Learning Automated Intrusion Response

Ericsson Research

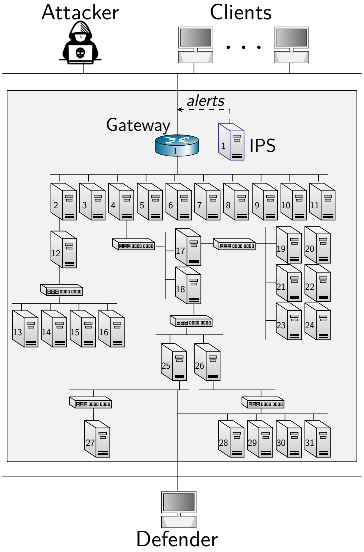
Kim Hammar & Rolf Stadler

kimham@kth.se
Division of Network and Systems Engineering
KTH Royal Institute of Technology

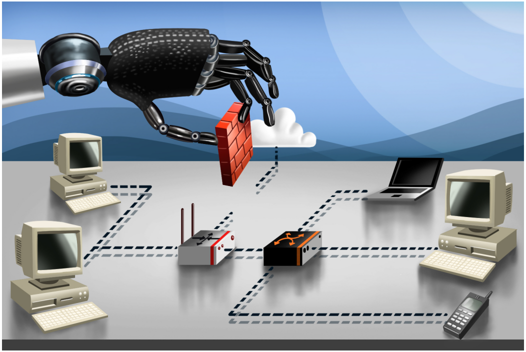
December 8, 2023

Use Case: Intrusion Response

- 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



Automated Intrusion Response



Levels of security automation

No automation.
Manual detection.
Manual prevention.
Lack of tools.

1980s

Operator assistance.
Audit logs
Manual detection.
Manual prevention.

1990s

Partial automation.
Manual configuration.
Intrusion detection systems.
Intrusion prevention systems.

2000s-Now

High automation.
System automatically
updates itself.

Research

Automated Intrusion Response

Can we find effective security strategies through decision-theoretic methods?

Levels of security automation

No automation. Manual detection. Manual prevention. Lack of tools.

1980s

Operator assistance. Audit logs Manual detection. Manual prevention.

1990s

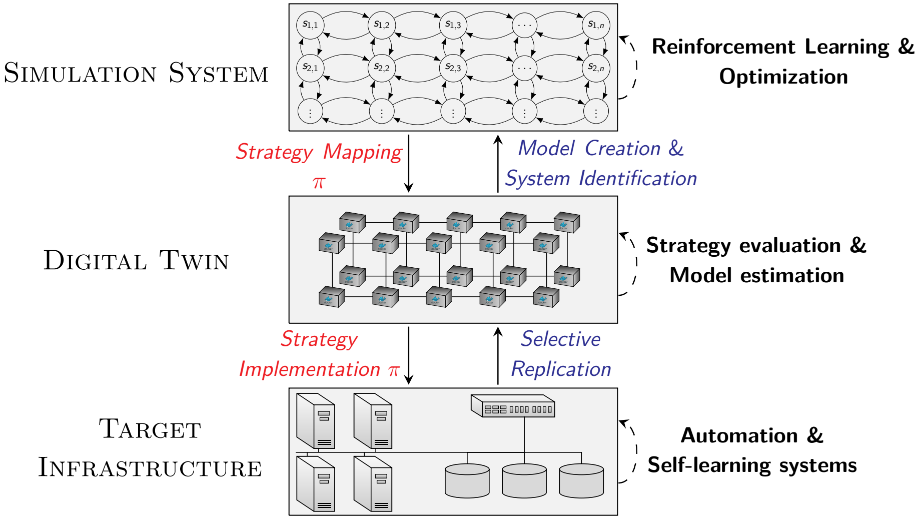
Partial automation. Manual configuration. Intrusion detection systems. Intrusion prevention systems.

2000s-Now

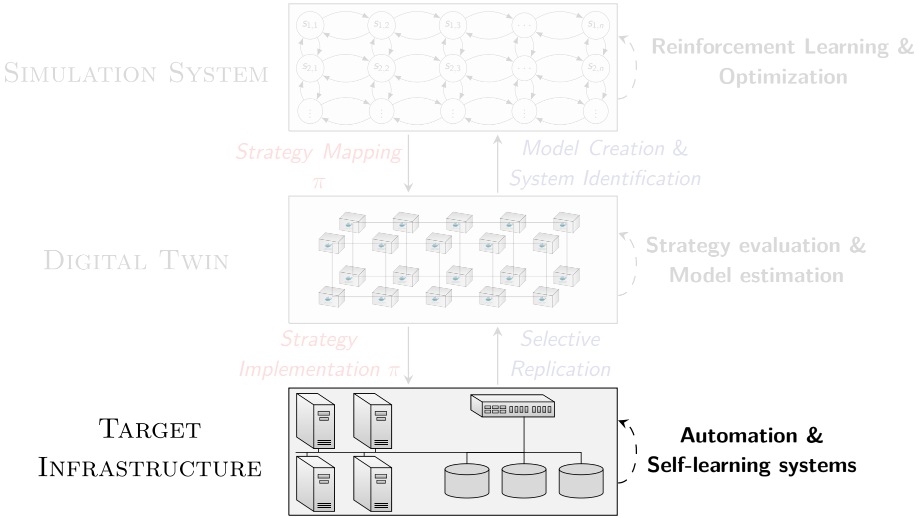
High automation. System automatically updates itself.

Research

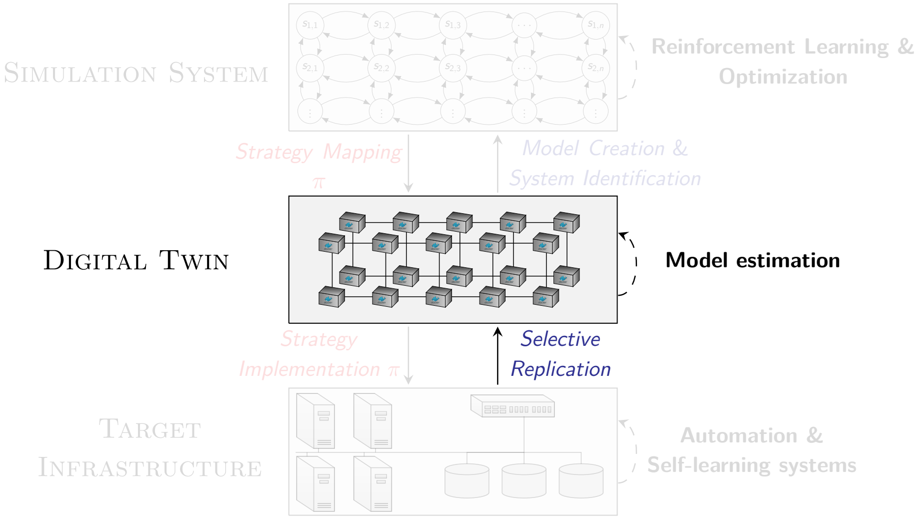
Our Framework for Automated Intrusion Response



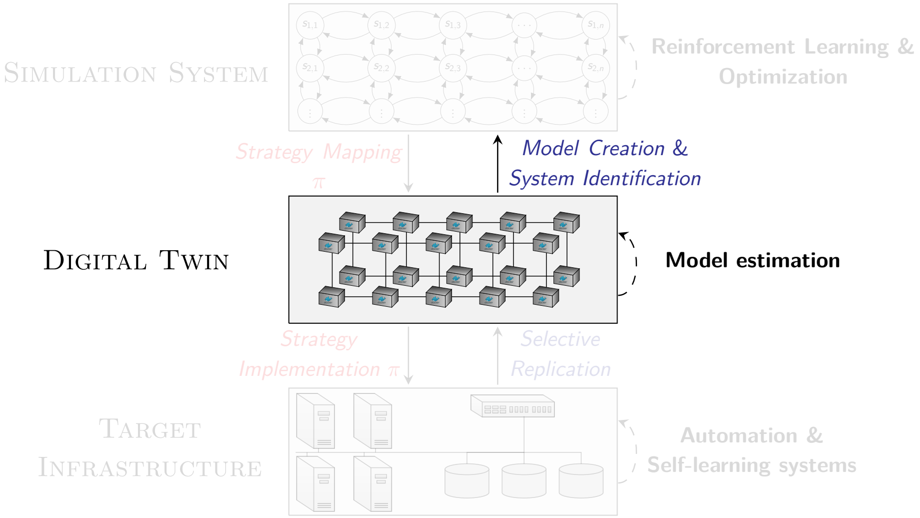
Our Framework for Automated Intrusion Response



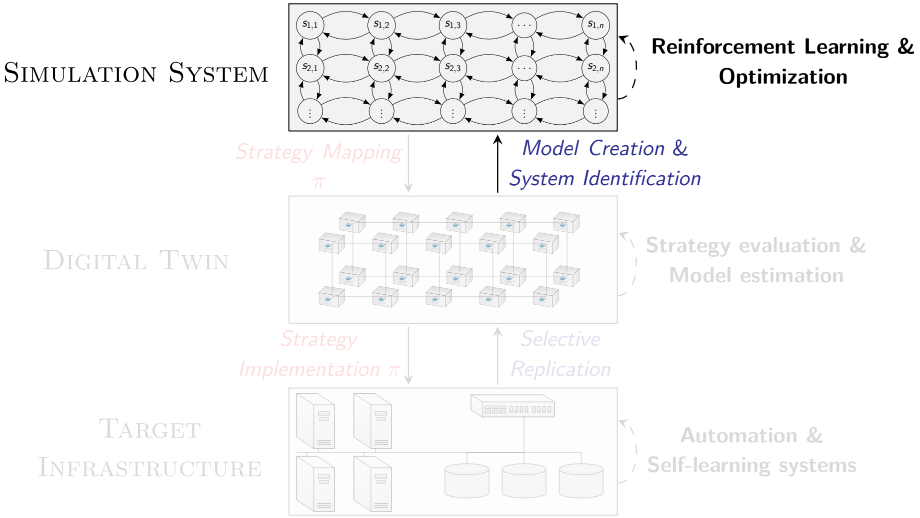
Our Framework for Automated Intrusion Response



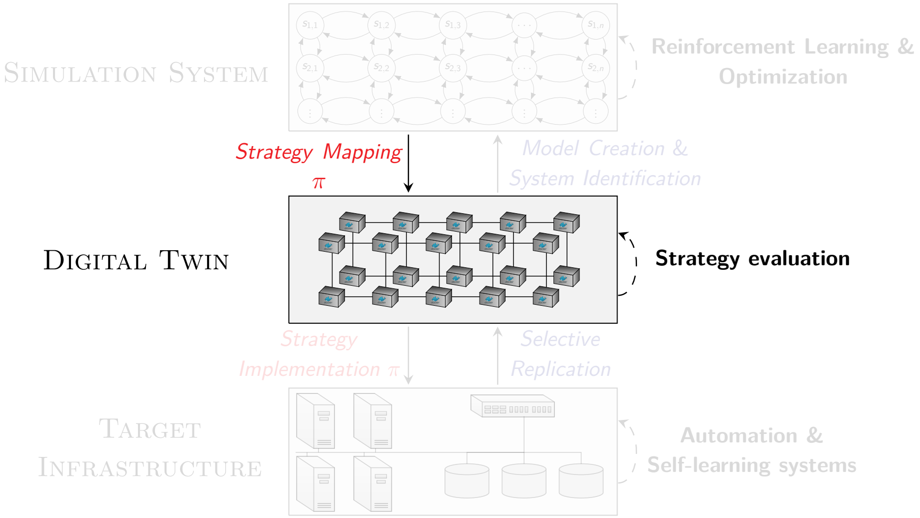
Our Framework for Automated Intrusion Response



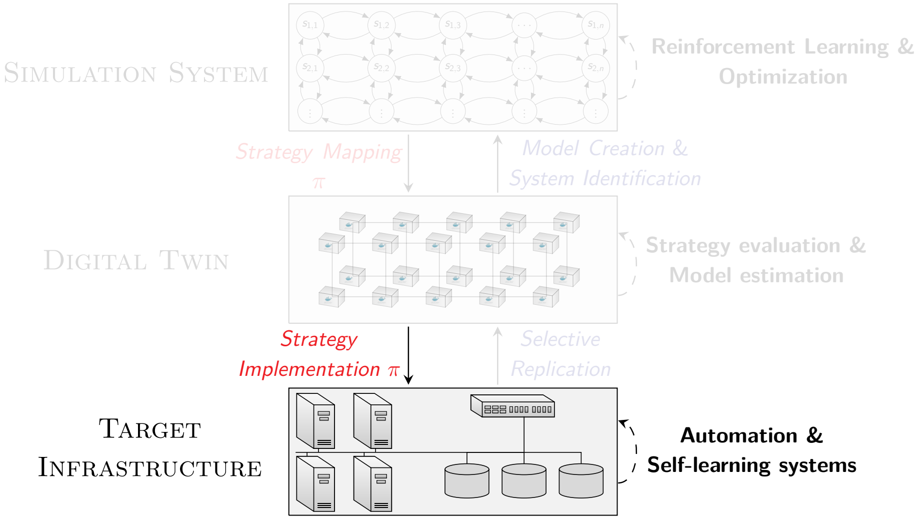
Our Framework for Automated Intrusion Response



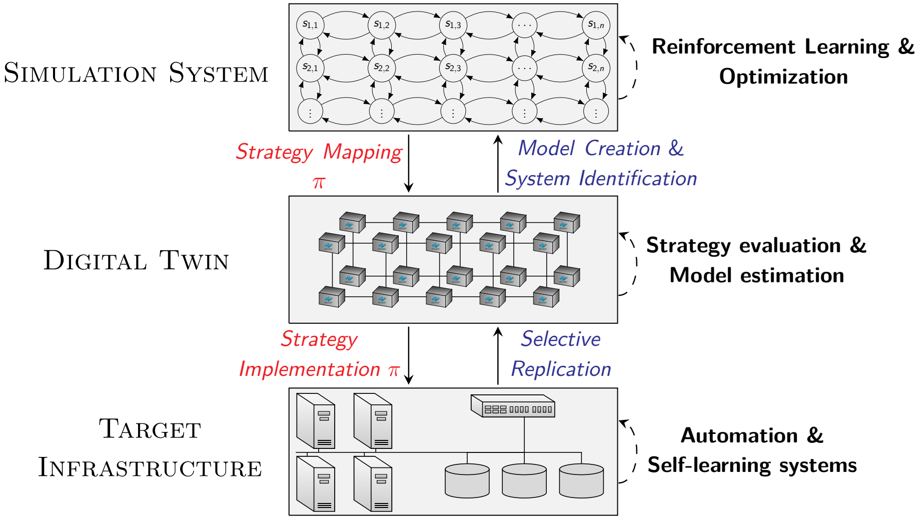
Our Framework for Automated Intrusion Response



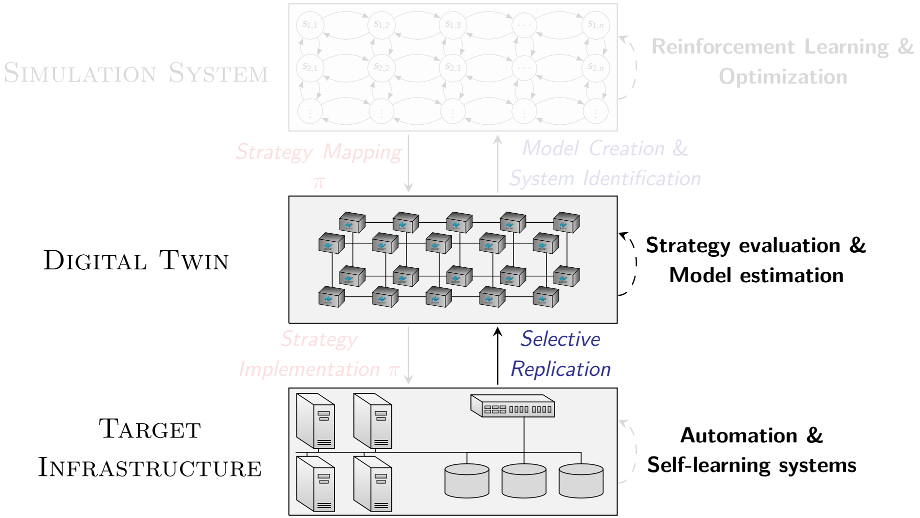
Our Framework for Automated Intrusion Response



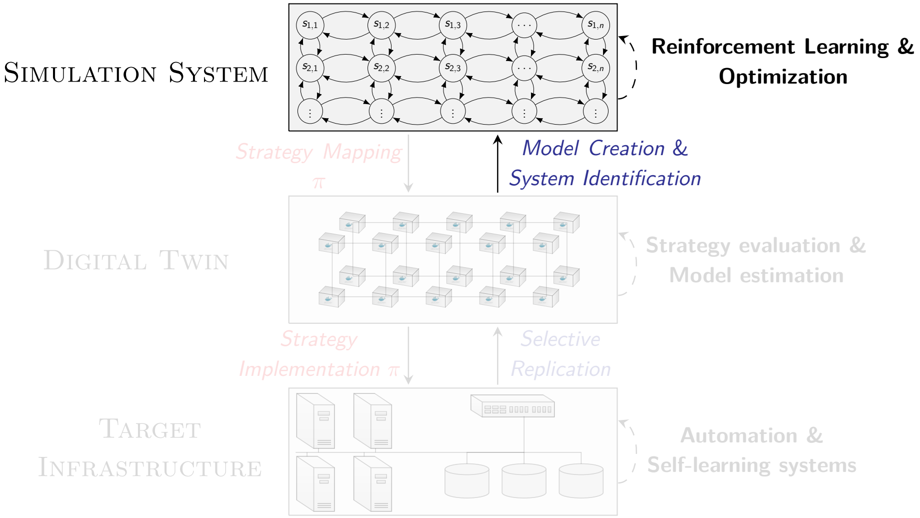
Our Framework for Automated Intrusion Response



Creating a Digital Twin of the Target Infrastructure

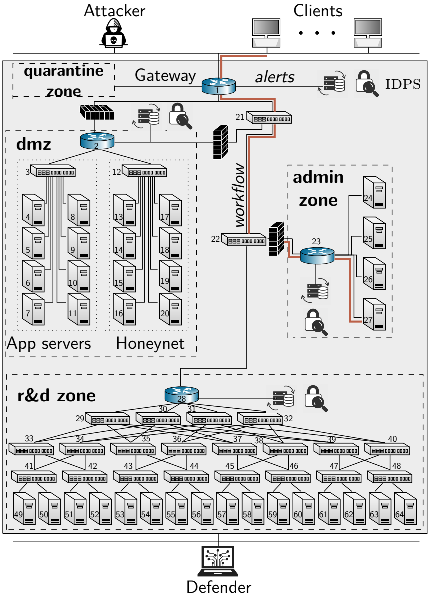


Learning of Defender Strategies

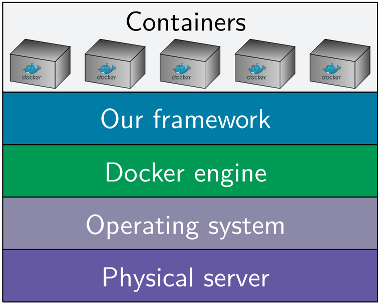


Example Infrastructure Configuration

- 64 **nodes**
  - 24 OVS switches
  - 3 gateways
  - 6 honeypots
  - 8 application servers
  - 4 administration servers
  - 15 compute servers
- 11 **vulnerabilities**
  - CVE-2010-0426
  - CVE-2015-3306
  - etc.
- **Management**
  - 1 SDN controller
  - 1 Kafka server
  - 1 elastic server

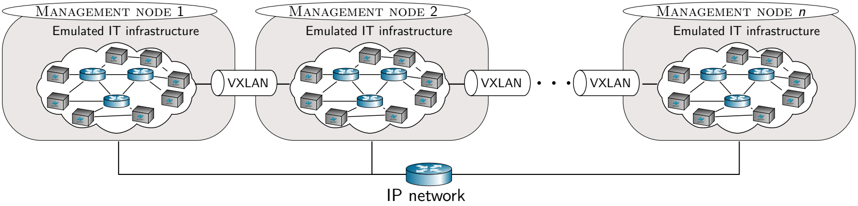


Emulating Physical Components



- We emulate physical components with **Docker containers**
- Focus on linux-based systems
- Our framework provides the orchestration layer

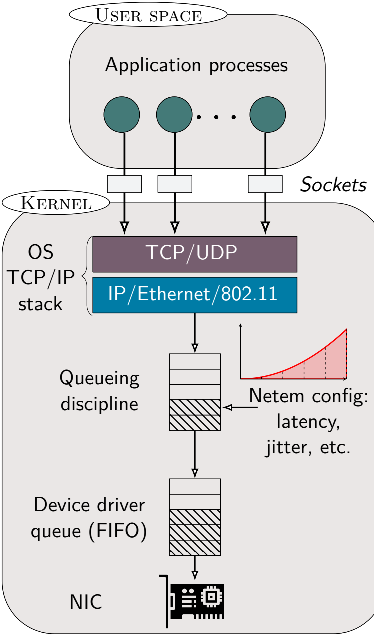
Emulating Network Connectivity



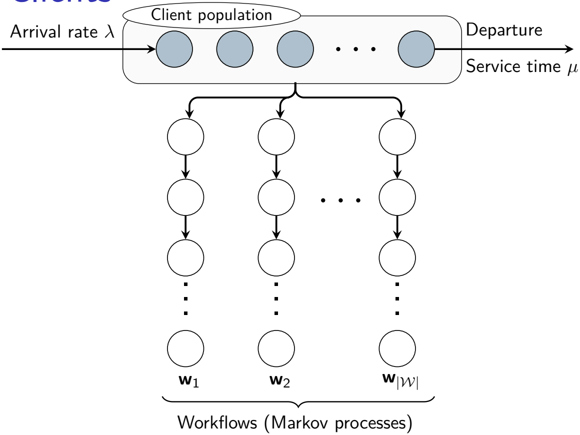
- We emulate network connectivity on the same host using **network namespaces**
- Connectivity across physical hosts is achieved using **VXLAN tunnels** with Docker swarm

Emulating Network Conditions

- Traffic shaping using NetEm
- Allows to configure:
  - Delay
  - Capacity
  - Packet Loss
  - Jitter
  - Queueing delays
  - etc.



Emulating Clients



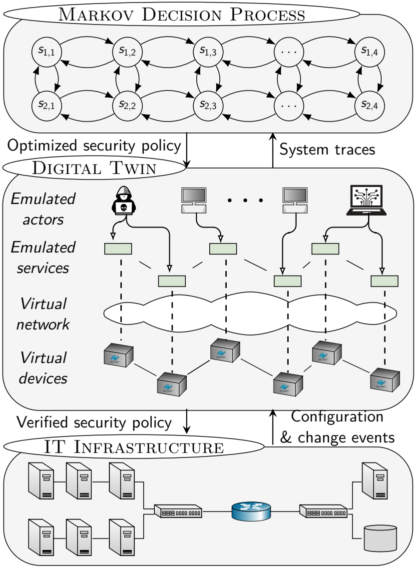
- Homogeneous client population
- Clients arrive according to $Po(\lambda)$
- Client service times $Exp(\mu)$
- Service dependencies $(S_t)_{t=1,2,\ldots} \sim \mathrm{MC}$

Emulating The Attacker and The Defender

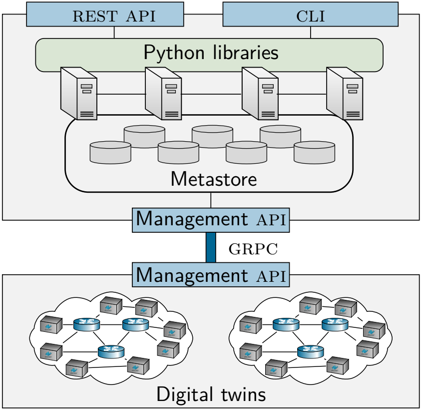
- **API for automated defender and attacker actions**

- **Attacker actions:**
  - Exploits
  - Reconnaissance
  - Pivoting
  - etc.

- **Defender actions:**
  - Shut downs
  - Redirect
  - Isolate
  - Recover
  - Migrate
  - etc.

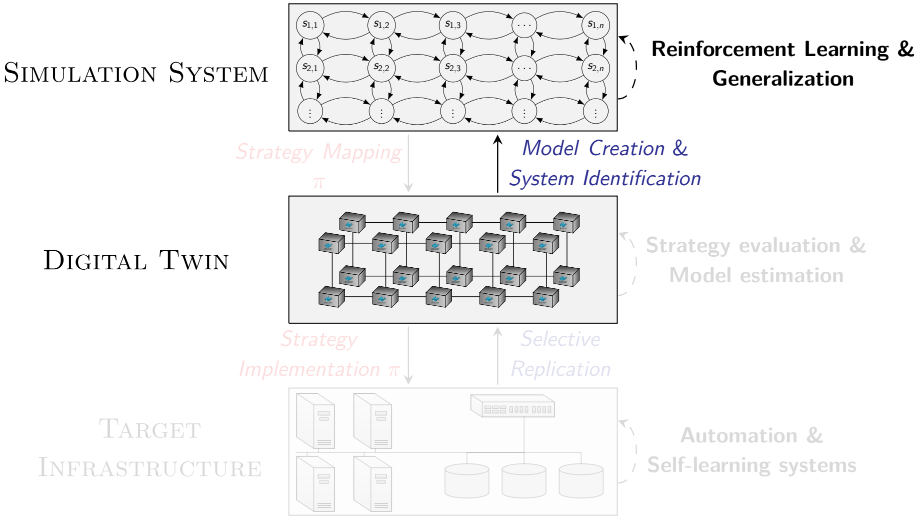


Software framework

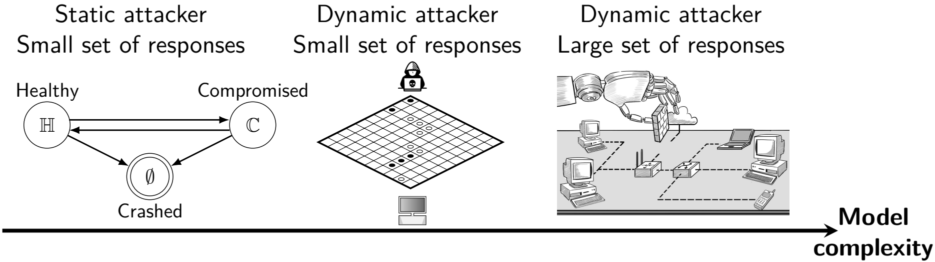


- More details about the software framework
  - Source code: https://github.com/Limmen/csle
  - Documentation: http://limmen.dev/csle/
  - Demo: https://www.youtube.com/watch?v=iE2KPmtIs2A

System Identification

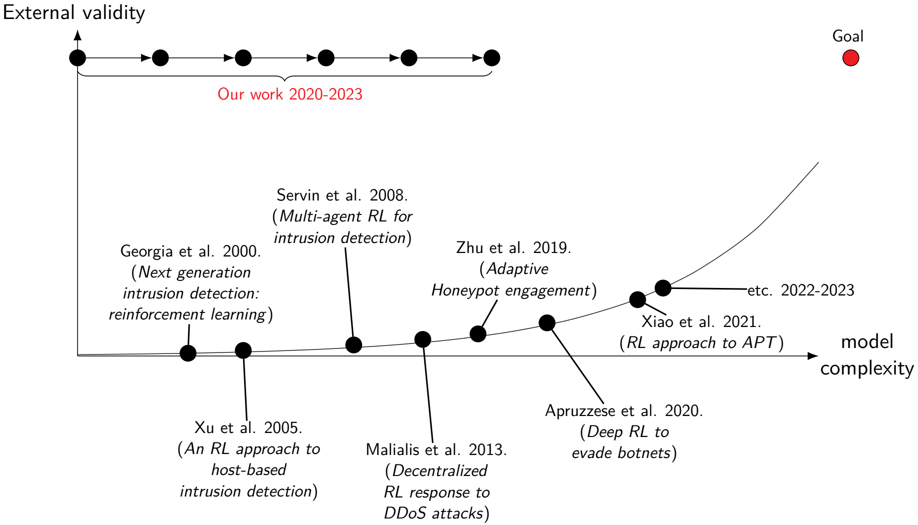


System Model



- Intrusion response can be **modeled in many ways**
  - As a *parametric optimization problem*
  - As an *optimal stopping problem*
  - As a *dynamic program*
  - As a *game*
  - etc.

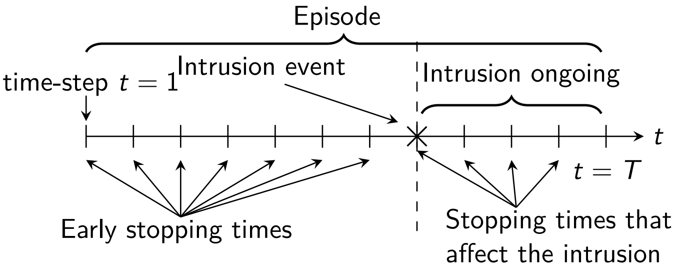
Related Work on Learning Automated Intrusion Response



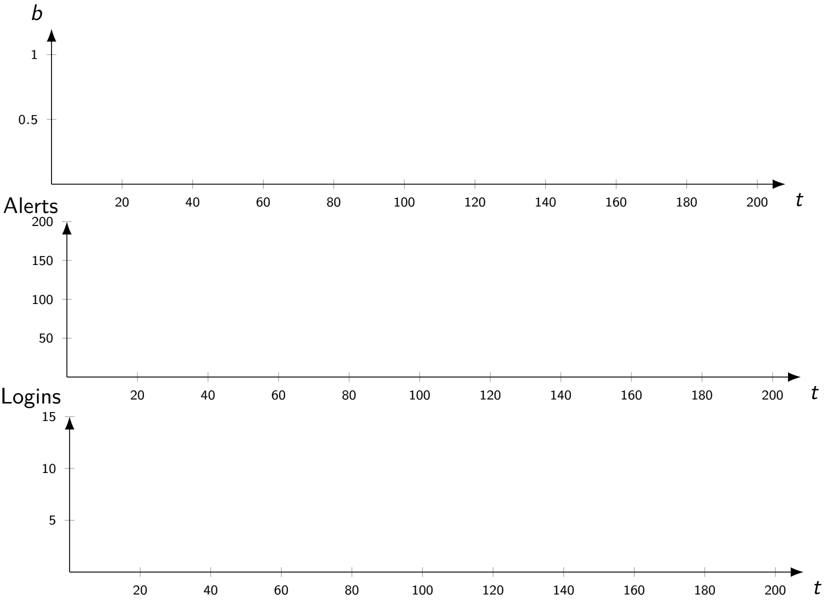
Intrusion Response through Optimal Stopping

- **Suppose**
  - The attacker follows a fixed strategy (no adaptation)
  - We only have one response action, e.g., block the gateway

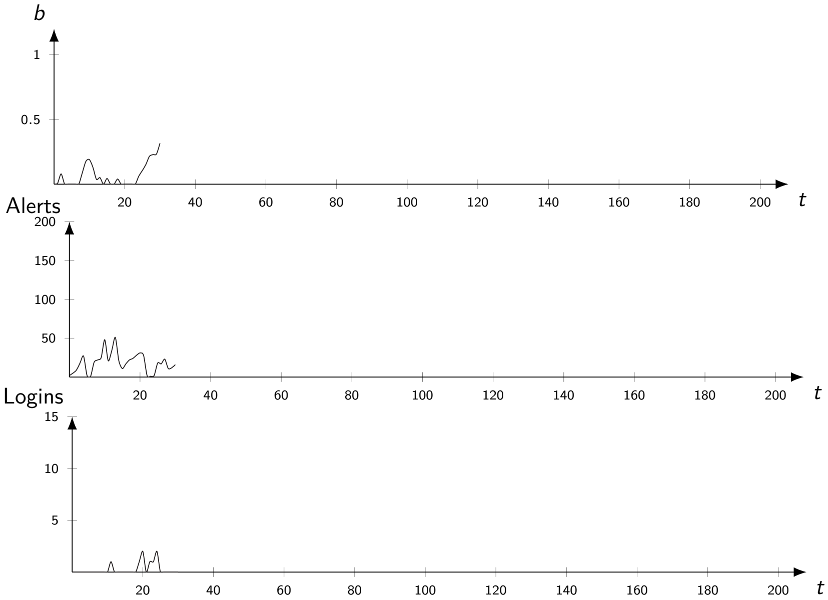
- Formulate intrusion response as **optimal stopping**



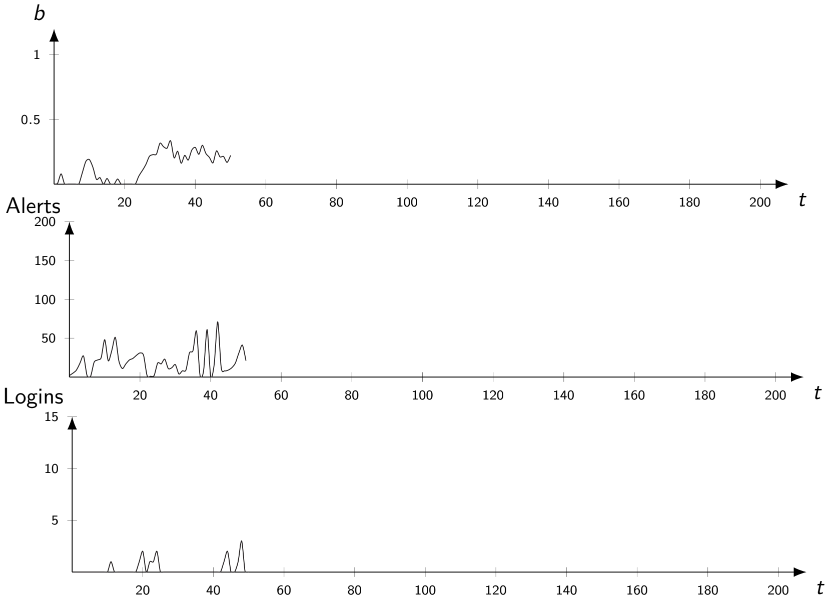
Intrusion Response from the Defender's Perspective



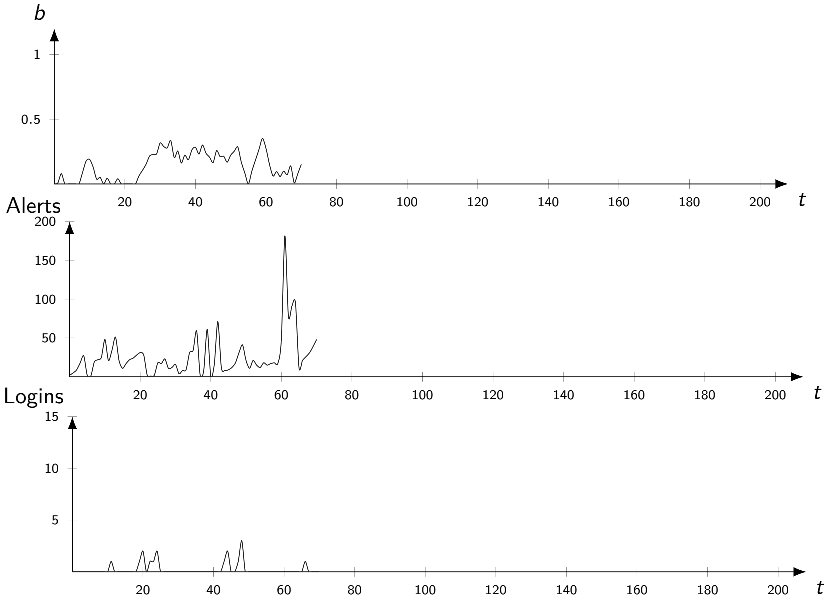
Intrusion Response from the Defender's Perspective



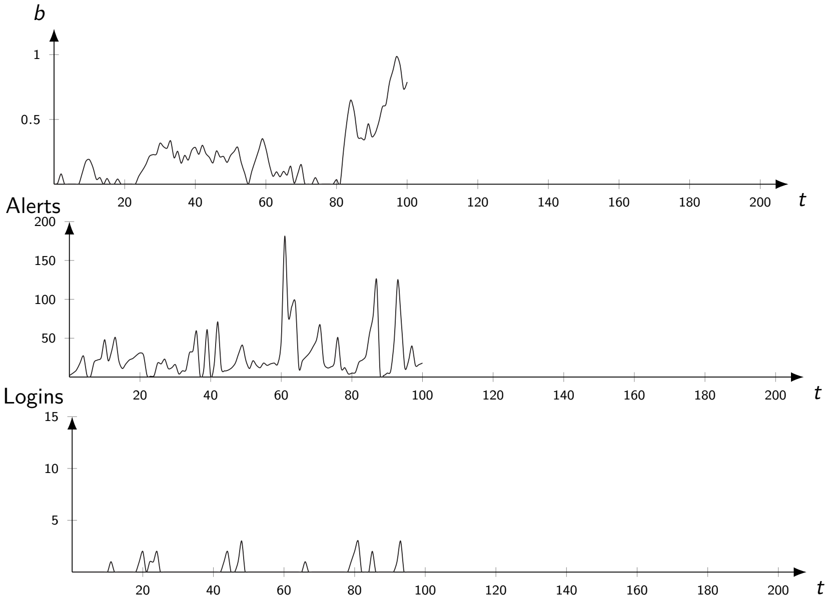
Intrusion Response from the Defender's Perspective



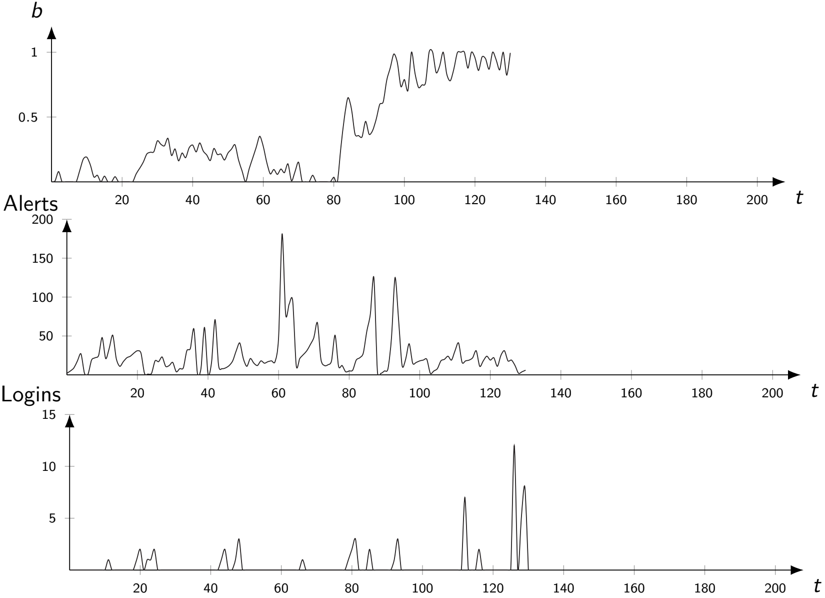
Intrusion Response from the Defender's Perspective



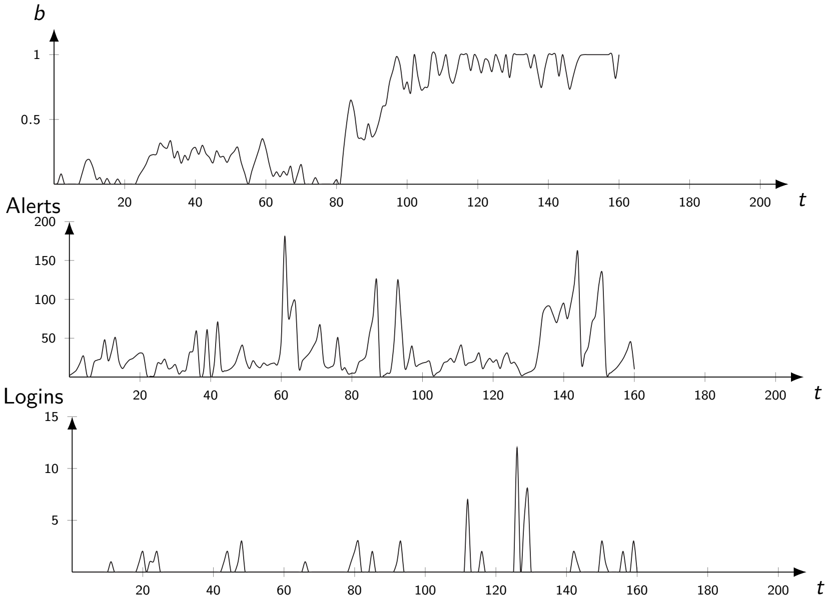
Intrusion Response from the Defender's Perspective



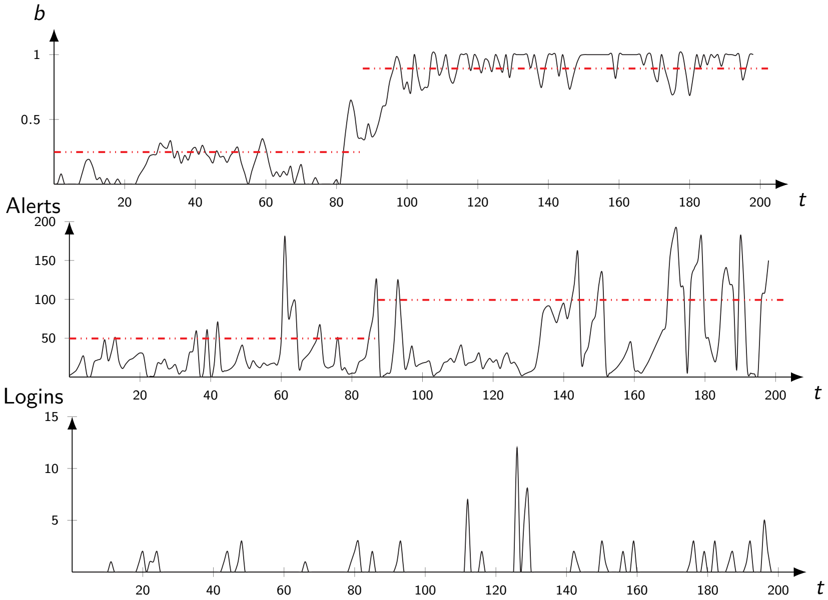
Intrusion Response from the Defender's Perspective



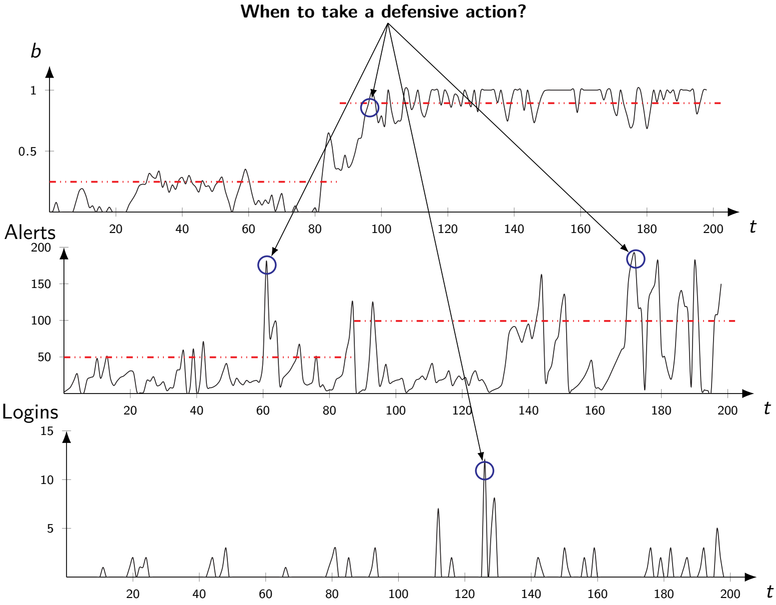
Intrusion Response from the Defender's Perspective



Intrusion Response from the Defender's Perspective



Intrusion Response from the Defender's Perspective



The Defender's Optimal Stopping Problem (1/3)

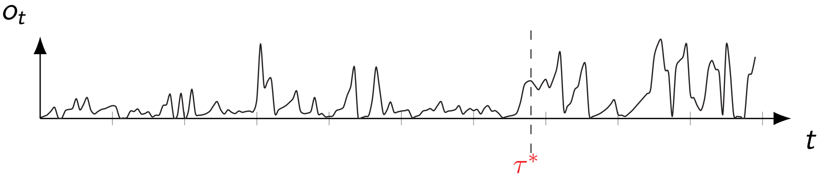
- Infrastructure is a **discrete-time dynamical system** $(s_t)_{t=1}^{T}$
- Defender observes a **noisy observation process** $(o_t)_{t=1}^{T}$
- Two options at each time t: $(\mathfrak{C})$ontinue and $(\mathfrak{S})$stop

- Find the *optimal stopping time* $\tau^\star$:

$$\tau^\star \in \arg\max_{\tau} \mathbb{E}_\tau \left[\sum_{t=1}^{\tau-1} \gamma^{t-1} \mathcal{R}^{\mathfrak{C}}_{s_t s_{t+1}} + \gamma^{\tau-1} \mathcal{R}^{\mathfrak{S}}_{s_\tau s_\tau} \right]$$

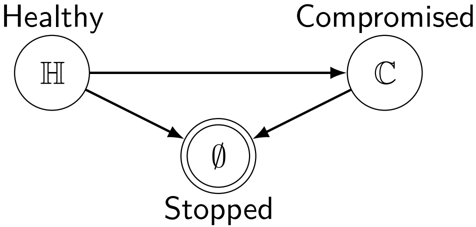
where $\mathcal{R}^{\mathfrak{S}}_{ss'}$ & $\mathcal{R}^{\mathfrak{C}}_{ss'}$ are the stop/continue rewards and τ is

$$\tau = \inf\{t : t > 0, a_t = \mathfrak{S}\}$$



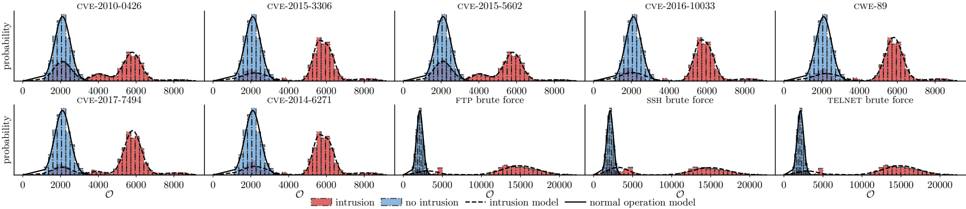
The Defender's Optimal Stopping Problem (2/3)

- **Objective:** stop the attack as soon as possible
- Let the **state space** be $\mathcal{S} = \{\mathbb{H}, \mathbb{C}, \emptyset\}$



The Defender's Optimal Stopping Problem (3/3)

- Let the **observation process** $(o_t)_{t=1}^{T}$ represent IDS alerts



- **Estimate the observation distribution** based on M samples from the twin
- E.g., compute empirical distribution $\widehat{Z}$ as estimate of Z
- $\widehat{Z} \rightarrow^{\text{a.s}} Z$ as $M \rightarrow \infty$ (Glivenko-Cantelli theorem)

Optimal Stopping Strategy

- The defender can compute the **belief**

$$b_t \triangleq \mathbb{P}[S_{i,t} = \mathbb{C} \mid b_1, o_1, o_2, \ldots o_t]$$

- **Stopping strategy**: $\pi(b) : [0,1] \rightarrow \{\mathfrak{S}, \mathfrak{C}\}$

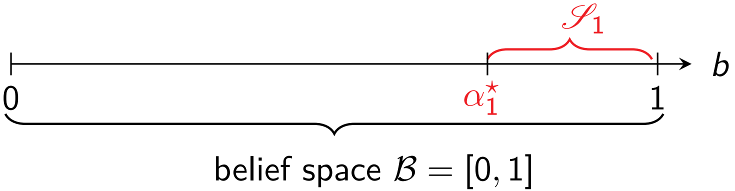
Optimal Threshold Strategy

Theorem

There exists an optimal defender strategy of the form:

$$\pi^{\star}(b) = \mathfrak{S} \iff b \geq \alpha^{\star} \qquad \alpha^{\star} \in [0,1]$$

i.e., the stopping set is $\mathscr{S} = [\alpha^{\star}, 1]$



Optimal **Multiple** Stopping

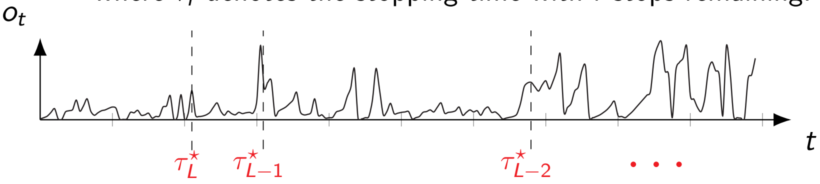
- Suppose the defender can take $L \geq 1$ **response actions**
- Find the *optimal stopping times* $\tau_L^\star, \tau_{L-1}^\star, \ldots, \tau_1^\star$:

$$(\tau_l^\star)_{l=1,\ldots,L} \in \underset{\tau_1,\ldots,\tau_L}{\arg\max}\, \mathbb{E}_{\tau_1,\ldots,\tau_L}\Bigg[\sum_{t=1}^{\tau_L-1} \gamma^{t-1}\mathcal{R}^{\mathfrak{C}}_{s_t s_{t+1}} + \gamma^{\tau_L-1}\mathcal{R}^{\mathfrak{S}}_{s_{\tau_L} s_{\tau_L}} +$$

$$\sum_{t=\tau_L+1}^{\tau_{L-1}-1} \gamma^{t-1}\mathcal{R}^{\mathfrak{C}}_{s_t s_{t+1}} + \gamma^{\tau_{L-1}-1}\mathcal{R}^{\mathfrak{S}}_{s_{\tau_{L-1}} s_{\tau_{L-2}}} + \ldots +$$

$$\sum_{t=\tau_2+1}^{\tau_1-1} \gamma^{t-1}\mathcal{R}^{\mathfrak{C}}_{s_t s_{t+1}} + \gamma^{\tau_1-1}\mathcal{R}^{\mathfrak{S}}_{s_{\tau_1} s_{\tau_1}}\Bigg]$$

where τ_l denotes the stopping time with l stops remaining.



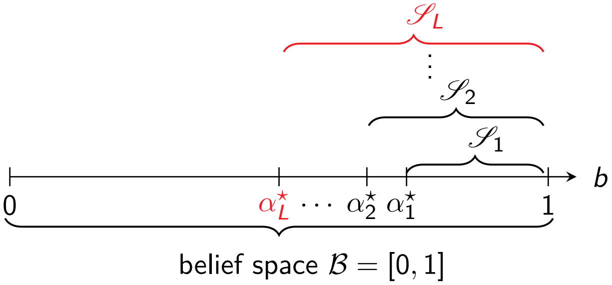
Optimal **Multi**-Threshold Strategy

Theorem

- *Stopping sets are nested* $\mathscr{S}_{l-1} \subseteq \mathscr{S}_l$ *for* $l = 2, \dots L$.
- *If* $(o_t)_{t \geq 1}$ *is totally positive of order 2 (TP2), there exists an optimal defender strategy of the form:*

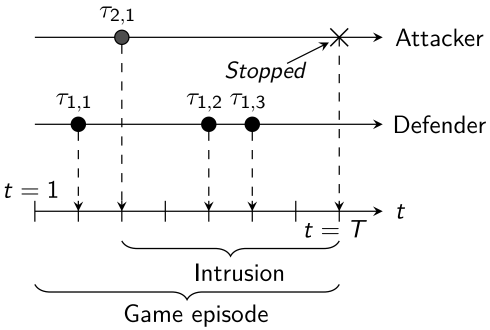
$$\pi_l^\star(b) = \mathfrak{S} \iff b \geq \alpha_l^\star, \qquad l = 1, \dots, L$$

where $\alpha_l^\star \in [0, 1]$ *is decreasing in* l.



Optimal Stopping **Game**

- Suppose the attacker is dynamic and **decides when to start and abort** its intrusion.



- Find the *optimal stopping times*

$$\underset{\tau_{D,1},\ldots,\tau_{D,L}}{\text{maximize}}\ \underset{\tau_{A,1},\tau_{A,2}}{\text{minimize}}\ \mathbb{E}[J]$$

where J is the defender's objective.

Best-Response Multi-Threshold Strategies (1/2)

Theorem

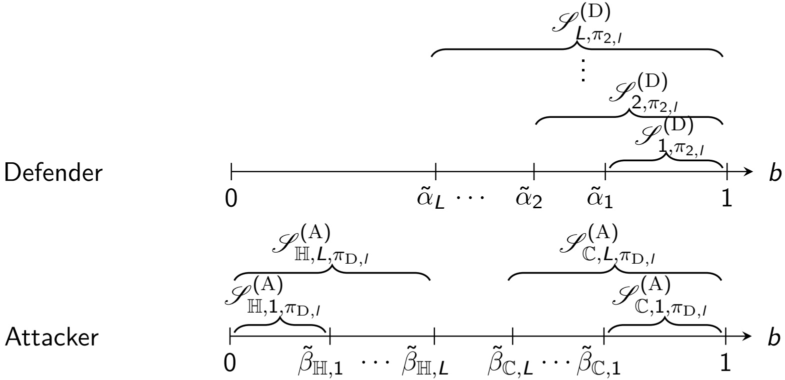
- *The defender's best response is of the form:*

$$\tilde{\pi}_{\mathrm{D},l}(b) = \mathfrak{S} \iff b \geq \tilde{\alpha}_l, \qquad l = 1, \ldots, L$$

- *The attacker's best response is of the form:*

$$\tilde{\pi}_{\mathrm{A},l}(b) = \mathfrak{C} \iff \tilde{\pi}_{\mathrm{D},l}(\mathfrak{S} \mid b) \geq \tilde{\beta}_{\mathbb{H},l}, \quad l = 1, \ldots, L, s = \mathbb{H}$$
$$\tilde{\pi}_{\mathrm{A},l}(b) = \mathfrak{S} \iff \tilde{\pi}_{\mathrm{D},l}(\mathfrak{S} \mid b) \geq \tilde{\beta}_{\mathbb{C},l}, \quad l = 1, \ldots, L, s = \mathbb{C}$$

Best-Response Multi-Threshold Strategies (2/2)



Efficient Computation of Best Responses

Algorithm 1: Threshold Optimization

Input: Objective function J, number of thresholds L, parametric optimizer PO

Output: A approximate best response strategy $\hat{\pi}_\theta$

Algorithm

$\Theta \leftarrow [0, 1]^L$

For each $\theta \in \Theta$, define $\pi_\theta(b_t)$ as

$$\pi_\theta(b_t) \triangleq \begin{cases} \mathfrak{S} & \text{if } b_t \geq \theta_i \\ \mathfrak{C} & \text{otherwise} \end{cases}$$

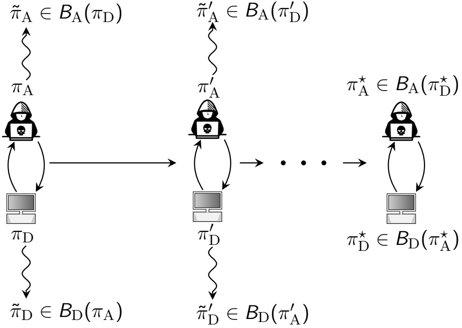
$J_\theta \leftarrow \mathbb{E}_{\pi_\theta}[J]$

$\hat{\pi}_\theta \leftarrow \mathrm{PO}(\Theta, J_\theta)$

return $\hat{\pi}_\theta$

- Examples of parameteric optimization algorithmns: CEM, BO, CMA-ES, DE, SPSA, etc.

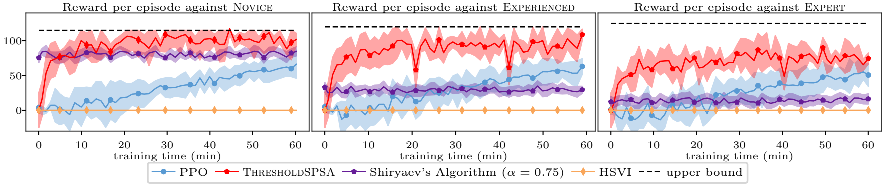
Threshold-Fictitious Play to Approximate an Equilibrium

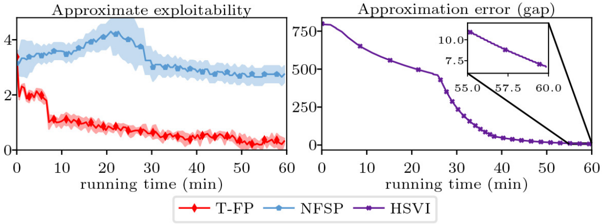


Fictitious play: iterative averaging of best responses.

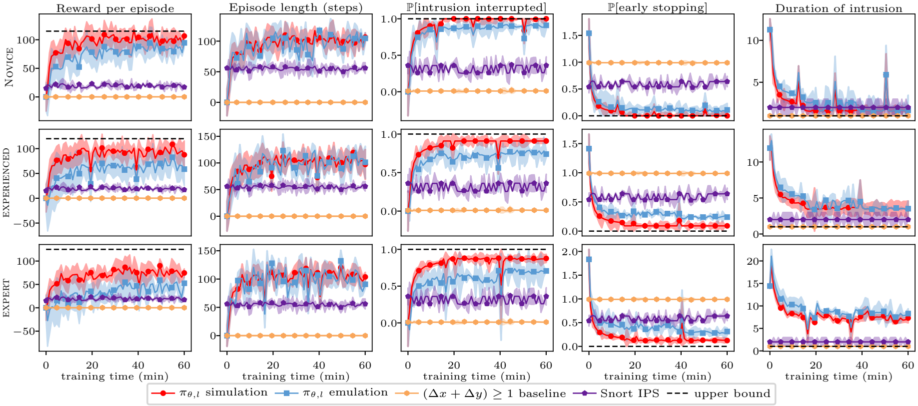
- **Learn best response** strategies iteratively
- Average best responses to **approximate the equilibrium**

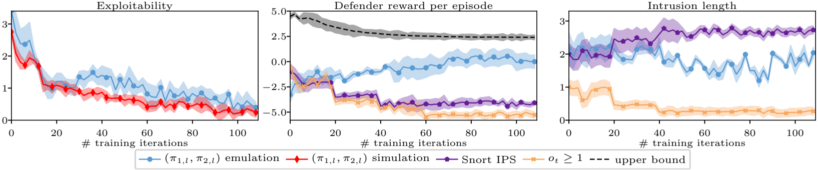
Comparison against State-of-the-art Algorithms



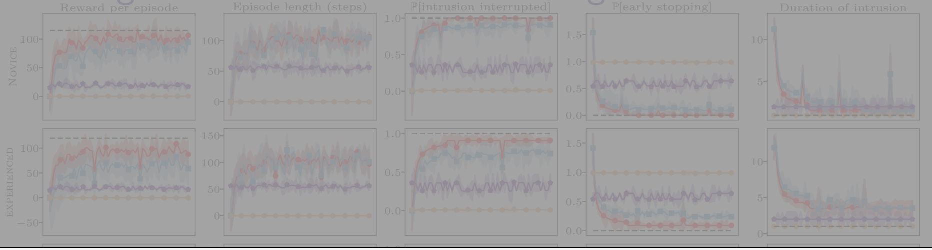


Learning Curves in Simulation and Digital Twin

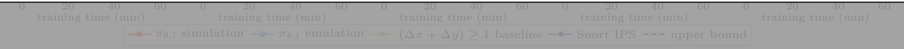


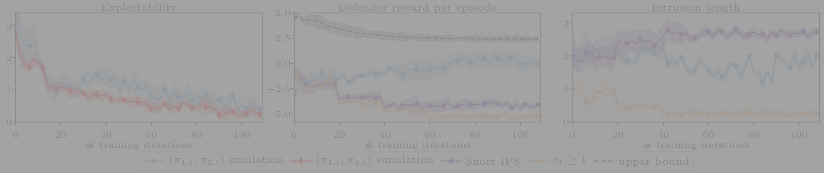


Learning Curves in Simulation and Digital Twin



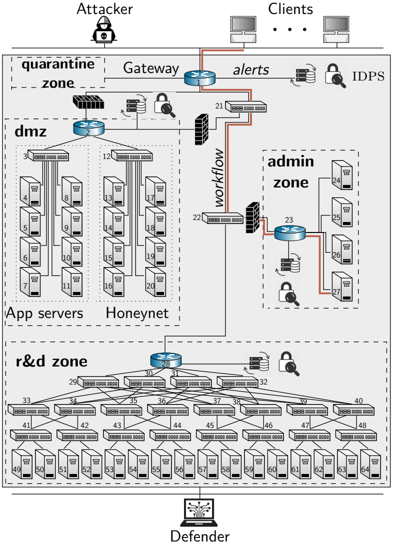
Stopping is about **timing**; now we consider timing + action selection





General Intrusion Response Game

- Suppose the defender and the attacker can take L actions **per node**
- $\mathcal{G} = \langle \{\text{gw}\} \cup \mathcal{V}, \mathcal{E} \rangle$: directed tree representing the virtual infrastructure
- $\mathcal{V}$: set of virtual nodes
- $\mathcal{E}$: set of node dependencies
- $\mathcal{Z}$: set of zones



General Intrusion Response Game

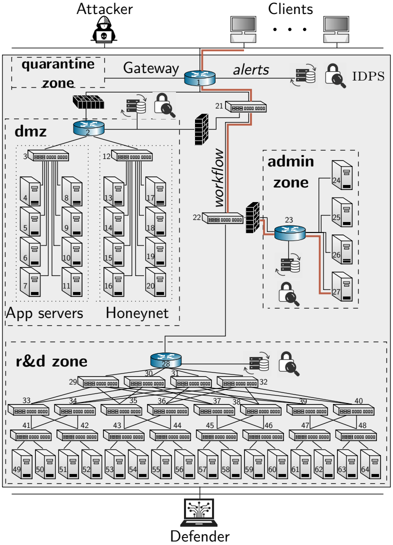
- Suppose the defender and the attacker can take L actions **per node**

- $\mathcal{G} = \langle \{\text{gw}\} \cup \mathcal{V}, \mathcal{E} \rangle$: directed tree representing the virtual infrastructure

- $\mathcal{V}$: set of virtual nodes

- $\mathcal{E}$: set of node dependencies

- $\mathcal{Z}$: set of zones



State Space

- Each $i \in \mathcal{V}$ has a state

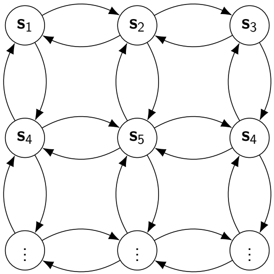
$$\boldsymbol{v}_{i,t} = (\underbrace{v_{t,i}^{(\mathrm{Z})}}_{\mathrm{D}}, \underbrace{v_{t,i}^{(\mathrm{I})}, v_{t,i}^{(\mathrm{R})}}_{\mathrm{A}})$$

- System state $\mathbf{s}_t = (\mathbf{v}_{t,i})_{i \in \mathcal{V}} \sim \mathbf{S}_t$

- Markovian time-homogeneous dynamics:

$$\mathbf{s}_{t+1} \sim f(\cdot \mid \mathbf{S}_t, \mathbf{A}_t)$$

$\mathbf{A}_t = (\mathbf{A}_t^{(\mathrm{A})}, \mathbf{A}_t^{(\mathrm{D})})$ are the actions.



State Space

- Each $i \in \mathcal{V}$ has a state

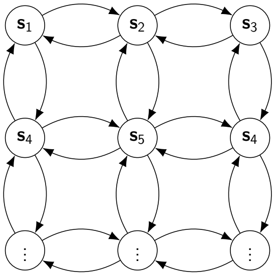
$$\boldsymbol{v}_{i,t} = (\underbrace{v_{t,i}^{(\mathrm{Z})}}_{\mathrm{D}}, \underbrace{v_{t,i}^{(\mathrm{I})}, v_{t,i}^{(\mathrm{R})}}_{\mathrm{A}})$$

- System state $\mathbf{s}_t = (\mathbf{v}_{t,i})_{i \in \mathcal{V}} \sim \mathbf{S}_t$

- Markovian time-homogeneous dynamics:

$$\mathbf{s}_{t+1} \sim f(\cdot \mid \mathbf{S}_t, \mathbf{A}_t)$$

$\mathbf{A}_t = (\mathbf{A}_t^{(\mathrm{A})}, \mathbf{A}_t^{(\mathrm{D})})$ are the actions.



State Space

- Each $i \in \mathcal{V}$ has a state

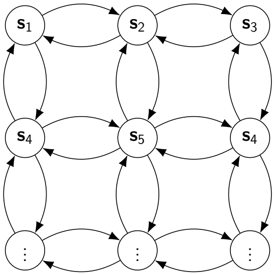
$$\boldsymbol{v}_{i,t} = (\underbrace{v_{t,i}^{(\mathrm{Z})}}_{\mathrm{D}}, \underbrace{v_{t,i}^{(\mathrm{I})}, v_{t,i}^{(\mathrm{R})}}_{\mathrm{A}})$$

- System state $\mathbf{s}_t = (\mathbf{v}_{t,i})_{i \in \mathcal{V}} \sim \mathbf{S}_t$

- Markovian time-homogeneous dynamics:

$$\mathbf{s}_{t+1} \sim f(\cdot \mid \mathbf{S}_t, \mathbf{A}_t)$$

$\mathbf{A}_t = (\mathbf{A}_t^{(\mathrm{A})}, \mathbf{A}_t^{(\mathrm{D})})$ are the actions.

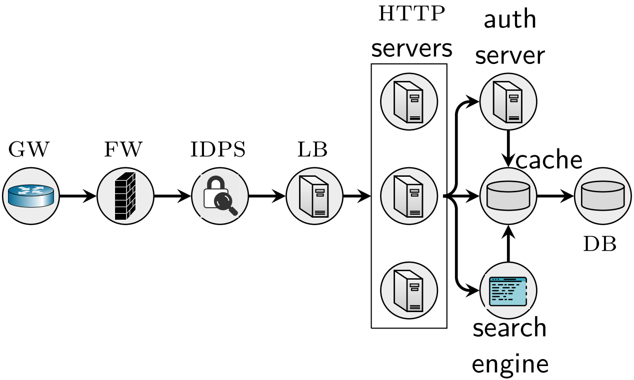


Workflows

- Services are connected into **workflows** $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{W}|}\}$.

Workflows

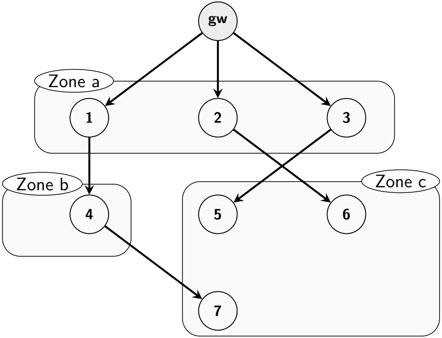
- Services are connected into **workflows** $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{W}|}\}$.



Workflows

- Services are connected into **workflows** $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{W}|}\}$.
- Each $\mathbf{w} \in \mathcal{W}$ is realized as a subtree $\mathcal{G}_{\mathbf{w}} = \langle \{\mathrm{gw}\} \cup \mathcal{V}_{\mathbf{w}}, \mathcal{E}_{\mathbf{w}} \rangle$ of $\mathcal{G}$
- $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{W}|}\}$ induces a partitioning

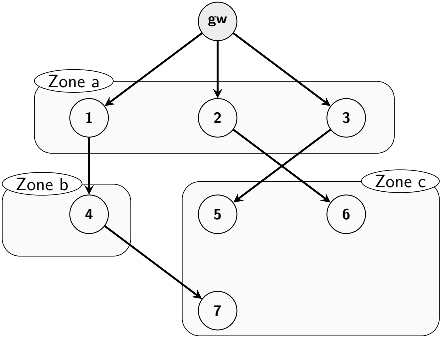
$$\mathcal{V} = \bigcup_{\mathbf{w}_i \in \mathcal{W}} \mathcal{V}_{\mathbf{w}_i} \text{ such that } i \neq j \implies \mathcal{V}_{\mathbf{w}_i} \cap \mathcal{V}_{\mathbf{w}_j} = \emptyset$$



A workflow tree

Workflows

- Services are connected into **workflows** $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{W}|}\}$.
- Each $\mathbf{w} \in \mathcal{W}$ is realized as a subtree $\mathcal{G}_{\mathbf{w}} = \langle \{\text{gw}\} \cup \mathcal{V}_{\mathbf{w}}, \mathcal{E}_{\mathbf{w}} \rangle$ of $\mathcal{G}$
- $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{W}|}\}$ induces a partitioning



A workflow tree

$$\mathcal{V} = \bigcup_{\mathbf{w}_i \in \mathcal{W}} \mathcal{V}_{\mathbf{w}_i} \text{ such that } i \neq j \implies \mathcal{V}_{\mathbf{w}_i} \cap \mathcal{V}_{\mathbf{w}_j} = \emptyset$$

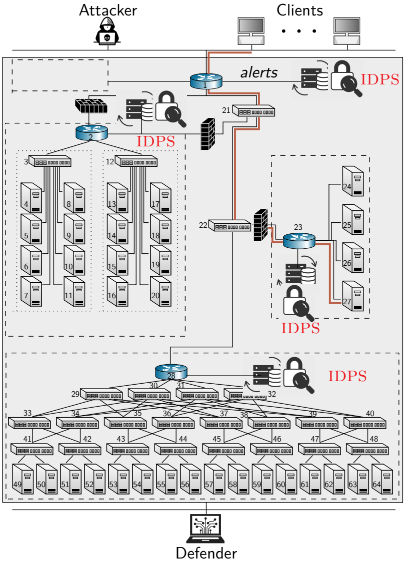
Observations

- IDPSs inspect network traffic and generate alert vectors:

$$\mathbf{o}_t \triangleq \left(\mathbf{o}_{t,1}, \ldots, \mathbf{o}_{t,|\mathcal{V}|}\right) \in \mathbb{N}_0^{|\mathcal{V}|}$$

  $\mathbf{o}_{t,i}$ is the number of alerts related to node $i \in \mathcal{V}$ at time-step t.

- $\mathbf{o}_t = \left(\mathbf{o}_{t,1}, \ldots, \mathbf{o}_{t,|\mathcal{V}|}\right)$ is a realization of the random vector $\mathbf{O}_t$ with joint distribution Z



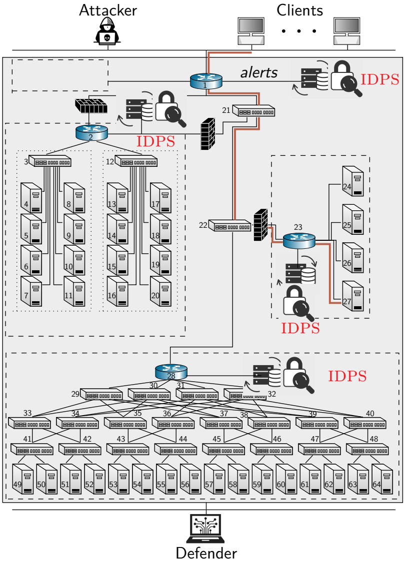
Observations

- IDPSs inspect network traffic and generate alert vectors:

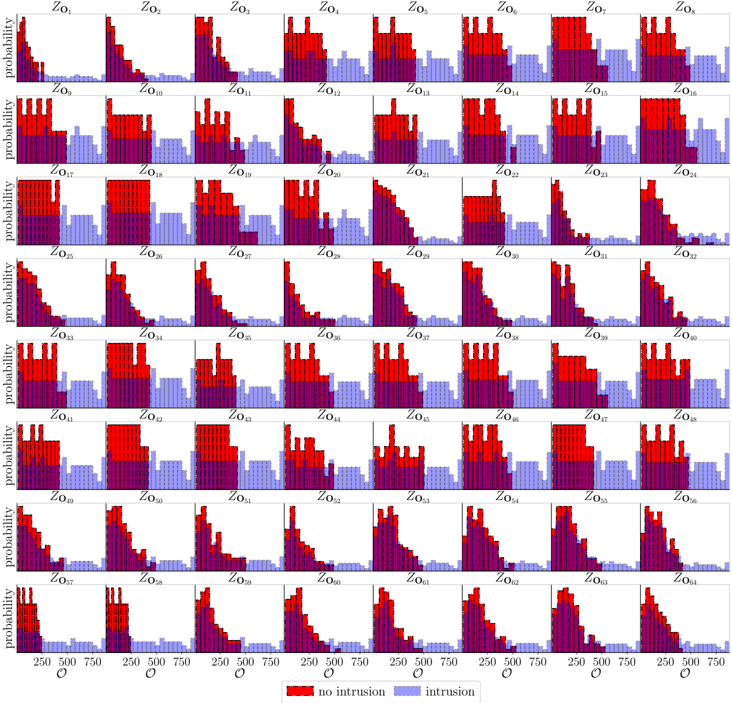
$$\mathbf{o}_t \triangleq \left(\mathbf{o}_{t,1}, \ldots, \mathbf{o}_{t,|\mathcal{V}|}\right) \in \mathbb{N}_0^{|\mathcal{V}|}$$

  $\mathbf{o}_{t,i}$ is the number of alerts related to node $i \in \mathcal{V}$ at time-step t.

- $\mathbf{o}_t = \left(\mathbf{o}_{t,1}, \ldots, \mathbf{o}_{t,|\mathcal{V}|}\right)$ is a realization of the random vector $\mathbf{O}_t$ with joint distribution Z

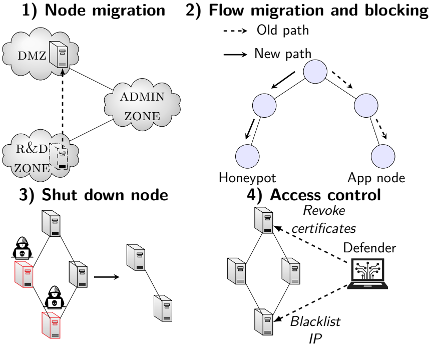


Distributions of # alerts weighted by priority $Z_{\mathbf{O}_i}(\mathbf{O}_i \mid \mathbf{S}_i^{(\mathrm{D})}, \mathbf{A}_i^{(\mathrm{A})})$ per node $i \in \mathcal{V}$



Defender

- Defender action:
  $\mathbf{a}_t^{(D)} \in \{0, 1, 2, 3, 4\}^{|\mathcal{V}|}$
- 0 means do nothing. $1 - 4$ correspond to defensive actions (see fig)



- A **defender strategy** is a function $\pi_{\mathrm{D}} \in \Pi_{\mathrm{D}} : \mathcal{H}_{\mathrm{D}} \to \Delta(\mathcal{A}_{\mathrm{D}})$, where

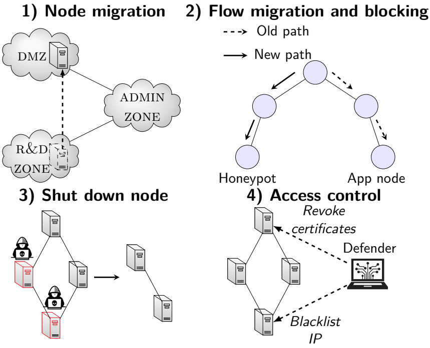
$$\mathbf{h}_t^{(\mathrm{D})} = (\mathbf{s}_1^{(\mathrm{D})}, \mathbf{a}_1^{(\mathrm{D})}, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}^{(\mathrm{D})}, \mathbf{s}_t^{(\mathrm{D})}, \mathbf{o}_t) \in \mathcal{H}_{\mathrm{D}}$$

- Objective: (i) maintain workflows; and (ii) stop a possible intrusion:

$$J \triangleq \sum_{t=1}^{T} \gamma^{t-1} \Bigg(\underbrace{\eta \sum_{i=1}^{|\mathcal{W}|} u_{\mathrm{W}}(\mathbf{w}_i, \mathbf{s}_t)}_{\text{workflows utility}} - \underbrace{(1-\eta) \sum_{j=1}^{|\mathcal{V}|} c_{\mathrm{I}}(\mathbf{s}_{t,j}, \mathbf{a}_{t,j})}_{\text{intrusion and defense costs}} \Bigg)$$

Defender

- Defender action:
  $\mathbf{a}_t^{(D)} \in \{0,1,2,3,4\}^{|\mathcal{V}|}$
- 0 means do nothing. 1 − 4 correspond to defensive actions (see fig)



- A **defender strategy** is a function $\pi_{\mathrm{D}} \in \Pi_{\mathrm{D}} : \mathcal{H}_{\mathrm{D}} \to \Delta(\mathcal{A}_{\mathrm{D}})$, where

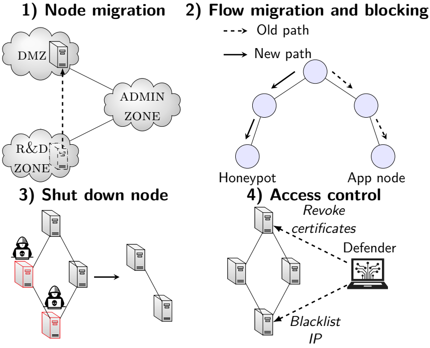
$$\mathbf{h}_t^{(\mathrm{D})} = (\mathbf{s}_1^{(\mathrm{D})}, \mathbf{a}_1^{(\mathrm{D})}, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}^{(\mathrm{D})}, \mathbf{s}_t^{(\mathrm{D})}, \mathbf{o}_t) \in \mathcal{H}_{\mathrm{D}}$$

- Objective: (*i*) maintain workflows; and (*ii*) stop a possible intrusion:

$$J \triangleq \sum_{t=1}^{T} \gamma^{t-1} \Bigg(\underbrace{\eta \sum_{i=1}^{|\mathcal{W}|} u_{\mathrm{W}}(\mathbf{w}_i, \mathbf{s}_t)}_{\text{workflows utility}} - \underbrace{(1-\eta) \sum_{j=1}^{|\mathcal{V}|} c_{\mathrm{I}}(\mathbf{s}_{t,j}, \mathbf{a}_{t,j})}_{\text{intrusion and defense costs}} \Bigg)$$

Defender

- Defender action:
  $\mathbf{a}_t^{(D)} \in \{0, 1, 2, 3, 4\}^{|\mathcal{V}|}$
- 0 means do nothing. $1 - 4$ correspond to defensive actions (see fig)



- A **defender strategy** is a function $\pi_{\mathrm{D}} \in \Pi_{\mathrm{D}} : \mathcal{H}_{\mathrm{D}} \to \Delta(\mathcal{A}_{\mathrm{D}})$, where

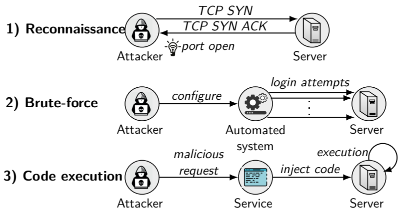
$$\mathbf{h}_t^{(\mathrm{D})} = (\mathbf{s}_1^{(\mathrm{D})}, \mathbf{a}_1^{(\mathrm{D})}, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}^{(\mathrm{D})}, \mathbf{s}_t^{(\mathrm{D})}, \mathbf{o}_t) \in \mathcal{H}_{\mathrm{D}}$$

- Objective: (*i*) maintain workflows; and (*ii*) stop a possible intrusion:

$$J \triangleq \sum_{t=1}^{T} \gamma^{t-1} \Bigg(\underbrace{\eta \sum_{i=1}^{|\mathcal{W}|} u_{\mathrm{W}}(\mathbf{w}_i, \mathbf{s}_t)}_{\text{workflows utility}} - \underbrace{(1-\eta) \sum_{j=1}^{|\mathcal{V}|} c_{\mathrm{I}}(\mathbf{s}_{t,j}, \mathbf{a}_{t,j})}_{\text{intrusion and defense costs}} \Bigg)$$

Attacker

- Attacker action: $\mathbf{a}_t^{(A)} \in \{0,1,2,3\}^{|\mathcal{V}|}$
- 0 means do nothing. $1-3$ correspond to attacks (see fig)



- An **attacker strategy** is a function $\pi_{\mathrm{A}} \in \Pi_{\mathrm{A}} : \mathcal{H}_{\mathrm{A}} \rightarrow \Delta(\mathcal{A}_{\mathrm{A}})$, where $\mathcal{H}_{\mathrm{A}}$ is the space of all possible attacker histories

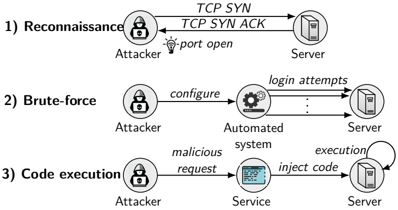
$$\mathbf{h}_t^{(\mathrm{A})} = (\mathbf{s}_1^{(\mathrm{A})}, \mathbf{a}_1^{(\mathrm{A})}, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}^{(\mathrm{A})}, \mathbf{s}_t^{(\mathrm{A})}, \mathbf{o}_t) \in \mathcal{H}_{\mathrm{A}}$$

- Objective: (i) disrupt workflows; and (ii) **compromise nodes**:

$$-J$$

Attacker

- Attacker action: $\mathbf{a}_t^{(A)} \in \{0,1,2,3\}^{|\mathcal{V}|}$
- 0 means do nothing. $1-3$ correspond to attacks (see fig)



- An **attacker strategy** is a function $\pi_{\mathrm{A}} \in \Pi_{\mathrm{A}} : \mathcal{H}_{\mathrm{A}} \rightarrow \Delta(\mathcal{A}_{\mathrm{A}})$, where $\mathcal{H}_{\mathrm{A}}$ is the space of all possible attacker histories

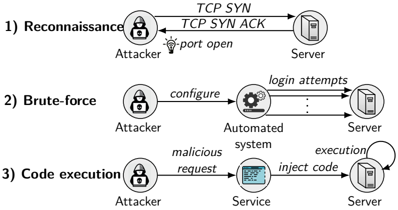
$$\mathbf{h}_t^{(\mathrm{A})} = (\mathbf{s}_1^{(\mathrm{A})}, \mathbf{a}_1^{(\mathrm{A})}, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}^{(\mathrm{A})}, \mathbf{s}_t^{(\mathrm{A})}, \mathbf{o}_t) \in \mathcal{H}_{\mathrm{A}}$$

- Objective: (*i*) disrupt workflows; and (*ii*) **compromise nodes**:

$$-J$$

Attacker

- Attacker action: $\mathbf{a}_t^{(A)} \in \{0, 1, 2, 3\}^{|\mathcal{V}|}$
- 0 means do nothing. $1 - 3$ correspond to attacks (see fig)



- An **attacker strategy** is a function $\pi_{\mathrm{A}} \in \Pi_{\mathrm{A}} : \mathcal{H}_{\mathrm{A}} \to \Delta(\mathcal{A}_{\mathrm{A}})$, where $\mathcal{H}_{\mathrm{A}}$ is the space of all possible attacker histories

$$\mathbf{h}_t^{(\mathrm{A})} = (\mathbf{s}_1^{(\mathrm{A})}, \mathbf{a}_1^{(\mathrm{A})}, \mathbf{o}_1, \ldots, \mathbf{a}_{t-1}^{(\mathrm{A})}, \mathbf{s}_t^{(\mathrm{A})}, \mathbf{o}_t) \in \mathcal{H}_{\mathrm{A}}$$

- Objective: (i) disrupt workflows; and (ii) **compromise nodes**:

$$- J$$

The Intrusion Response Problem

$$
\begin{aligned}
\underset{\pi_{\mathrm{D}}\in\Pi_{\mathrm{D}}}{\text{maximize}}\ \underset{\pi_{\mathrm{A}}\in\Pi_{\mathrm{A}}}{\text{minimize}}\ & \mathbb{E}_{(\pi_{\mathrm{D}},\pi_{\mathrm{A}})}\left[J\right] \\
\text{subject to } & \mathbf{s}^{(\mathrm{D})}_{t+1} \sim f_{\mathrm{D}}\big(\cdot \mid \mathbf{A}^{(\mathrm{D})}_t, \mathbf{A}^{(\mathrm{D})}_t\big) && \forall t \\
& \mathbf{s}^{(\mathrm{A})}_{t+1} \sim f_{\mathrm{A}}\big(\cdot \mid \mathbf{S}^{(\mathrm{A})}_t, \mathbf{A}_t\big) && \forall t \\
& \mathbf{o}_{t+1} \sim Z\big(\cdot \mid \mathbf{S}^{(\mathrm{D})}_{t+1}, \mathbf{A}^{(\mathrm{A})}_t\big) && \forall t \\
& \mathbf{a}^{(\mathrm{A})}_t \sim \pi_{\mathrm{A}}\big(\cdot \mid \mathbf{H}^{(\mathrm{A})}_t\big),\ \mathbf{a}^{(\mathrm{A})}_t \in \mathcal{A}_{\mathrm{A}}(\mathbf{s}_t) && \forall t \\
& \mathbf{a}^{(\mathrm{D})}_t \sim \pi_{\mathrm{D}}\big(\cdot \mid \mathbf{H}^{(\mathrm{D})}_t\big),\ \mathbf{a}^{(\mathrm{D})}_t \in \mathcal{A}_{\mathrm{D}} && \forall t
\end{aligned}
$$

$\mathbb{E}_{(\pi_{\mathrm{D}},\pi_{\mathrm{A}})}$ denotes the expectation of the random vectors $(\mathbf{S}_t, \mathbf{O}_t, \mathbf{A}_t)_{t\in\{1,\ldots,T\}}$ when following the strategy profile $(\pi_{\mathrm{D}}, \pi_{\mathrm{A}})$

(1) can be formulated as a zero-sum Partially Observed Stochastic Game with Public Observations (a PO-POSG):

$$\Gamma = \langle \mathcal{N}, (\mathcal{S}_i)_{i\in\mathcal{N}}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (f_i)_{i\in\mathcal{N}}, u, \gamma, (\mathbf{b}^{(i)}_1)_{i\in\mathcal{N}}, \mathcal{O}, Z \rangle$$

Existence of a Solution

Theorem

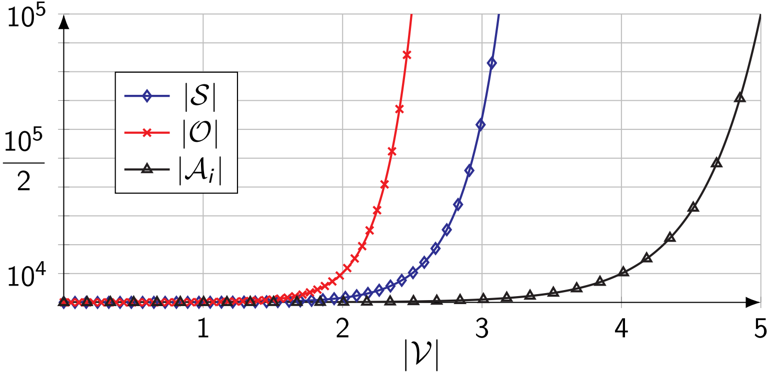
Given the PO-POSG Γ, *the following holds:*

(A) Γ ***has a mixed Nash equilibrium*** *and a value function* $V^\star : \mathcal{B}_{\mathrm{D}} \times \mathcal{B}_{\mathrm{A}} \to \mathbb{R}$.

(B) *For each strategy pair* $(\pi_{\mathrm{A}}, \pi_{\mathrm{D}}) \in \Pi_{\mathrm{A}} \times \Pi_{\mathrm{D}}$, *the* ***best response sets*** $B_{\mathrm{D}}(\pi_{\mathrm{A}})$ ***and*** $B_{\mathrm{A}}(\pi_{\mathrm{D}})$ ***are non-empty***.

The Curse of Dimensionality

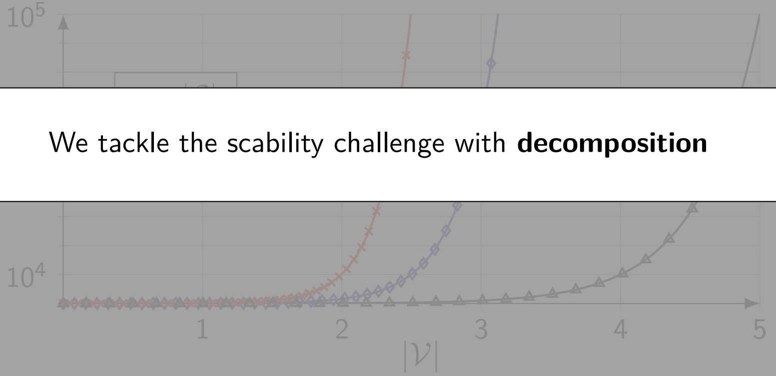
- While Γ has a value, computing it is intractable. The state, action, and observation spaces of the game **grow exponentially** with $|\mathcal{V}|$.



Growth of $|\mathcal{S}|$, $|\mathcal{O}|$, and $|\mathcal{A}_i|$ in function of the number of nodes $|\mathcal{V}|$

The Curse of Dimensionality

- While (1) has a solution (i.e the game Γ has a value (Thm 1)), computing it is intractable since the state, action, and observation spaces of the game **grow exponentially** with $|\mathcal{V}|$.

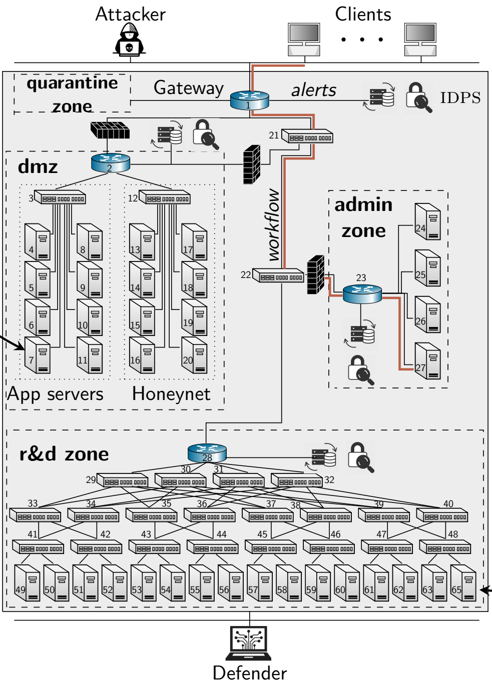


Growth of $|\mathcal{S}|$, $|\mathcal{O}|$, and $|\mathcal{A}_i|$ in function of the number of nodes $|\mathcal{V}|$

We tackle the scability challenge with **decomposition**

Intuitively..

The optimal action here...



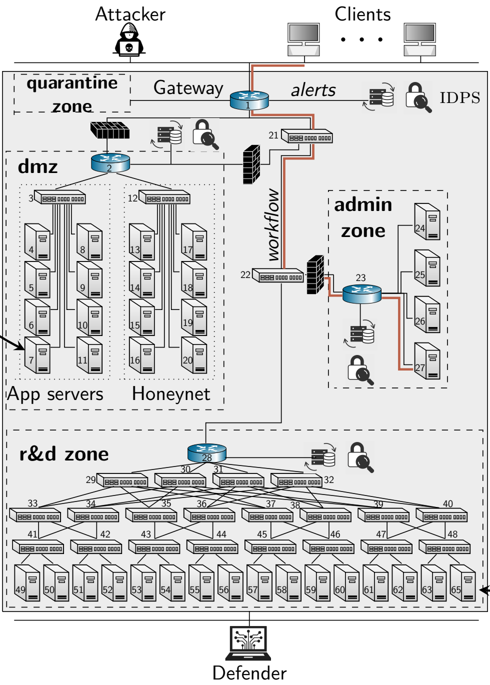
Does not directly depend on the state or action of a node down here

Intuitively..

The optimal action here...

But they are not completely independent either.

How can we exploit this structure?



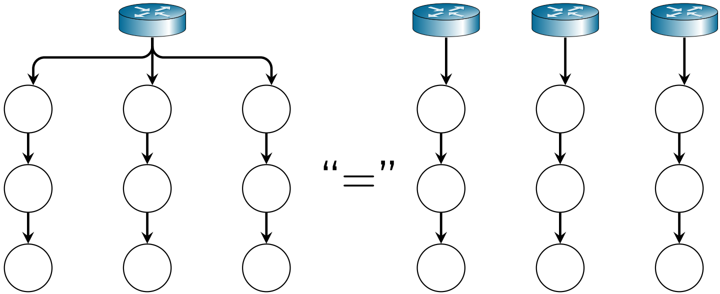
Does not directly depend on the state or action of a node down here

Our Approach: System Decomposition

To avoid explicitly enumerating the very large state, observation, and action spaces of Γ, we exploit three structural properties.

1. **Additive structure across workflows.**
   - The game decomposes into additive subgames on the workflow-level
2. **Optimal substructure within a workflow.**
   - The subgame for each workflow decomposes into subgames on the node-level with *optimal substructure*
3. **Threshold properties of local defender strategies.**
   - Optimal node-level strategies exhibit threshold structures

Additive Structure Across Workflows (Intuition)



- If there is no path between i and j in $\mathcal{G}$, then i and j are **independent** in the following sense:
  - Compromising i has no affect on the state of j.
  - Compromising i does not make it harder or easier to compromise j.
  - Compromising i does not affect the service provided by j.
  - Defending i does not affect the state of j.
  - Defending i does not affect the service provided by j.

Additive Structure Across Workflows

Definition (Transition independence)

A set of nodes $\mathcal{Q}$ are transition independent iff the transition probabilities factorize as

$$f(\mathbf{S}_{t+1} \mid \mathbf{S}_t, \mathbf{A}_t) = \prod_{i \in \mathcal{Q}} f(\mathbf{S}_{t+1,i} \mid \mathbf{S}_{t,i}, \mathbf{A}_{t,i})$$

Definition (Utility independence)

A set of nodes $\mathcal{Q}$ are utility independent iff there exists functions $u_1, \ldots, u_{|\mathcal{Q}|}$ such that the utility function u decomposes as

$$u(\mathbf{S}_t, \mathbf{A}_t) = f(u_1(\mathbf{S}_{t,1}, \mathbf{A}_{t,1}), \ldots, u_1(\mathbf{S}_{t,|\mathcal{Q}|}, \mathbf{A}_{t,\mathcal{Q}}))$$

and

$$u_i \leq u_i' \iff f(u_1, \ldots, u_i, \ldots, u_{|\mathcal{Q}|}) \leq f(u_1, \ldots, u_i', \ldots, u_{|\mathcal{Q}|})$$

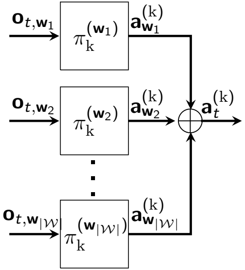
Additive Structure Across Workflows

Theorem (Node independencies)

(A) All nodes $\mathcal{V}$ in the game Γ are transition independent.
(B) If there is no path between i and j in the topology graph $\mathcal{G}$, then i and j are utility independent.

Corollary (Additive structure across workflows)

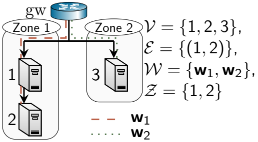
Γ decomposes into $|\mathcal{W}|$ additive subproblems that can be solved independently and in parallel.



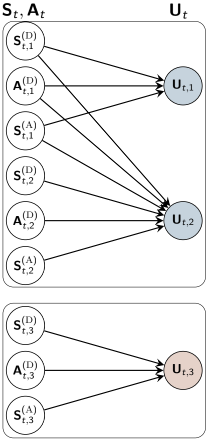
Optimal Substructure Within a Workflow

- Nodes in the same workflow are utility dependent.
- $\implies$ Adding locally-optimal strategies **does not** yield an optimal workflow strategy.
- However, the locally-optimal strategies satisfy the optimal substructure property.
- $\implies$ there exists an algorithm for constructing an optimal workflow strategy from locally-optimal strategies for each node.

IT infrastructure

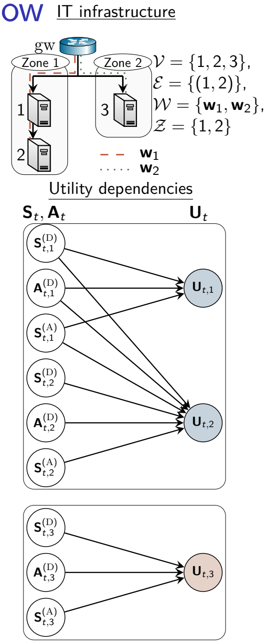


Utility dependencies

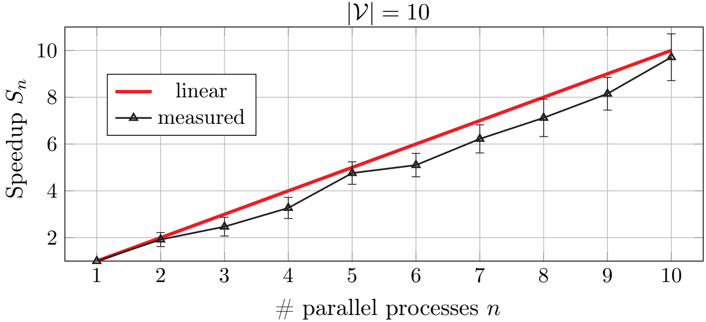


Optimal Substructure Within a Workflow

- Nodes in the same workflow are utility dependent.
- $\implies$ Adding locally-optimal strategies **does not** yield an optimal workflow strategy.
- However, the locally-optimal strategies satisfy the optimal substructure property.
- $\implies$ there exists an algorithm for constructing an optimal workflow strategy from locally-optimal strategies for each node.

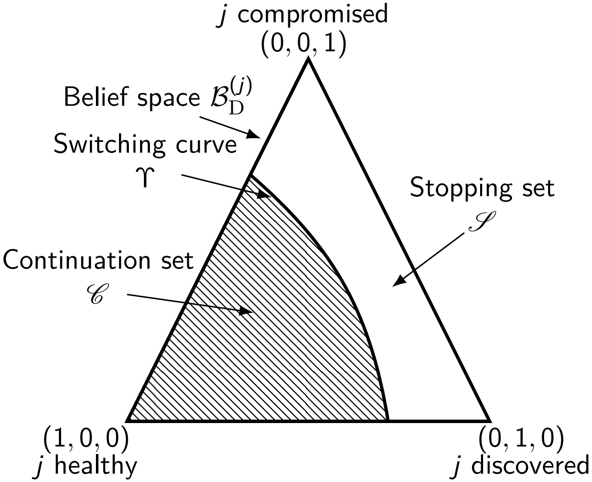


Scalable Learning through Decomposition



Speedup of best response computation for the decomposed game; T_n denotes the completion time with n processes; the speedup is calculated as $S_n = \frac{T_1}{T_n}$; the error bars indicate standard deviations from 3 measurements.

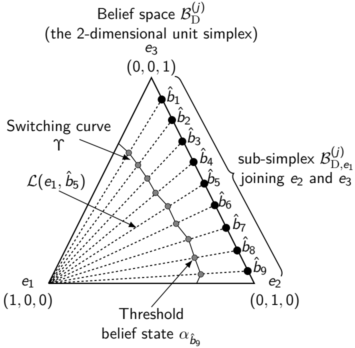
Threshold Properties of Local Defender Strategies.



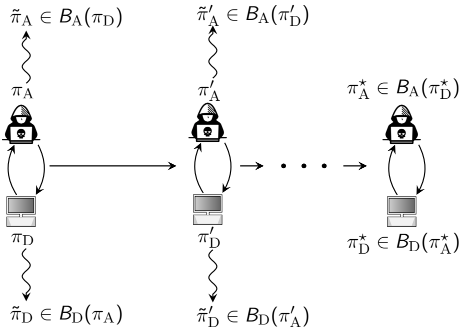
- A node can be in three attack states $s_t^{(\mathrm{A})}$: Healthy, Discovered, Compromised.
- The defender has a belief state $\mathbf{b}_t^{(\mathrm{D})}$

Proof Sketch (Threshold Properties)

- Let $\mathcal{L}(e_1, \hat{b})$ denote the line segment that starts at the belief state $e_1 = (1, 0, 0)$ and ends at $\hat{b}$, where $\hat{b}$ is in the sub-simplex that joins e_2 and e_3.
- All beliefs on $\mathcal{L}(e_1, \hat{b})$ are totally ordered according to the Monotone Likelihood Ratio (MLR) order. $\implies$ a threshold belief state $\alpha_{\hat{b}} \in \mathcal{L}(e_1, \hat{b})$ exists where the optimal strategy switches from C to S.
- Since the entire belief space can be covered by the union of lines $\mathcal{L}(e_1, \hat{b})$, the threshold belief states $\alpha_{\hat{b}_1}, \alpha_{\hat{b}_2}, \dots$ yield a switching curve Υ.



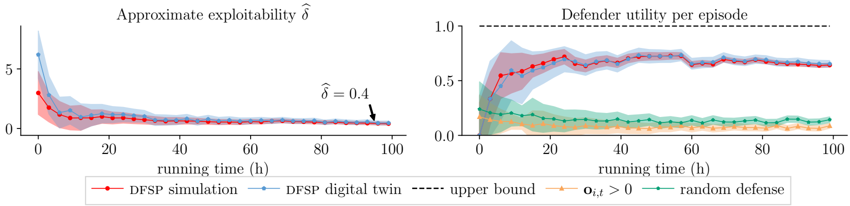
Decompositional Fictitious Play (DFSP)



Fictitious play: iterative averaging of best responses.

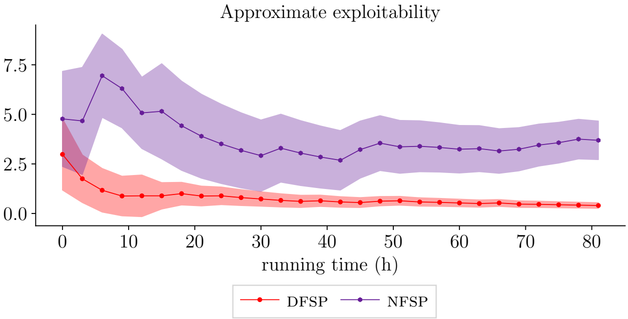
- **Learn best response** strategies iteratively through the parallel solving of subgames in the decomposition
- Average best responses to **approximate the equilibrium**

Learning Equilibrium Strategies



Learning curves obtained during training of DFSP to find optimal (equilibrium) strategies in the intrusion response game; **red and blue curves relate to dfsp**; black, orange and green curves relate to baselines.

Comparison with NFSP



Learning curves obtained during training of DFSP and NFSP to find optimal (equilibrium) strategies in the intrusion response game; **the red curve relate to dfsp** and the purple curve relate to NFSP; all curves show simulation results.

Conclusions

- We develop a *framework* to automatically learn security strategies.
- We apply the framework to an **intrusion response use case**.
- We derive properties of optimal security strategies.
- We evaluate strategies on a **digital twin**.
- Questions $\rightarrow$ demonstration