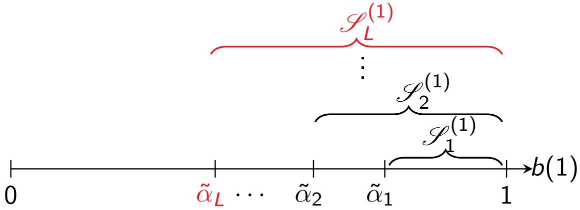
Structure of Best Response Strategies



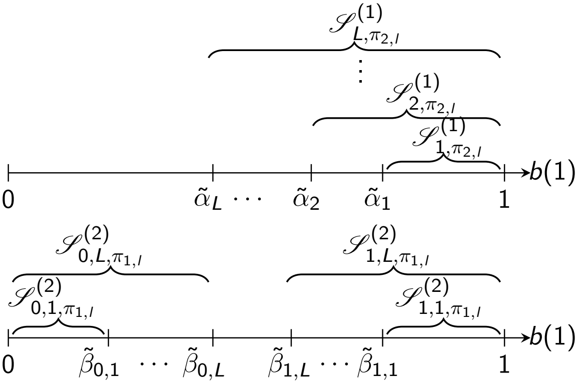
Structure of Best Response Strategies



Structure of Best Response Strategies



Structure of Best Response Strategies



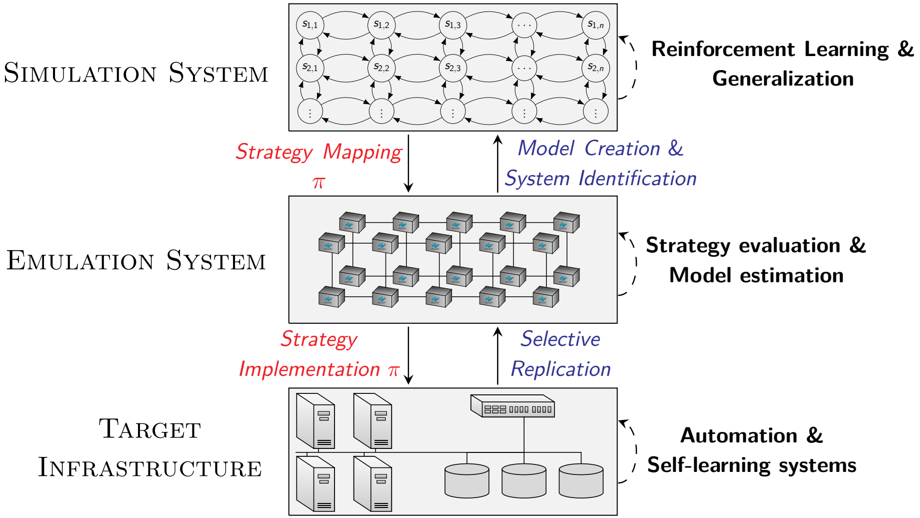
Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Outline

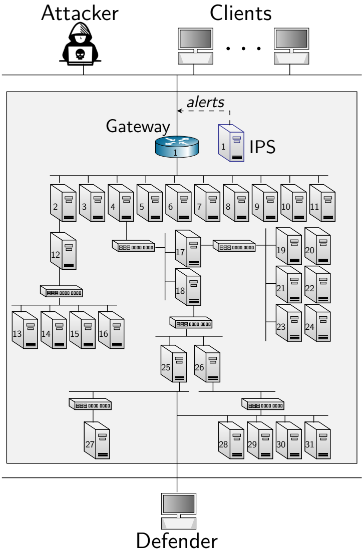
- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Our Method for Learning Effective Security Strategies

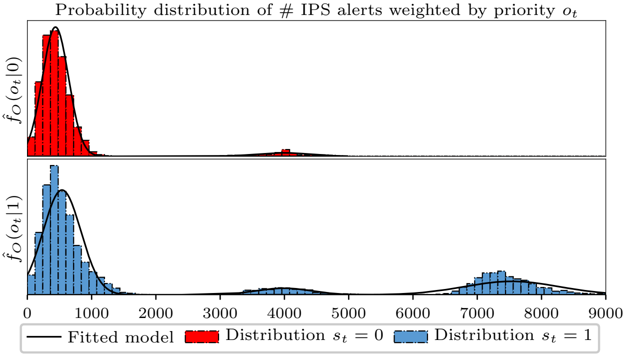


Emulating the Target Infrastructure

- Emulate **hosts** with docker containers
- Emulate **IPS and vulnerabilities** with software
- Network isolation and **traffic shaping** through NetEm in the Linux kernel
- Enforce **resource constraints** using cgroups.
- Emulate **client arrivals** with Poisson process
- **Internal connections** are full-duplex & loss-less with bit capacities of 1000 Mbit/s
- **External connections** are full-duplex with bit capacities of 100 Mbit/s & 0.1% packet loss in normal operation and random bursts of 1% packet loss



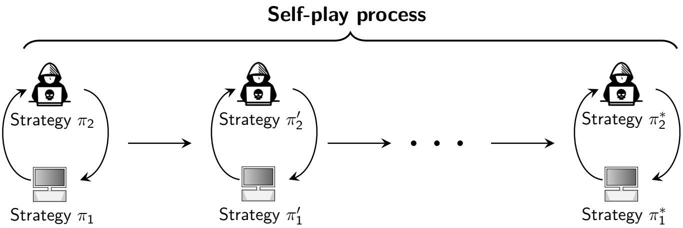
System Identification: Instantiating the Game Model based on Data from the Emulation



- We fit a Gaussian mixture distribution $\hat{f}_O$ as an estimate of f_O in the target infrastructure
- For each state s, we obtain the conditional distribution $\hat{f}_{O|s}$ through expectation-maximization

Our Reinforcement Learning Approach

- We learn a Nash equilibrium $(\pi^*_{1,l,\theta^{(1)}}, \pi^*_{2,l,\theta^{(2)}})$ through **fictitious self-play.**
- In each iteration:
  1. Learn a best response strategy of the defender by solving a POMDP $\tilde{\pi}_{1,l,\theta^{(1)}} \in B_1(\pi_{2,l,\theta^{(2)}})$.
  2. Learn a best response strategy of the attacker by solving an MDP $\tilde{\pi}_{2,l,\theta^{(2)}} \in B_2(\pi_{1,l,\theta^{(1)}})$.
  3. Store the best response strategies in two buffers Θ_1, Θ_2
  4. Update strategies to be the average of the stored strategies



(Pseudo-code is available in the paper)

Our Reinforcement Learning Algorithm for Learning Best-Response Threshold Strategies

- We **use the structural result** that threshold best response strategies exist (Theorem 1) **to design an efficient reinforcement learning algorithm** to learn best response strategies.

- We seek to learn:
  - L thresholds of the defender, $\tilde{\alpha}_1, \geq \tilde{\alpha}_2, \ldots, \geq \tilde{\alpha}_L \in [0,1]$
  - $2L$ thresholds of the attacker, $\tilde{\beta}_{0,1}, \tilde{\beta}_{1,1}, \ldots, \tilde{\beta}_{0,L}, \tilde{\beta}_{1,L} \in [0,1]$

- We learn these thresholds iteratively through Robbins and Monro's stochastic approximation algorithm.[1]

[1] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400 –407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

Our Reinforcement Learning Algorithm for Learning Best-Response Threshold Strategies

- We **use the structural result** that threshold best response strategies exist (Theorem 1) **to design an efficient reinforcement learning algorithm** to learn best response strategies.

- We seek to learn:
  - *L* thresholds of the defender, $\tilde{\alpha}_1, \geq \tilde{\alpha}_2, \ldots, \geq \tilde{\alpha}_L \in [0,1]$
  - $2L$ thresholds of the attacker, $\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_L \in [0,1]$

- We learn these thresholds iteratively through Robbins and Monro's stochastic approximation algorithm.[2]

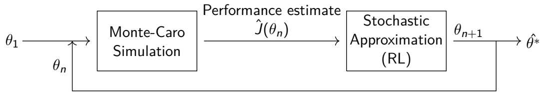
[2] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400 –407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

Our Reinforcement Learning Algorithm for Learning Best-Response Threshold Strategies

- We **use the structural result** that threshold best response strategies exist (Theorem 1) **to design an efficient reinforcement learning algorithm** to learn best response strategies.

- We seek to learn:
  - L thresholds of the defender, $\tilde{\alpha}_1, \geq \tilde{\alpha}_2, \ldots, \geq \tilde{\alpha}_L \in [0, 1]$
  - $2L$ thresholds of the attacker, $\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_L \in [0, 1]$

- We learn these thresholds iteratively through Robbins and Monro's stochastic approximation algorithm.[3]



[3] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. DOI: 10.1214/aoms/1177729586. URL: https://doi.org/10.1214/aoms/1177729586.

Our Reinforcement Learning Algorithm: T-FP

1. We learn the thresholds through simulation.
2. For each iteration $n \in \{1, 2, \ldots\}$, we perturb $\theta_n^{(i)}$ to obtain $\theta_n^{(i)} + c_n\Delta_n$ and $\theta_n^{(i)} - c_n\Delta_n$.
3. Then, we run two MDP or POMDP episodes
4. We then use the obtained episode outcomes $\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n)$ and $\hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)$ to estimate $\nabla_{\theta^{(i)}} J_i(\theta^{(i)})$ using the Simultaneous Perturbation Stochastic Approximation (SPSA) gradient estimator[4]:

$$\left(\hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})\right)_k = \frac{\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n) - \hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)}{2c_n(\Delta_n)_k}$$

5. Next, we use the estimated gradient and update the vector of thresholds through the stochastic approximation update:

$$\theta_{n+1}^{(i)} = \theta_n^{(i)} + a_n \hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})$$

[4] James C. Spall. "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* 37.3 (1992), pp. 332–341.

Our Reinforcement Learning Algorithm: T-FP

1. We learn the thresholds through simulation.
2. For each iteration $n \in \{1, 2, \ldots\}$, we perturb $\theta_n^{(i)}$ to obtain $\theta_n^{(i)} + c_n\Delta_n$ and $\theta_n^{(i)} - c_n\Delta_n$.
3. Then, we run two MDP or POMDP episodes
4. We then use the obtained episode outcomes $\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n)$ and $\hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)$ to estimate $\nabla_{\theta^{(i)}} J_i(\theta^{(i)})$ using the Simultaneous Perturbation Stochastic Approximation (SPSA) gradient estimator[4]:

$$\left(\hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})\right)_k = \frac{\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n) - \hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)}{2c_n(\Delta_n)_k}$$

5. Next, we use the estimated gradient and update the vector of thresholds through the stochastic approximation update:

$$\theta_{n+1}^{(i)} = \theta_n^{(i)} + a_n \hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})$$

[4] James C. Spall. "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* 37.3 (1992), pp. 332–341.

Our Reinforcement Learning Algorithm: T-FP

1. We learn the thresholds through simulation.
2. For each iteration $n \in \{1, 2, \ldots\}$, we perturb $\theta_n^{(i)}$ to obtain $\theta_n^{(i)} + c_n\Delta_n$ and $\theta_n^{(i)} - c_n\Delta_n$.
3. Then, we run two MDP or POMDP episodes
4. We then use the obtained episode outcomes $\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n)$ and $\hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)$ to estimate $\nabla_{\theta^{(i)}} J_i(\theta^{(i)})$ using the Simultaneous Perturbation Stochastic Approximation (SPSA) gradient estimator[4]:

$$\left(\hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})\right)_k = \frac{\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n) - \hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)}{2c_n(\Delta_n)_k}$$

5. Next, we use the estimated gradient and update the vector of thresholds through the stochastic approximation update:

$$\theta_{n+1}^{(i)} = \theta_n^{(i)} + a_n \hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})$$

---

[4] James C. Spall. "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* 37.3 (1992), pp. 332–341.

Our Reinforcement Learning Algorithm: T-FP

1. We learn the thresholds through simulation.
2. For each iteration $n \in \{1, 2, \dots\}$, we perturb $\theta_n^{(i)}$ to obtain $\theta_n^{(i)} + c_n\Delta_n$ and $\theta_n^{(i)} - c_n\Delta_n$.
3. Then, we run two MDP or POMDP episodes
4. We then use the obtained episode outcomes $\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n)$ and $\hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)$ to estimate $\nabla_{\theta^{(i)}} J_i(\theta^{(i)})$ using the Simultaneous Perturbation Stochastic Approximation (SPSA) gradient estimator[4]:

$$\left(\hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})\right)_k = \frac{\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n) - \hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)}{2c_n(\Delta_n)_k}$$

5. Next, we use the estimated gradient and update the vector of thresholds through the stochastic approximation update:

$$\theta_{n+1}^{(i)} = \theta_n^{(i)} + a_n \hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})$$

[4] James C. Spall. "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* 37.3 (1992), pp. 332–341.

Our Reinforcement Learning Algorithm: T-FP

1. We learn the thresholds through simulation.
2. For each iteration $n \in \{1, 2, \dots\}$, we perturb $\theta_n^{(i)}$ to obtain $\theta_n^{(i)} + c_n\Delta_n$ and $\theta_n^{(i)} - c_n\Delta_n$.
3. Then, we run two MDP or POMDP episodes
4. We then use the obtained episode outcomes $\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n)$ and $\hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)$ to estimate $\nabla_{\theta^{(i)}} J_i(\theta^{(i)})$ using the Simultaneous Perturbation Stochastic Approximation (SPSA) gradient estimator[4]:

$$\left(\hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})\right)_k = \frac{\hat{J}_i(\theta_n^{(i)} + c_n\Delta_n) - \hat{J}_i(\theta_n^{(i)} - c_n\Delta_n)}{2c_n(\Delta_n)_k}$$

5. Next, we use the estimated gradient and update the vector of thresholds through the stochastic approximation update:

$$\theta_{n+1}^{(i)} = \theta_n^{(i)} + a_n \hat{\nabla}_{\theta_n^{(i)}} J_i(\theta_n^{(i)})$$

[4] James C. Spall. "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation". In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* 37.3 (1992), pp. 332–341.

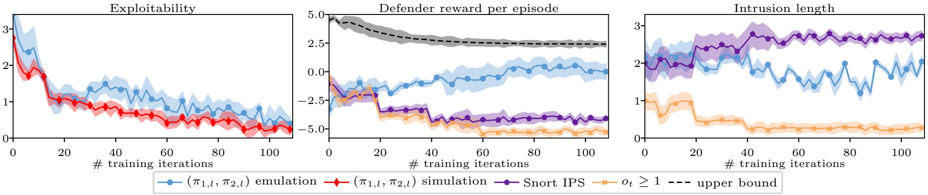
Outline

- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: T-FP
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Outline

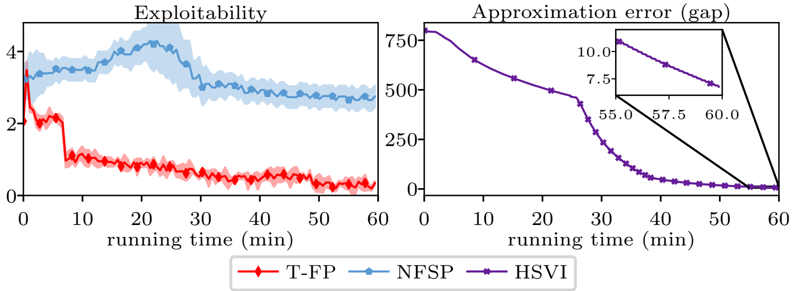
- **Use Case & Approach:**
  - Use case: Intrusion prevention
  - Approach: Emulation, simulation, and reinforcement learning
- **Game-Theoretic Model of The Use Case**
  - Intrusion prevention as an optimal stopping problem
  - Partially observed stochastic game
- **Game Analysis and Structure of $(\tilde{\pi}_1, \tilde{\pi}_2)$**
  - Existence of Nash Equilibria
  - Structural result: multi-threshold best responses
- **Our Method for Learning Equilibrium Strategies**
  - Our method for emulating the target infrastructure
  - Our system identification algorithm
  - Our reinforcement learning algorithm: $\mathrm{T\text{-}FP}$
- **Results & Conclusion**
  - Numerical evaluation results, conclusion, and future work

Evaluation Results: Learning Nash Equilibrium Strategies



Learning curves from the self-play process with $\mathrm{T\text{-}FP}$; the red curve show simulation results and the blue curves show emulation results; the purple, orange, and black curves relate to baseline strategies; the curves indicate the mean and the 95% confidence interval over four training runs with different random seeds.

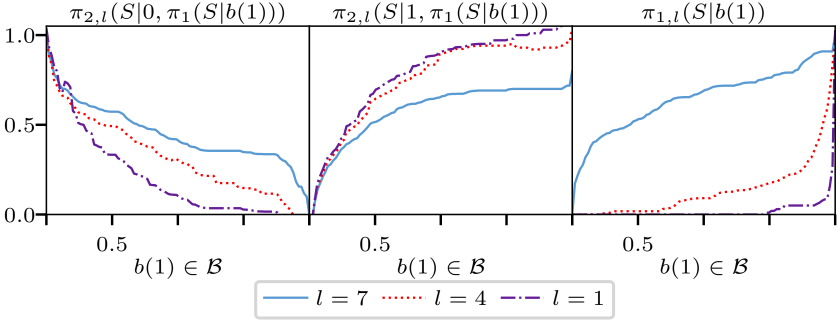
Evaluation Results: Converge Rates



Comparison between T-FP and two baseline algorithms: NFSP and HSVI; the red curve relate to T-FP; the blue curve to NFSP; the purple curve to HSVI; the left plot shows the approximate exploitability metric and the right plot shows the HSVI approximation error[5].

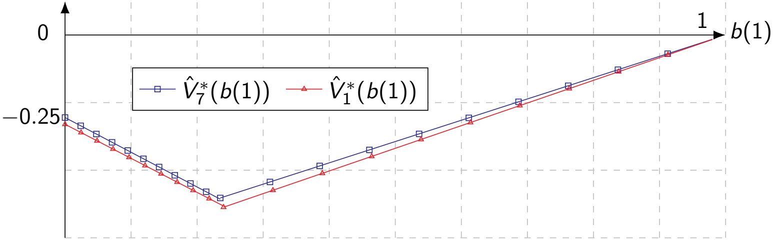
[5] Karel Horák, Branislav Bošanský, and Michal Pěchouček. "Heuristic Search Value Iteration for One-Sided Partially Observable Stochastic Games". In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2017). URL: https://ojs.aaai.org/index.php/AAAI/article/view/10597.

Evaluation Results: Inspection of Learned Strategies



Probability of the stop action S by the learned equilibrium strategies in function of $b(1)$ and l; the left and middle plots show the attacker's stopping probability when $s = 0$ and $s = 1$, respectively; the right plot shows the defender's stopping probability.

Evaluation Results: Inspection of Learned Game Values



The value function $\hat{V^*}_l(b(1))$ computed through the HSVI algorithm with approximation error 4; the blue and red curves relate to $l = 7$ and $l = 1$, respectively.

Conclusions & Future Work

- **Conclusions:**
  - We develop a *method* to automatically learn security strategies
    - (1) emulation system; (2) system identification; (3) simulation system; and (4) reinforcement learning.
  - We apply the method to an **intrusion prevention use case**
  - We formulate intrusion prevention as a stopping game
    - We present a Partially Observed Stochastic Game of the use case
    - We present a POMDP model of the defender's problem
    - We present a MDP model of the attacker's problem
    - We apply the stopping theory to establish structural results of the best response strategies and existence of Nash equilibria.
    - We show numerical results in realistic emulation environment
    - We show that our method outperforms two state-of-the-art methods
- **Our research plans:**
  - Extend the model (remove limiting assumptions)
    - Less restrictions on defender