CENTER FOR
CYBER DEFENSE AND
INFORMATION SECURITY

Self-Learning Systems for Cyber Security

Kim Hammar & Rolf Stadler

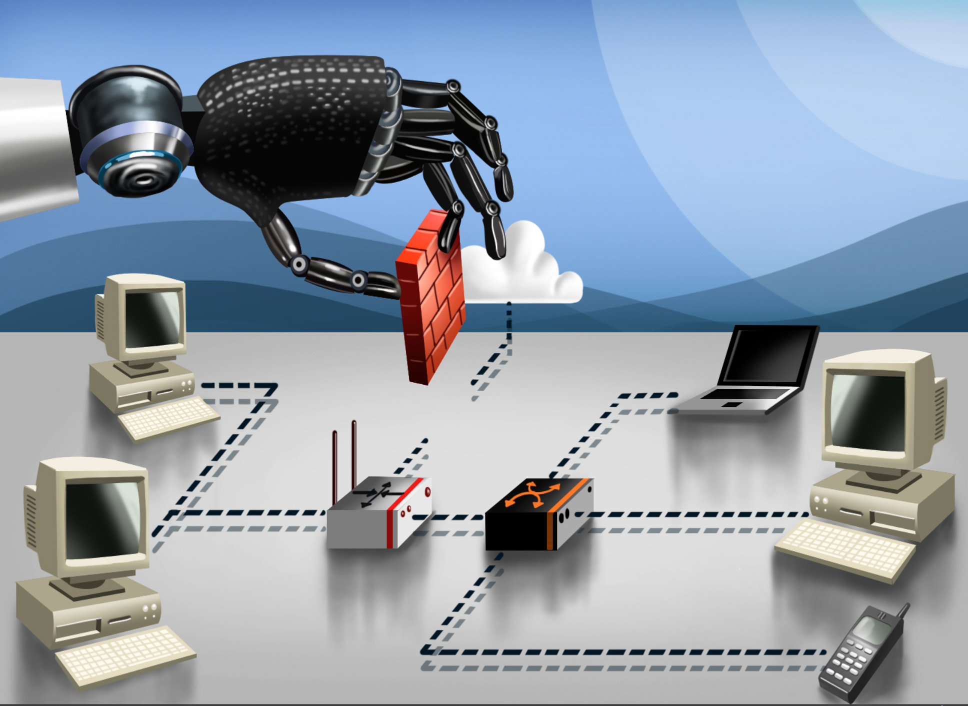
kimham@kth.se, stadler@kth.se
KTH Royal Institute of Technology

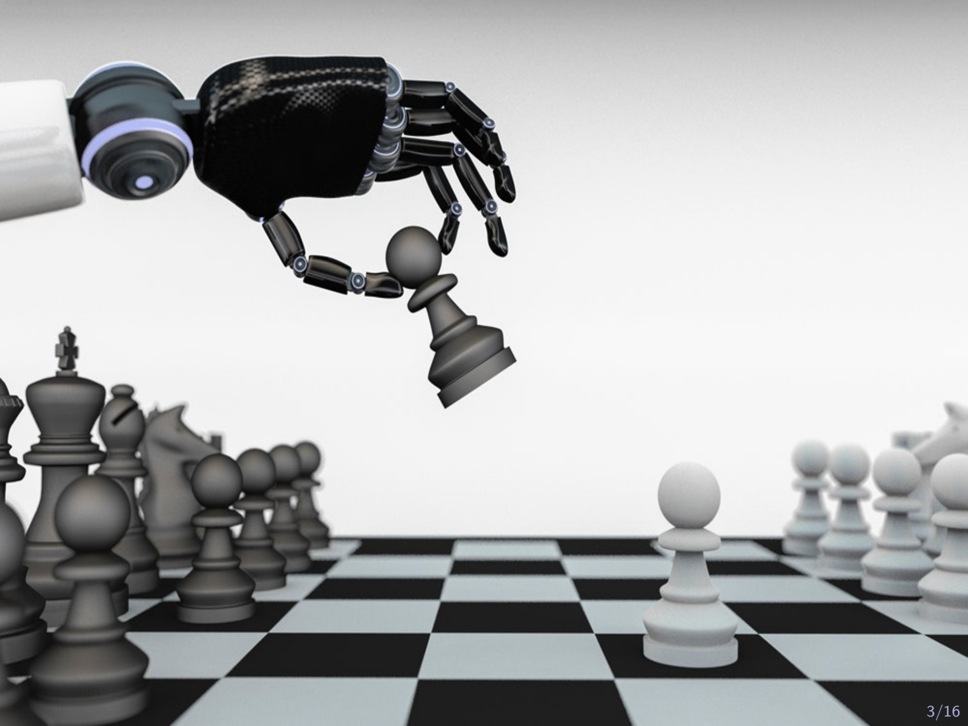
CDIS Spring Conference 2021
March 24, 2021

KTH VETENSKAP OCH KONST

FÖRSVARSMAKTEN

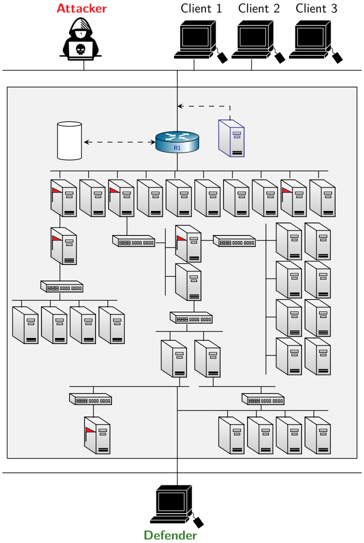
Försvarshögskolan





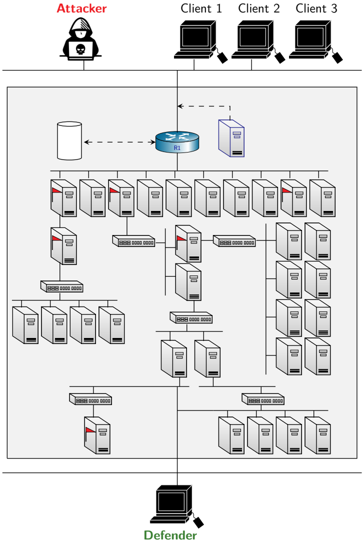
Challenges: Evolving and Automated Attacks

- **Challenges**:
  - Evolving & automated attacks
  - Complex infrastructures



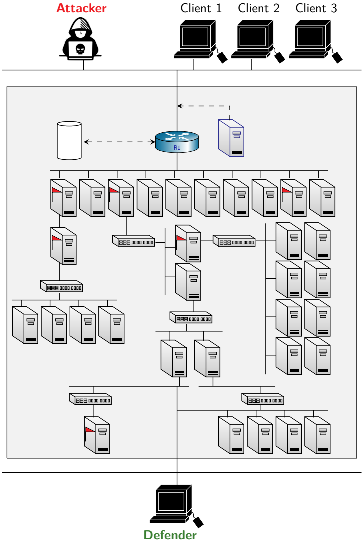
Goal: Automation and Learning

- **Challenges**
  - Evolving & automated attacks
  - Complex infrastructures
- **Our Goal**:
  - Automate security tasks
  - Adapt to changing attack methods



Approach: Game Model & Reinforcement Learning

- **Challenges**:
  - Evolving & automated attacks
  - Complex infrastructures
- **Our Goal**:
  - Automate security tasks
  - Adapt to changing attack methods
- **Our Approach**:
  - Model network attack and defense as *games*.
  - Use *reinforcement learning* to learn policies.
  - Incorporate learned policies in *self-learning systems*.



State of the Art

- **Game-Learning Programs**:
  - TD-Gammon, AlphaGo Zero[1], OpenAI Five etc.
  - $\implies$ Impressive empirical results of *RL and self-play*
- **Attack Simulations**:
  - Automated threat modeling[2], automated intrusion detection etc.
  - $\implies$ Need for *automation* and better security tooling
- **Mathematical Modeling**:
  - Game theory[3]
  - Markov decision theory
  - $\implies$ Many security operations involves *strategic decision making*

---

[1] David Silver et al. "Mastering the game of Go without human knowledge". In: *Nature* 550 (Oct. 2017), pp. 354–. URL: http://dx.doi.org/10.1038/nature24270.

[2] Pontus Johnson, Robert Lagerström, and Mathias Ekstedt. "A Meta Language for Threat Modeling and Attack Simulations". In: *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: Association for Computing Machinery, 2018. ISBN: 9781450364485. DOI: 10.1145/3230833.3232799. URL: https://doi.org/10.1145/3230833.3232799.

[3] Tansu Alpcan and Tamer Basar. *Network Security: A Decision and Game-Theoretic Approach*. 1st. USA: Cambridge University Press, 2010. ISBN: 0521119324.

State of the Art

- **Game-Learning Programs**:
  - TD-Gammon, AlphaGo Zero[4], OpenAI Five etc.
  - $\Longrightarrow$ Impressive empirical results of *RL and self-play*
- **Attack Simulations**:
  - Automated threat modeling[5], automated intrusion detection etc.
  - $\Longrightarrow$ Need for *automation* and better security tooling
- **Mathematical Modeling**:
  - Game theory[6]
  - Markov decision theory
  - $\Longrightarrow$ Many security operations involves *strategic decision making*

---

[4] David Silver et al. "Mastering the game of Go without human knowledge". In: *Nature* 550 (Oct. 2017), pp. 354–. URL: http://dx.doi.org/10.1038/nature24270.

[5] Pontus Johnson, Robert Lagerström, and Mathias Ekstedt. "A Meta Language for Threat Modeling and Attack Simulations". In: *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: Association for Computing Machinery, 2018. ISBN: 9781450364485. DOI: 10.1145/3230833.3232799. URL: https://doi.org/10.1145/3230833.3232799.

[6] Tansu Alpcan and Tamer Basar. *Network Security: A Decision and Game-Theoretic Approach*. 1st. USA: Cambridge University Press, 2010. ISBN: 0521119324.

State of the Art

- **Game-Learning Programs**:
  - TD-Gammon, AlphaGo Zero[7], OpenAI Five etc.
  - $\implies$ Impressive empirical results of *RL and self-play*
- **Attack Simulations**:
  - Automated threat modeling[8], automated intrusion detection etc.
  - $\implies$ Need for *automation* and better security tooling
- **Mathematical Modeling**:
  - Game theory[9]
  - Markov decision theory
  - $\implies$ Many security operations involves *strategic decision making*

---

[7] David Silver et al. "Mastering the game of Go without human knowledge". In: *Nature* 550 (Oct. 2017), pp. 354–. URL: http://dx.doi.org/10.1038/nature24270.

[8] Pontus Johnson, Robert Lagerström, and Mathias Ekstedt. "A Meta Language for Threat Modeling and Attack Simulations". In: *Proceedings of the 13th International Conference on Availability, Reliability and Security*. ARES 2018. Hamburg, Germany: Association for Computing Machinery, 2018. ISBN: 9781450364485. DOI: 10.1145/3230833.3232799. URL: https://doi.org/10.1145/3230833.3232799.

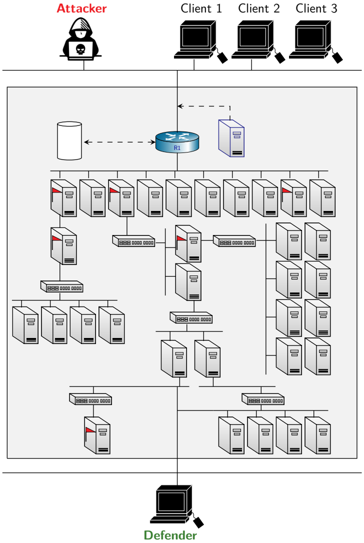
[9] Tansu Alpcan and Tamer Basar. *Network Security: A Decision and Game-Theoretic Approach*. 1st. USA: Cambridge University Press, 2010. ISBN: 0521119324.

Our Work

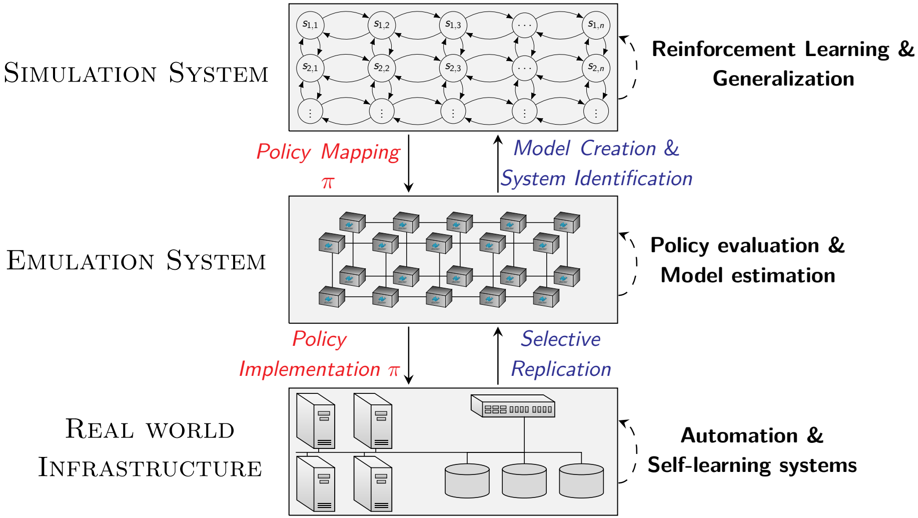
- **Use Case:** Intrusion Prevention
- **Our Method:**
  - Emulating computer infrastructures
  - System identification and model creation
  - Reinforcement learning and generalization
- **Results:** Learning to Capture The Flag
- **Conclusions and Future Work**

Use Case: Intrusion Prevention

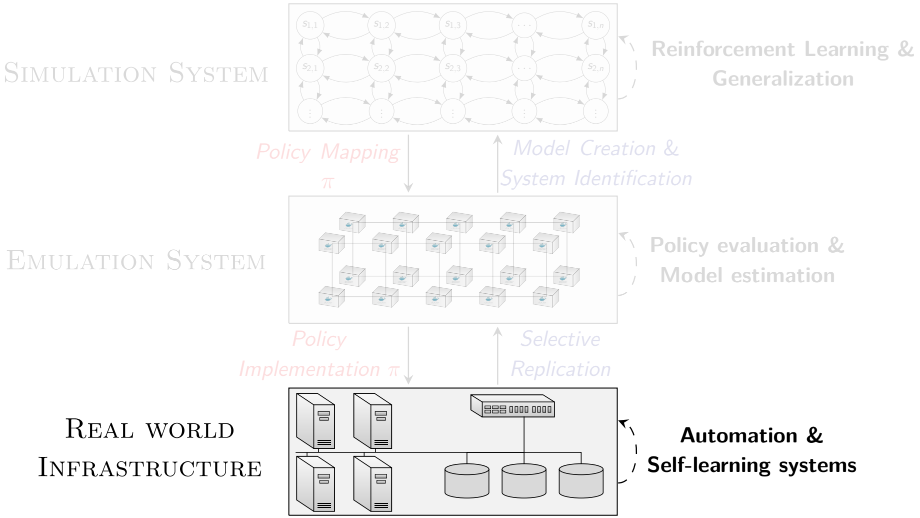
- A **Defender** owns an infrastructure
  - Consists of connected components
  - Components run network services
  - Defender defends the infrastructure by monitoring and patching
- An **Attacker** seeks to intrude on the infrastructure
  - Has a partial view of the infrastructure
  - Wants to compromise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



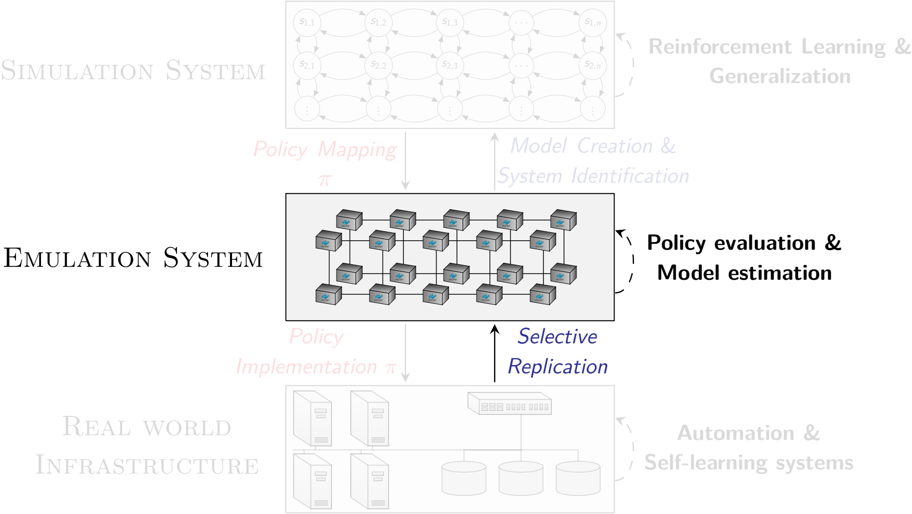
Our Method for Finding Effective Security Strategies



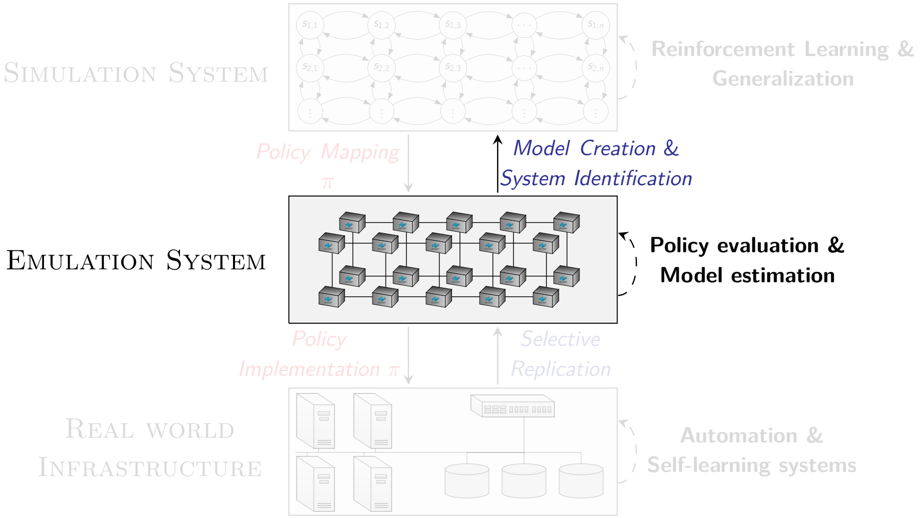
Our Method for Finding Effective Security Strategies



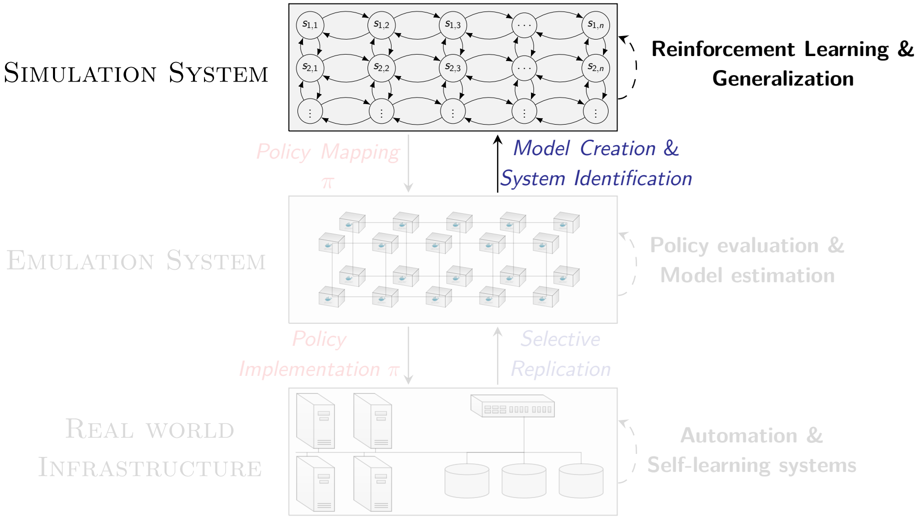
Our Method for Finding Effective Security Strategies



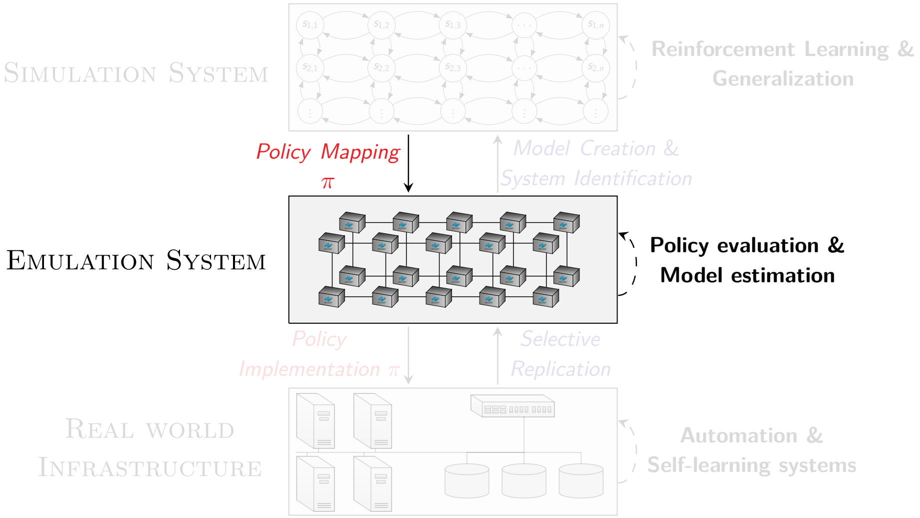
Our Method for Finding Effective Security Strategies



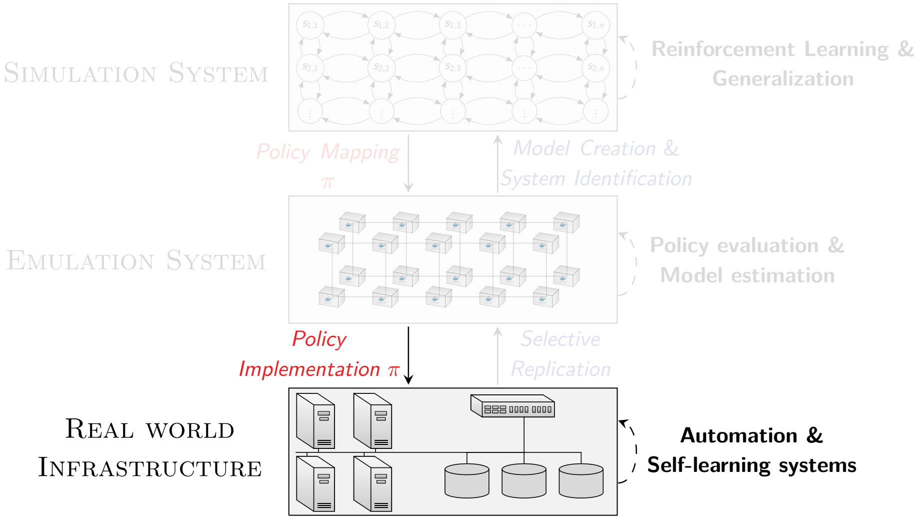
Our Method for Finding Effective Security Strategies



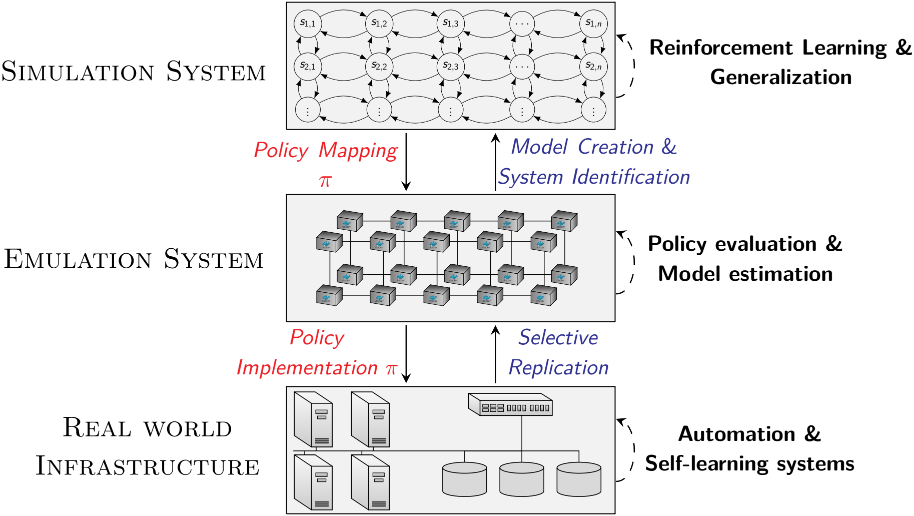
Our Method for Finding Effective Security Strategies



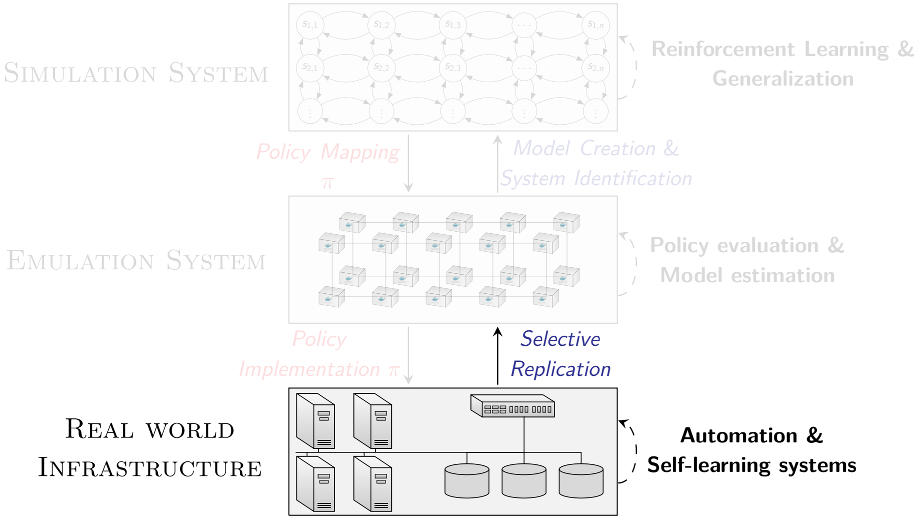
Our Method for Finding Effective Security Strategies



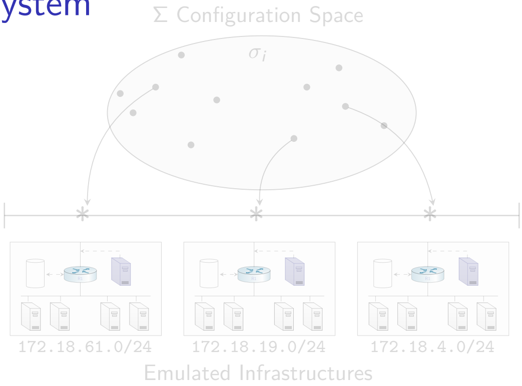
Our Method for Finding Effective Security Strategies



Our Method for Finding Effective Security Strategies



Emulation System

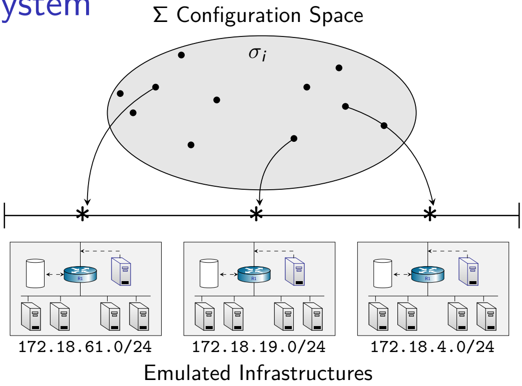


Emulation

A cluster of machines that runs a virtualized infrastructure which replicates important functionality of target systems.

- The set of virtualized configurations define a *configuration space* $\Sigma = \langle \mathcal{A}, \mathcal{O}, \mathcal{S}, \mathcal{U}, \mathcal{T}, \mathcal{V} \rangle$.
- A specific emulation is based on a configuration $\sigma_i \in \Sigma$.

Emulation System

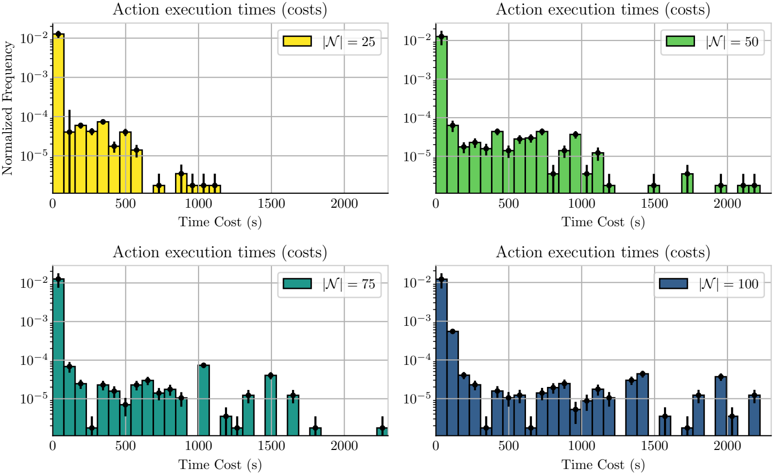


Emulation

A cluster of machines that runs a virtualized infrastructure which replicates important functionality of target systems.

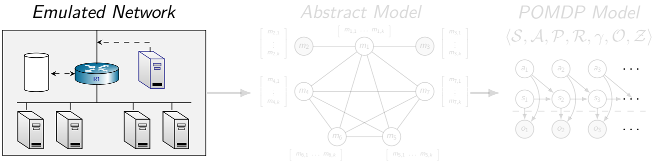
- The set of virtualized configurations define a *configuration space* $\Sigma = \langle \mathcal{A}, \mathcal{O}, \mathcal{S}, \mathcal{U}, \mathcal{T}, \mathcal{V} \rangle$.
- A specific emulation is based on a configuration $\sigma_i \in \Sigma$.

Emulation: Execution Times of Replicated Operations



- **Fundamental issue**: Computational methods for policy learning typically require samples on the order of $100k - 10M$.
- $\implies$ Infeasible to optimize in the emulation system

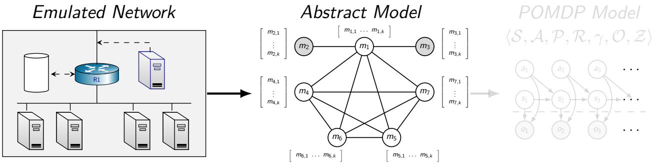
From Emulation to Simulation: System Identification



- **Abstract Model Based on Domain Knowledge**: Models the set of *controls*, the *objective function*, and the *features* of the emulated network.
  - Defines the static parts a POMDP model.
- **Dynamics Model ($\mathcal{P}$, $\mathcal{Z}$) Identified using System Identification**: Algorithm based on random walks and maximum-likelihood estimation.

$$\mathcal{M}(b'|b,a) \triangleq \frac{n(b,a,b')}{\sum_{j'} n(s,a,j')}$$

From Emulation to Simulation: System Identification

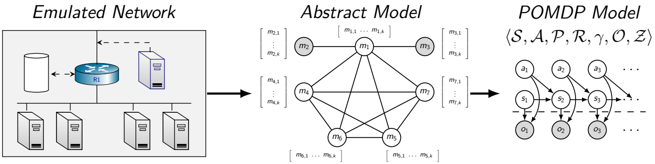


- **Abstract Model Based on Domain Knowledge**: Models the set of *controls*, the *objective function*, and the *features* of the emulated network.
  - Defines the static parts a POMDP model.

- **Dynamics Model ($\mathcal{P}$, $\mathcal{Z}$) Identified using System Identification**: Algorithm based on random walks and maximum-likelihood estimation.

$$\mathcal{M}(b'|b,a) \triangleq \frac{n(b,a,b')}{\sum_{j'} n(s,a,j')}$$

From Emulation to Simulation: System Identification



- **Abstract Model Based on Domain Knowledge**: Models the set of *controls*, the *objective function*, and the *features* of the emulated network.
  - Defines the static parts a POMDP model.

- **Dynamics Model ($\mathcal{P}$, $\mathcal{Z}$) Identified using System Identification**: Algorithm based on random walks and maximum-likelihood estimation.

$$\mathcal{M}(b'|b, a) \triangleq \frac{n(b, a, b')}{\sum_{j'} n(s, a, j')}$$

Policy Optimization in the Simulation System using Reinforcement Learning

- **Goal**:
  - Approximate $\pi^* = \arg\max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_{t+1}\right]$

- **Learning Algorithm**:
  - Represent π by π_θ
  - Define objective $J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}[R]$
  - Maximize $J(\theta)$ by stochastic gradient ascent with gradient $\nabla_\theta J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|o) A^{\pi_\theta}(o, a)\right]$

- **Domain-Specific Challenges**:
  - Partial observability
  - Large state space $|\mathcal{S}| = (w+1)^{|\mathcal{N}| \cdot m \cdot (m+1)}$
  - Large action space $|\mathcal{A}| = |\mathcal{N}| \cdot (m+1)$
  - Non-stationary Environment due to presence of adversary
  - Generalization

Policy Optimization in the Simulation System using Reinforcement Learning

- **Goal**:
  - Approximate $\pi^* = \arg\max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_{t+1}\right]$
- **Learning Algorithm**:
  - Represent π by π_θ
  - Define objective $J(\theta) = \mathbb{E}_{o\sim\rho^{\pi_\theta}, a\sim\pi_\theta}[R]$
  - Maximize $J(\theta)$ by stochastic gradient ascent with gradient $\nabla_\theta J(\theta) = \mathbb{E}_{o\sim\rho^{\pi_\theta}, a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|o) A^{\pi_\theta}(o, a)\right]$
- **Domain-Specific Challenges**:
  - Partial observability
  - Large state space $|\mathcal{S}| = (w+1)^{|\mathcal{N}|\cdot m\cdot(m+1)}$
  - Large action space $|\mathcal{A}| = |\mathcal{N}| \cdot (m+1)$
  - Non-stationary Environment due to presence of adversary
  - Generalization

Policy Optimization in the Simulation System using Reinforcement Learning

- **Goal**:
  - Approximate $\pi^* = \arg\max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_{t+1}\right]$

- **Learning Algorithm**:
  - Represent π by π_θ
  - Define objective $J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}[R]$
  - Maximize $J(\theta)$ by stochastic gradient ascent with gradient $\nabla_\theta J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|o) A^{\pi_\theta}(o, a)\right]$

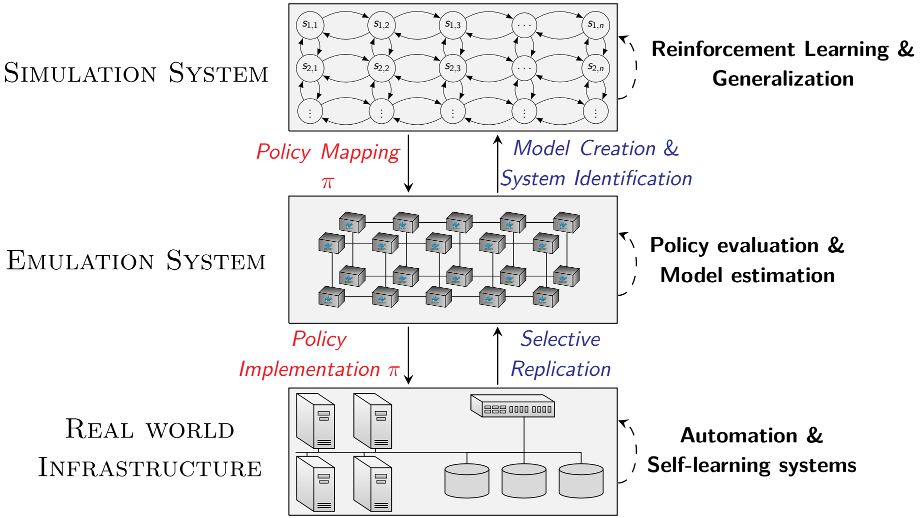
- **Domain-Specific Challenges**:
  - Partial observability
  - Large state space $|\mathcal{S}| = (w+1)^{|\mathcal{N}| \cdot m \cdot (m+1)}$
  - Large action space $|\mathcal{A}| = |\mathcal{N}| \cdot (m+1)$
  - Non-stationary Environment due to presence of adversary
  - Generalization

Policy Optimization in the Simulation System using Reinforcement Learning

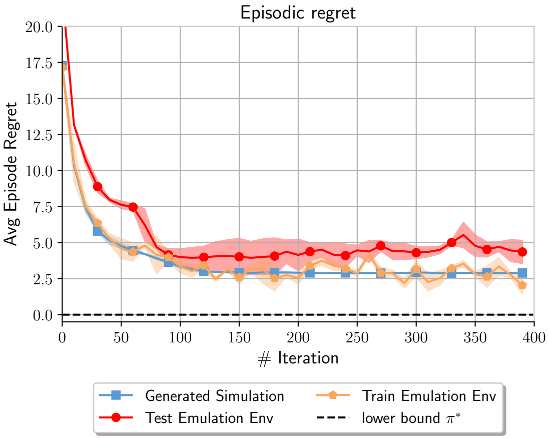
- **Goal**:
  - Approximate $\pi^* = \arg\max_\pi \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_{t+1}\right]$
- **Learning Algorithm**:
  - Represent π by π_θ
  - Define objective $J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}[R]$
  - Maximize $J(\theta)$ by stochastic gradient ascent with gradient $\nabla_\theta J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|o) A^{\pi_\theta}(o,a)\right]$
- **Domain-Specific Challenges**:
  - Partial observability
  - Large state space $|\mathcal{S}| = (w+1)^{|\mathcal{N}| \cdot m \cdot (m+1)}$
  - Large action space $|\mathcal{A}| = |\mathcal{N}| \cdot (m+1)$
  - Non-stationary Environment due to presence of adversary
  - Generalization
- Finding Effective Security Strategies through Reinforcement Learning and Self-Play[a]

[a] Kim Hammar and Rolf Stadler. "Finding Effective Security Strategies through Reinforcement Learning and Self-Play". In: *International Conference on Network and Service Management (CNSM 2020) (CNSM 2020)*. Izmir, Turkey, Nov. 2020.

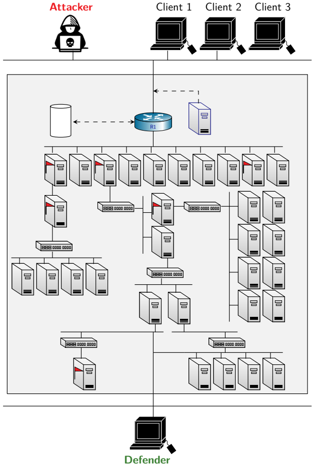
Our Method for Finding Effective Security Strategies



Learning Capture-the-Flag Strategies

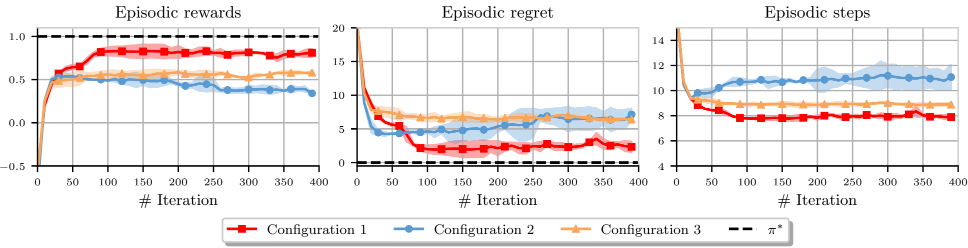


Learning curves (train and eval) of our proposed method.

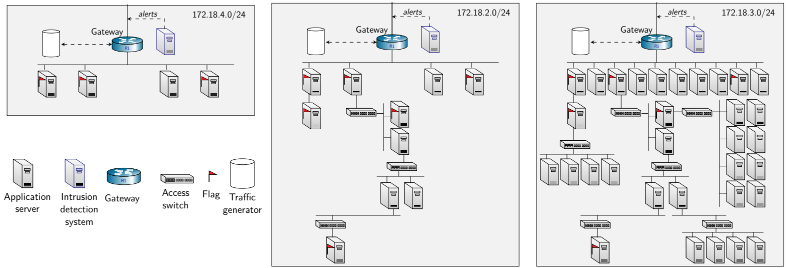


Evaluation infrastructure.

Learning Capture-the-Flag Strategies



Configuration 1 Configuration 2 Configuration 3



Conclusions & Future Work

- **Conclusions:**
  - We develop a *method* to find effective strategies for intrusion prevention
    - (1) emulation system; (2) system identification; (3) simulation system; (4) reinforcement learning and (5) domain randomization and generalization.
  - We show that self-learning can be successfully applied to network infrastructures.
    - Self-play reinforcement learning in Markov security game
  - *Key challenges*: stable convergence, sample efficiency, complexity of emulations, large state and action spaces
- **Our research plans:**
  - Improving the system identification algorithm & generalization
  - Evaluation on real world infrastructures