Finding Effective Security Strategies through Reinforcement Learning and Self-Play[1]

CNSM 2020 | International Conference on Network and Service Management

Kim Hammar & Rolf Stadler

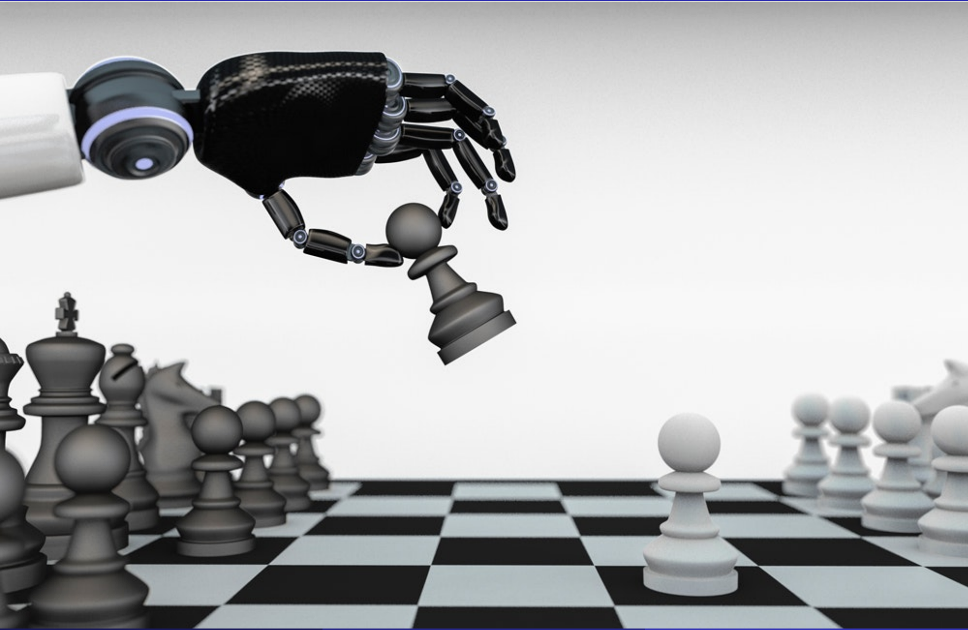
kimham@kth.se & *stadler@kth.se*

Division of Network and Systems Engineering
KTH Royal Institute of Technology

November 3, 2020

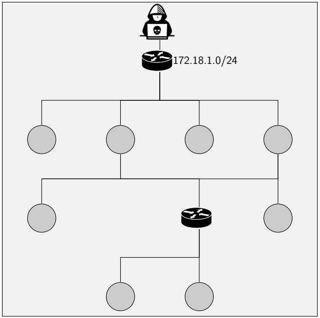
[1] Kim Hammar and Rolf Stadler. "Finding Effective Security Strategies through Reinforcement Learning and Self-Play". In: *International Conference on Network and Service Management (CNSM 2020) (CNSM 2020)*. Izmir, Turkey, Nov. 2020.

Game Learning Programs



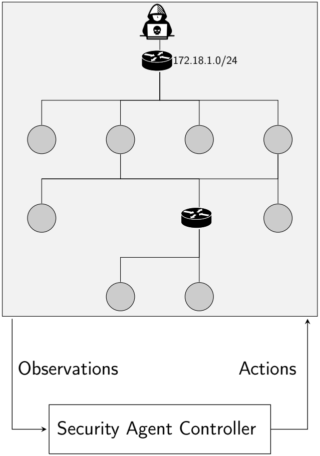
Challenges: Evolving and Automated Attacks

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



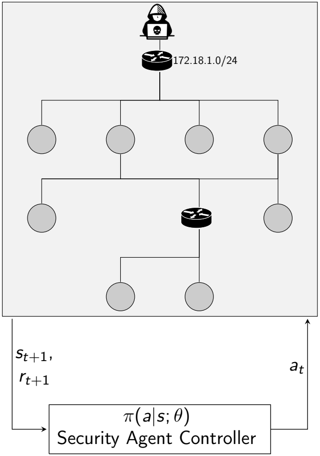
Goal: Automation and Learning

- 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 as Markov Game $\mathcal{M}_G = \langle \mathcal{S}, \mathcal{A}_1, \dots, \mathcal{A}_N, \mathcal{T}, \mathcal{R}_1, \dots \mathcal{R}_N \rangle$
  - Compute policies π for $\mathcal{M}_G$
  - Incorporate π in self-learning systems



Related Work

- **Game-Learning Programs**:
  - TD-Gammon[2], AlphaGo Zero[3], OpenAI Five etc.
  - $\Longrightarrow$ Impressive empirical results of *RL and self-play*
- **Network Security**:
  - Automated threat modeling[4], automated intrusion detection etc.
  - $\Longrightarrow$ Need for *automation* and better security tooling
- **Game Theory**:
  - Network Security: A Decision and Game-Theoretic Approach[5].
  - $\Longrightarrow$ Many security operations involves *strategic decision making*

---

[2] Gerald Tesauro. "TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play". In: *Neural Comput.* 6.2 (Mar. 1994), 215–219. ISSN: 0899-7667. DOI: 10.1162/neco.1994.6.2.215. URL: https://doi.org/10.1162/neco.1994.6.2.215.

[3] 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.

[4] 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.

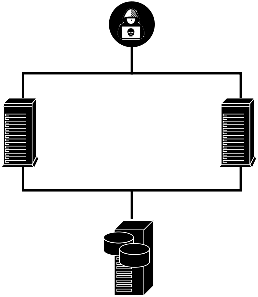
[5] Tansu Alpcan and Tamer Basar. *Network Security: A Decision and Game-Theoretic Approach*. 1st. USA: Cambridge University Press, 2010. ISBN: 0521119324.

Outline

- Use Case
- Markov Game Model for Intrusion Prevention
- Reinforcement Learning Problem
- Method
- Results
- Conclusions

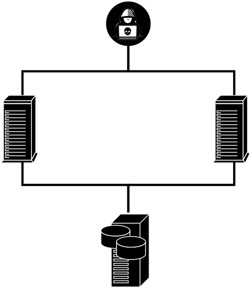
Use Case: Intrusion Prevention

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



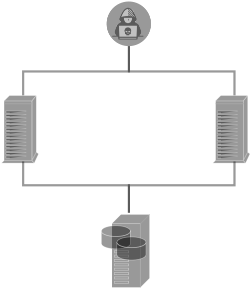
Markov Game Model for Intrusion Prevention

(1) Network Infrastructure

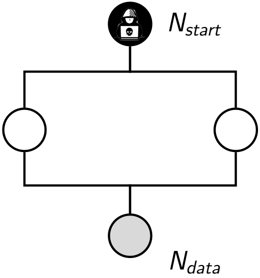


Markov Game Model for Intrusion Prevention

(1) Network Infrastructure

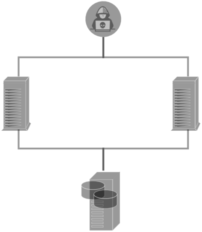


(2) Graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$

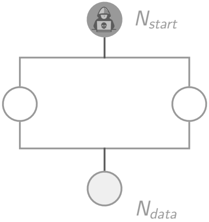


Markov Game Model for Intrusion Prevention

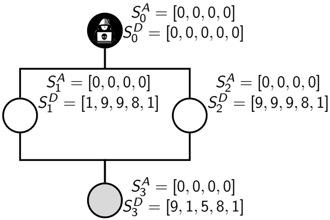
(1) Network Infrastructure



(2) Graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$

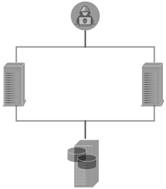


(3) State space $|\mathcal{S}| = (w+1)^{|\mathcal{N}| \cdot m \cdot (m+1)}$

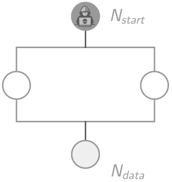


Markov Game Model for Intrusion Prevention

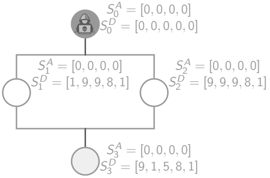
(1) Network Infrastructure



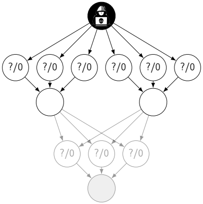
(2) Graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$



(3) State space $|\mathcal{S}| = (w+1)^{|\mathcal{N}| \cdot m \cdot (m+1)}$

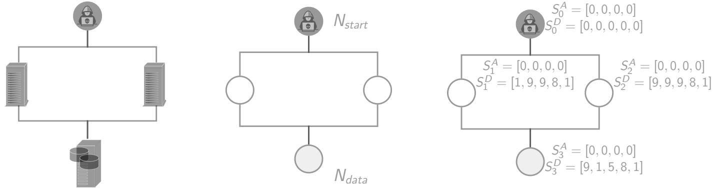


(4) Action space $|\mathcal{A}| = |\mathcal{N}| \cdot (m+1)$

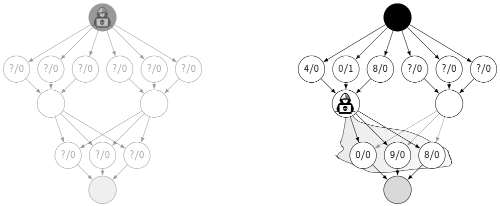


Markov Game Model for Intrusion Prevention

(1) Network Infrastructure (2) Graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ (3) State space $|\mathcal{S}| = (w+1)^{|\mathcal{N}| \cdot m \cdot (m+1)}$

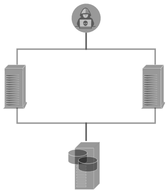


(4) Action space $|\mathcal{A}| = |\mathcal{N}| \cdot (m+1)$ (5) Game Dynamics $\mathcal{T}, \mathcal{R}_1, \mathcal{R}_2, \rho_0$

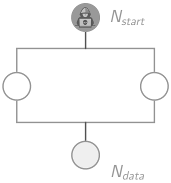


Markov Game Model for Intrusion Prevention

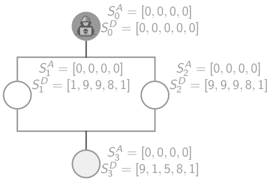
(1) Network Infrastructure



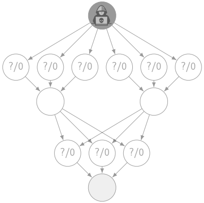
(2) Graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$



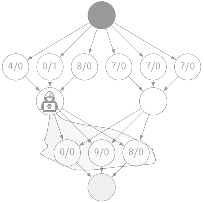
(3) State space $|\mathcal{S}| = (w+1)^{|\mathcal{N}| \cdot m \cdot (m+1)}$



(4) Action space $|\mathcal{A}| = |\mathcal{N}| \cdot (m+1)$



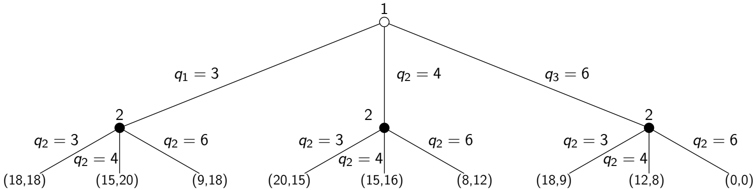
(5) Game Dynamics $\mathcal{T}, \mathcal{R}_1, \mathcal{R}_2, \rho_0$



- Markov game
- Zero-sum
- 2 players
- Partially observed
- Stochastic elements
- Round-based

$\mathcal{M}_G = \langle \mathcal{S}, \mathcal{A}_1, \mathcal{A}_2, \mathcal{T}, \mathcal{R}_1, \mathcal{R}_2, \gamma, \rho_0 \rangle$

Automatic Learning of Security Strategies



- **Finding strategies for the Markov game model**:
  - Evolutionary methods
  - Computational game theory
  - Self-Play Reinforcement learning
    - Attacker vs Defender
    - Strategies evolve without human intervention
- **Motivation for Reinforcement Learning**:
  - Strong empirical results in related work
  - Can adapt to new attack methods and threats
  - Can be used for complex domains that are hard to model exactly

The Reinforcement Learning Problem

- **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: captured in the model
  - 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

- 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: captured in the model
  - 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

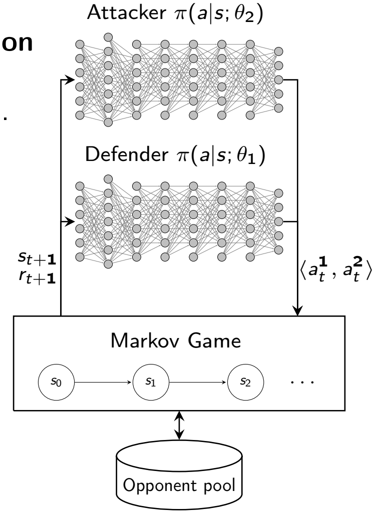
The Reinforcement Learning Problem

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

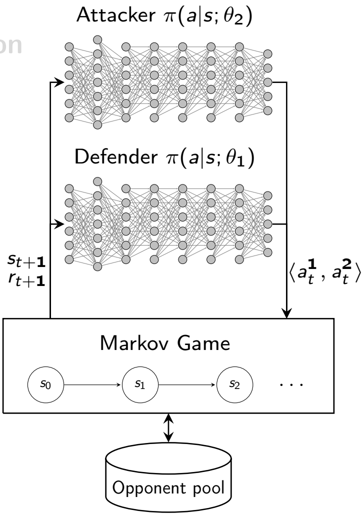
- **Learning Algorithm**:
  - Represent π by π_θ
  - Define objective $J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}[R]$
  - Maximize $J(\theta)$ by stochastic gradient ascent with gradient $\nabla_\theta J(\theta) = \mathbb{E}_{o \sim \rho^{\pi_\theta}, a \sim \pi_\theta}\left[\nabla_\theta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

- **Domain-Specific Challenges**:
  - Partial observability: captured in the model
  - 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

- **Policy Gradient & Function Approximation**
  - To deal with large state space $\mathcal{S}$
  - π_θ parameterized by weights $\theta \in \mathbb{R}^d$ of NN.
  - PPO & REINFORCE (stochastic π)
- **Auto-Regressive Policy Representation**
  - To deal with large action space $\mathcal{A}$
  - To minimize interference
  - $\pi(a, n|o) = \pi(a|n, o) \cdot \pi(n|o)$
- **Opponent Pool**
  - To avoid overfitting
  - Want agent to learn a general strategy

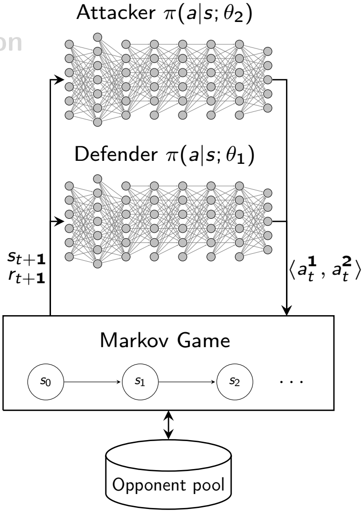


- Policy Gradient & Function Approximation
  - To deal with large state space $\mathcal{S}$
  - π_θ parameterized by weights $\theta \in \mathbb{R}^d$ of NN.
  - PPO & REINFORCE (stochastic π)
- **Auto-Regressive Policy Representation**
  - To deal with large action space $\mathcal{A}$
  - To minimize interference
  - $\pi(a, n|o) = \pi(a|n, o) \cdot \pi(n|o)$
- Opponent Pool
  - To avoid overfitting
  - Want agent to learn a general strategy

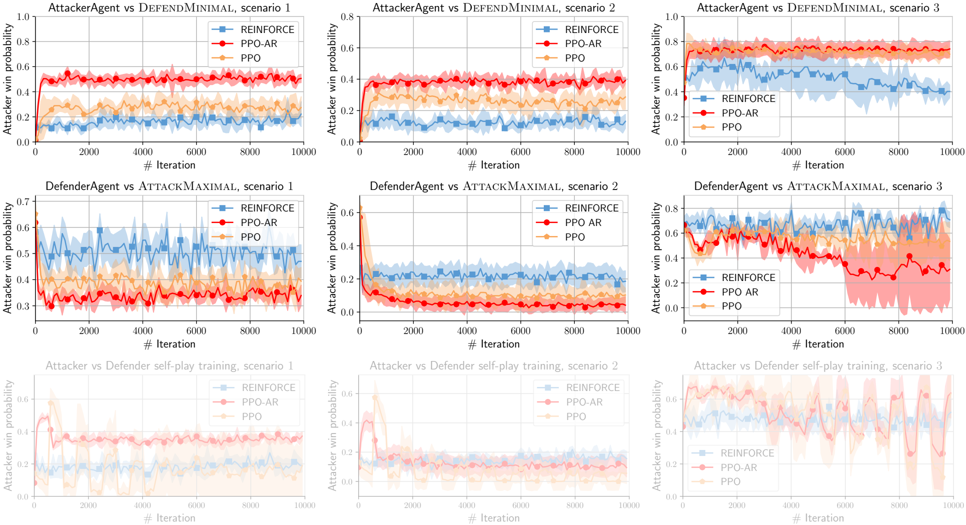


Our Reinforcement Learning Method

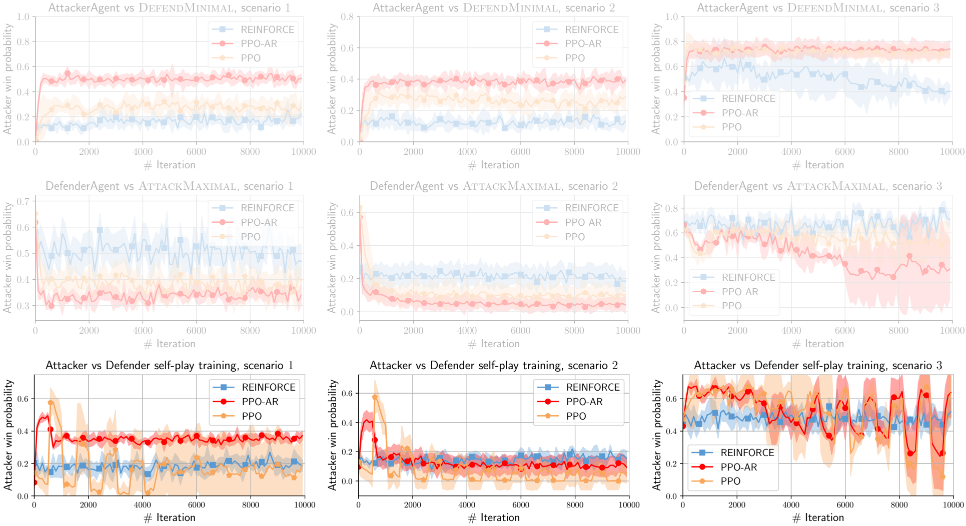
- Policy Gradient & Function Approximation
  - To deal with large state space $\mathcal{S}$
  - π_θ parameterized by weights $\theta \in \mathbb{R}^d$ of NN.
  - PPO & REINFORCE (stochastic π)
- Auto-Regressive Policy Representation
  - To deal with large action space $\mathcal{A}$
  - To minimize interference
  - $\pi(a, n|o) = \pi(a|n, o) \cdot \pi(n|o)$
- **Opponent Pool**
  - To avoid overfitting
  - Want agent to learn a general strategy



Experimentation: Learning from Zero Knowledge



Experimentation: Learning from Zero Knowledge



Conclusion & Future Work

- **Conclusions:**
  - We have proposed a *Method* to automatically find security strategies
  - $\implies$ Model as Markov game & evolve strategies using self-play reinforcement learning
  - Addressed domain-specific challenges with Auto-regressive policy, opponent pool, and function approximation.
  - Challenges of applied reinforcement learning
    - Stable convergence remains a challenge
    - Sample-efficiency is a problem
    - Generalization is a challenge
- **Current & Future Work:**
  - Study techniques for mitigation of identified RL challenges
  - Learn security strategies by interacion with a cyber range

Thank you

- All code for reproducing the results is open source: https://github.com/Limmen/gym-idsgame