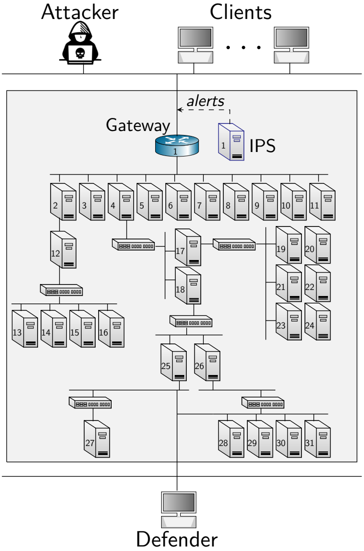
Digital Twins for Security Automation

IEEE/IFIP Network Operations and Management Symposium
8-12 May 2023, Miami FL USA

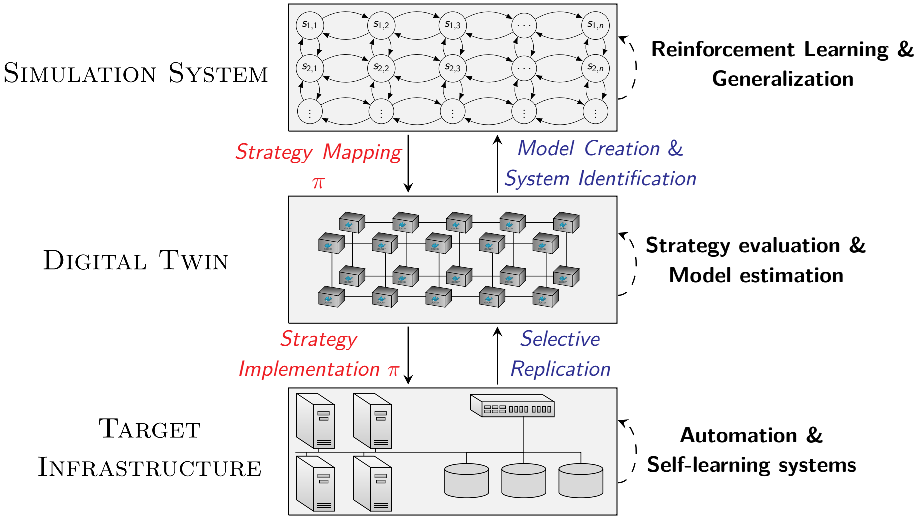
Kim Hammar & Rolf Stadler

Use Case: Intrusion Response

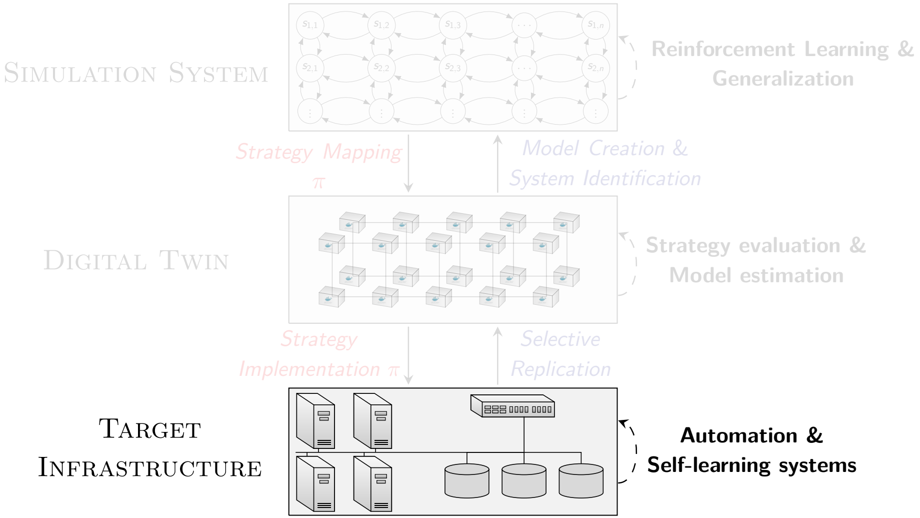
- A **defender** owns an infrastructure
  - Consists of connected components
  - Components run network services
  - Defender defends the infrastructure by monitoring and active defense
  - Has partial observability
- An **attacker** seeks to intrude on the infrastructure
  - Has a partial view of the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



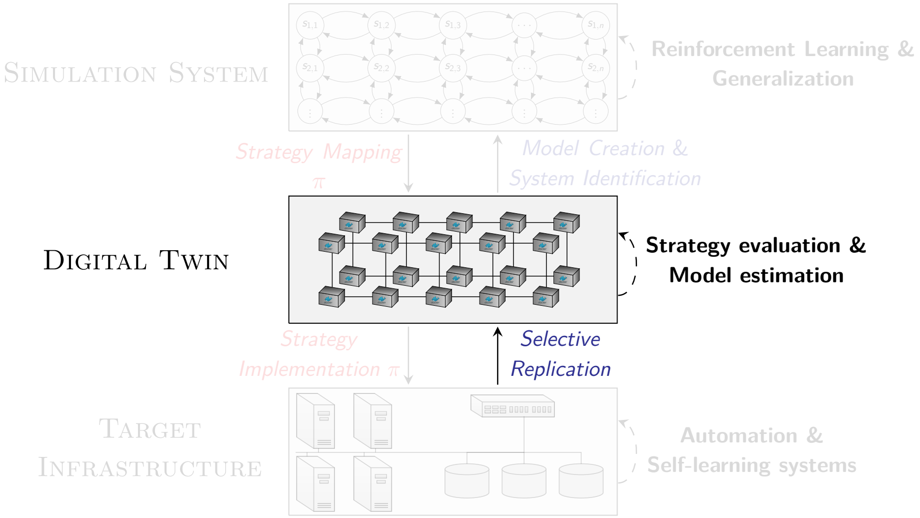
Our Approach for Automated Network Security



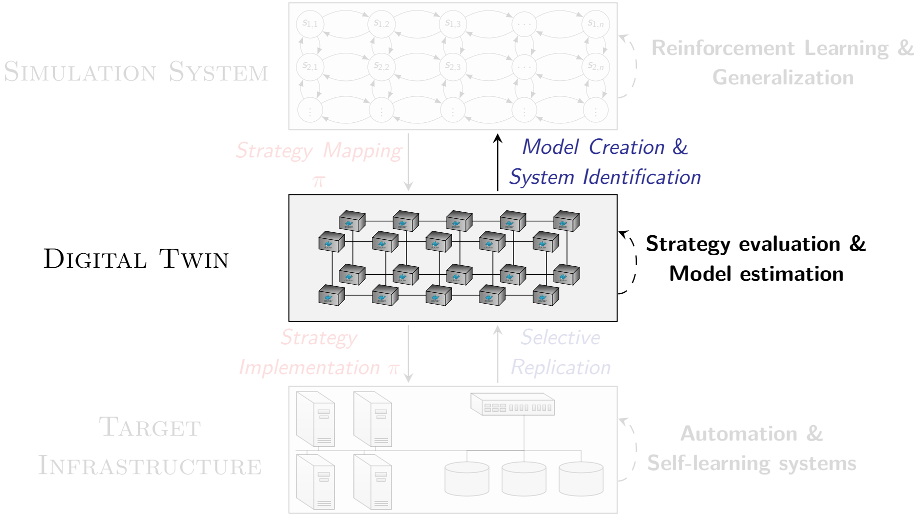
Our Approach for Automated Network Security



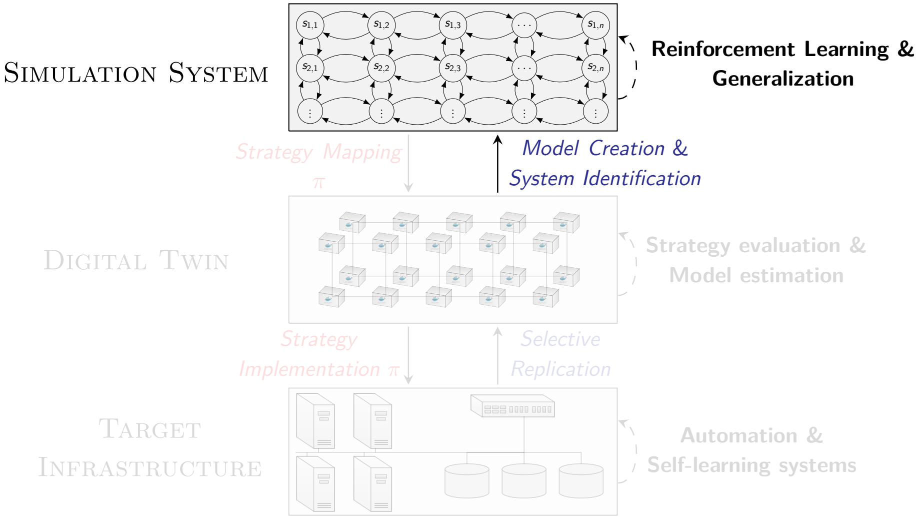
Our Approach for Automated Network Security



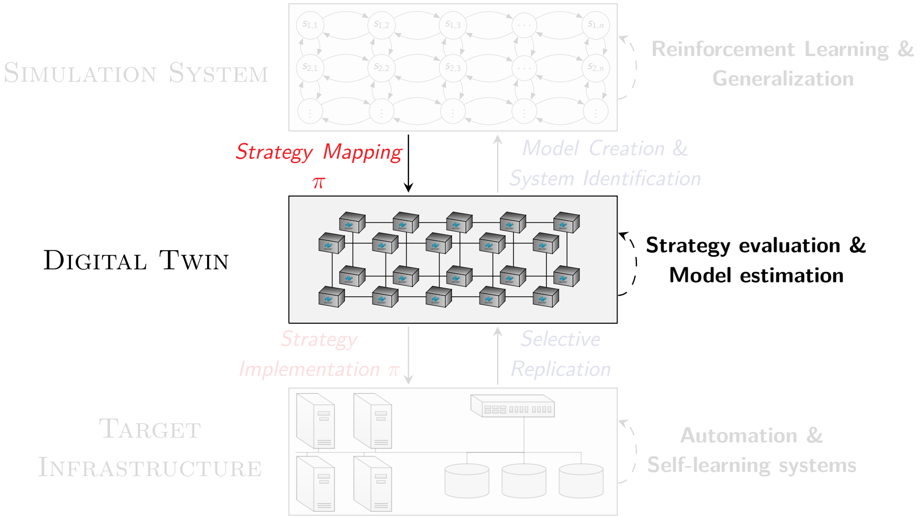
Our Approach for Automated Network Security



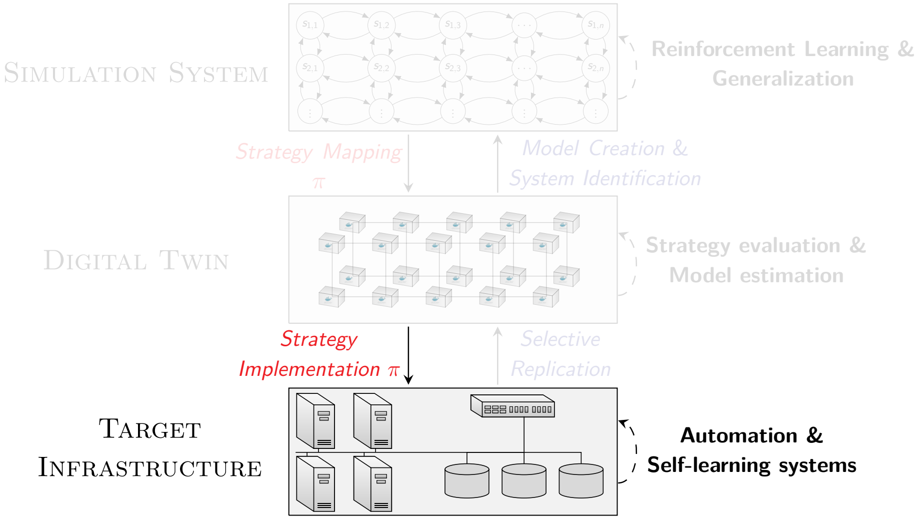
Our Approach for Automated Network Security



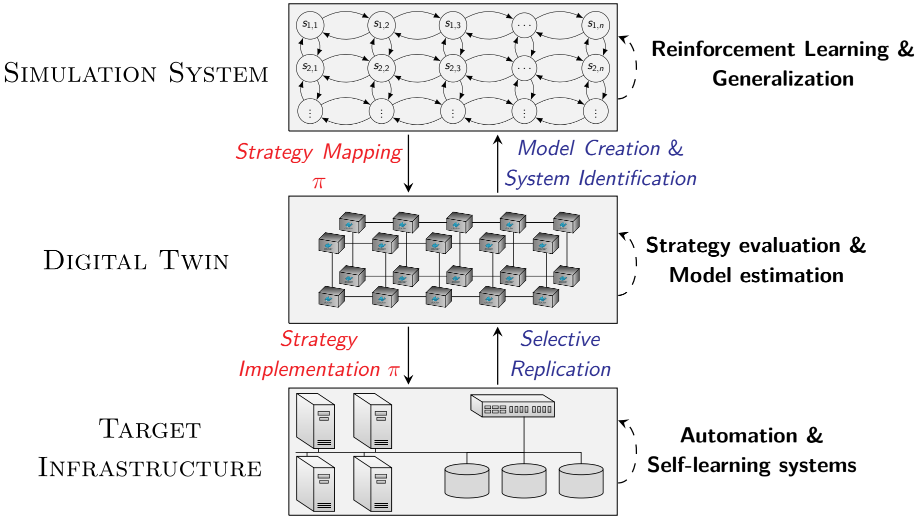
Our Approach for Automated Network Security



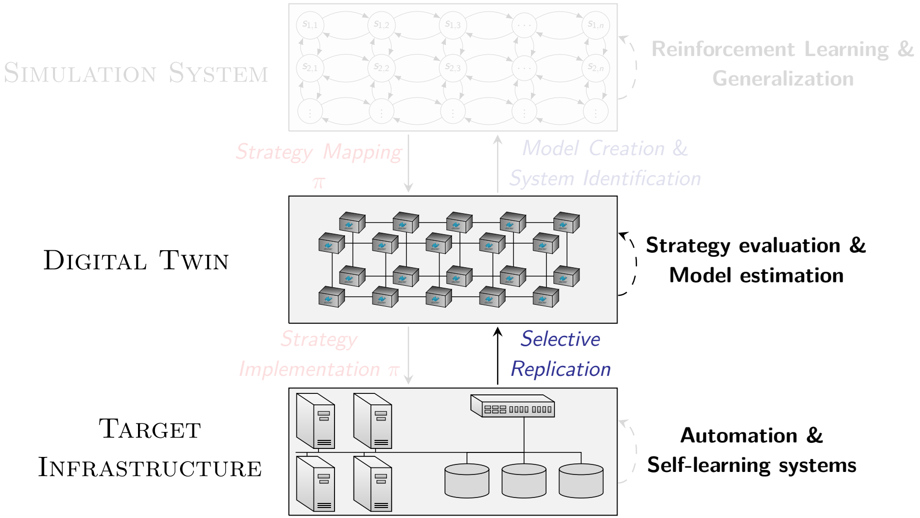
Our Approach for Automated Network Security



Our Approach for Automated Network Security

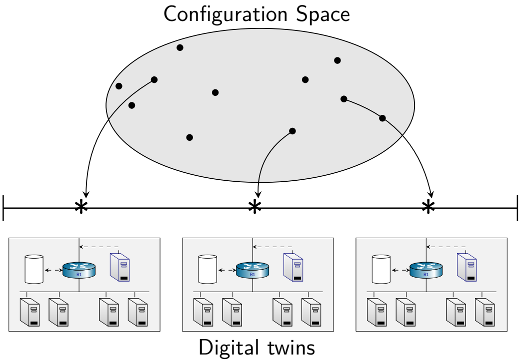


Creating a Digital Twin of the Target Infrastructure



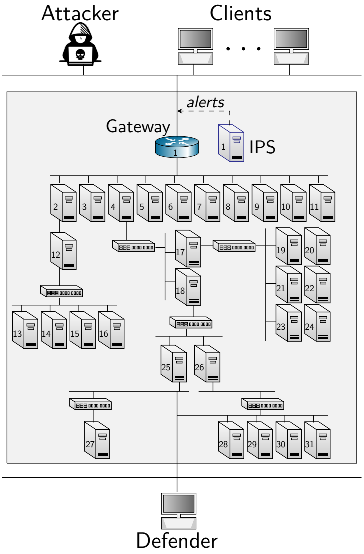
Creating a Digital Twin of the Target Infrastructure

- An infrastructure is defined by its configuration.
- Set of configurations supported by our framework can be seen as a **configuration space**
- The configuration space defines the class of infrastructures for which we can create digital twins.



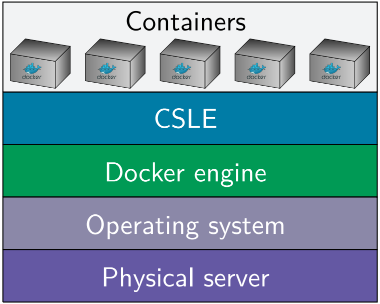
The Target Infrastructure

- 33 **components**
- Topology shown to the right
- Components run network services, e.g. IDPS, SSH, Web, etc.
- A subset of components have vulnerabilities
  - CVE-2017-7494, CVE-2015-3306, CVE-2015-5602
  - CVE-2014-6271, CVE-2016-10033, CVE-2015-1427, etc.
- Clients and the attacker access the infrastructure through the public gateway

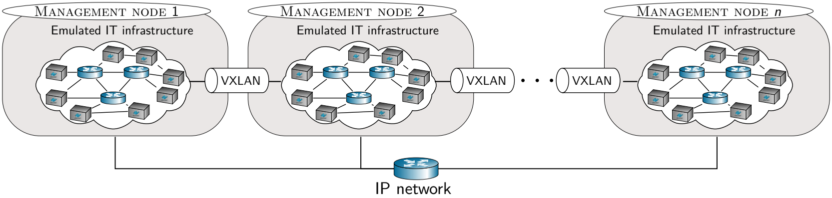


Emulating Physical Components

- We emulate physical components with **Docker containers**
- Focus on linux-based systems
- The containers include everything needed to emulate the host: a runtime system, code, system tools, system libraries, and configurations.
- Examples of containers: IDPS container, client container, attacker container, CVE-2015-1427 container, etc.



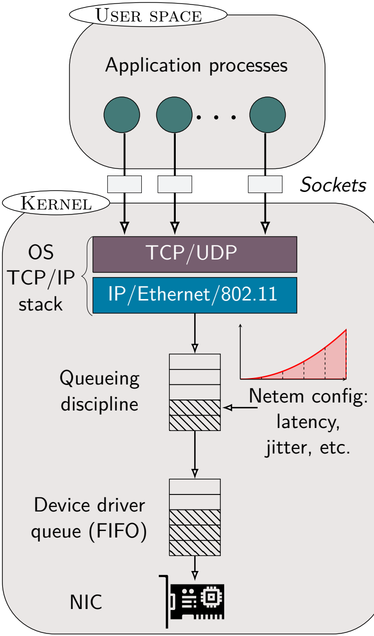
Emulating Network Connectivity



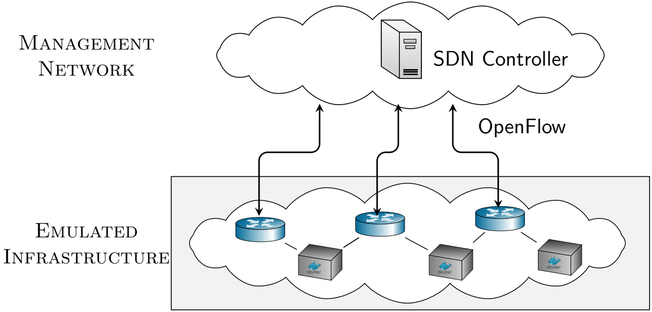
- We emulate network connectivity on the same host using **network namespaces**.
- Connectivity across physical hosts is achieved using **VXLAN tunnels** with Docker swarm.

Emulating Network Conditions

- We do traffic shaping using NetEm in the Linux kernel
- Emulate **internal connections** are full-duplex & loss-less with bit capacities of 1000 Mbit/s
- Emulate **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



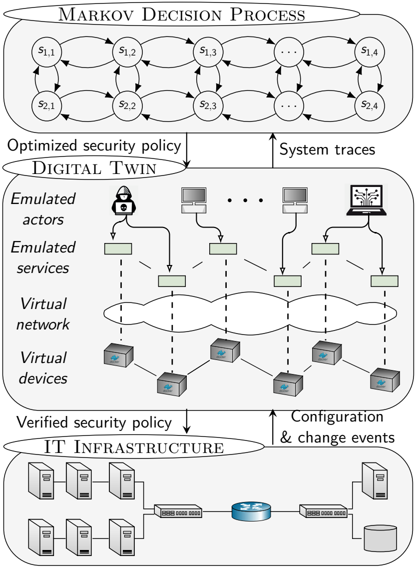
Emulating Physical Switches



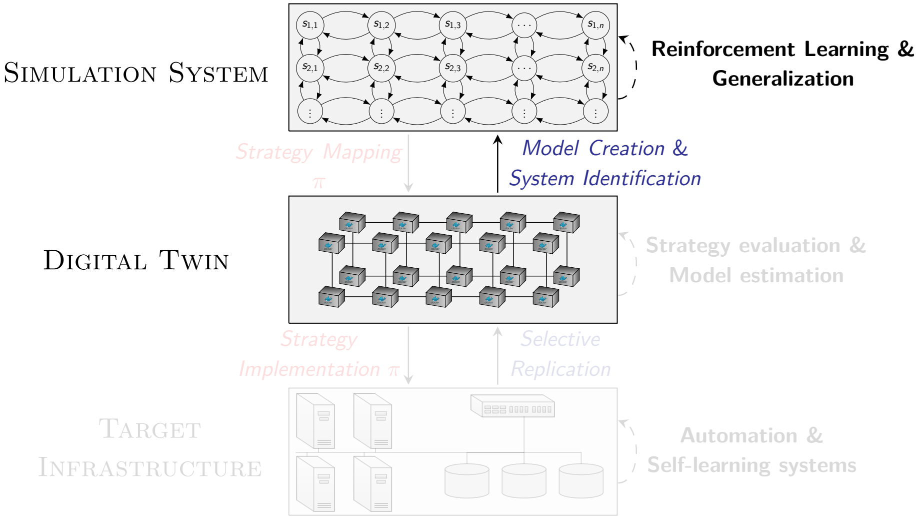
- Physical switches are emulated with Docker containers that run **Open vSwitch (OVS)**
- The emulated switches connect to an SDN controller using the OpenFlow protocol version 1.3 over a secure TLS tunnel
- The **SDN controller is emulated by a container** that resides in the management network.

Emulating Actors

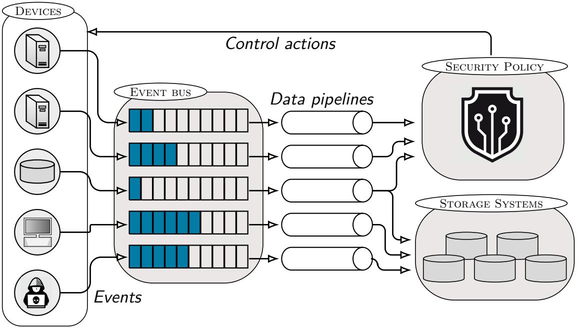
- We emulate **client arrivals with Poisson processes**
- We emulate client interactions with load generators
- Attackers are emulated by automated programs that select actions from a pre-defined set
- Defender actions are emulated through a **custom gRPC API**.



System Identification



Monitoring and Telemetry

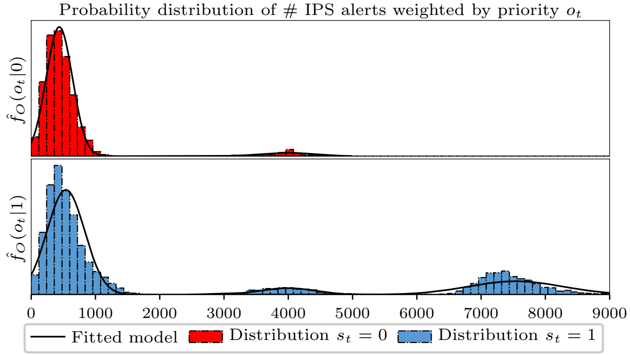


- Emulated devices run monitoring agents that **periodically push metrics to a Kafka event bus**.

- The data in the event bus is consumed by data pipelines that process the data and write to storage systems.

- The processed data is used by an automated security policy to decide on control actions to execute in the digital twin.

Estimating Metric Distributions

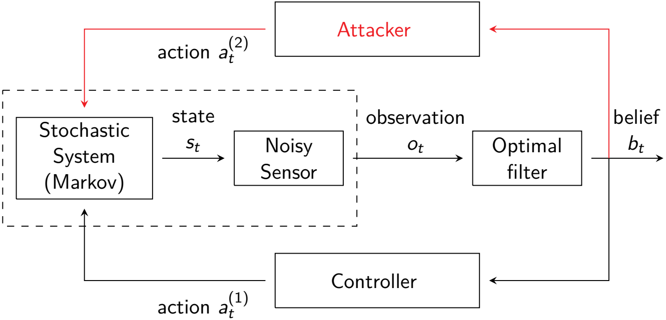


- We use the collected data to **estimate metric distributions**.
- We use the estimated distributions to instantiate Markov games and Markov decision processes.

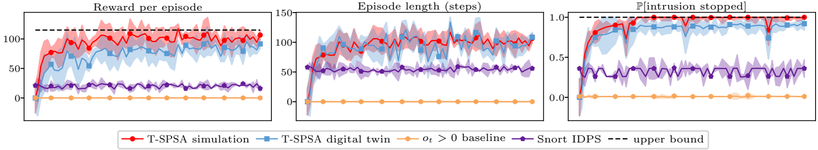
Learning Security Strategies

- We model the evolution of the system with a **discrete-time dynamical system**.
- We assume a Markovian system with stochastic dynamics and partial observability.

- A Partially Observed Markov Decision Process (POMDP)
  - If attacker is static.
- A Partially Observed Stochastic Game (POSG)
  - If attacker is dynamic.



Learning Security Strategies



- T-SPSA is our reinforcement learning algorithm
- T-SPSA outperforms Snort and converges to **near-optimal** strategies
- While the performance is slightly better in simulation than in the digital twin, it is clear that the performance in the two environments are correlated.

For more details about the theory

- *Finding Effective Security Strategies through Reinforcement Learning and Self-Play*[1]
- *Learning Intrusion Prevention Policies through Optimal Stopping*[2]
- *A System for Interactive Examination of Learned Security Policies*[3]
- *Intrusion Prevention Through Optimal Stopping*[4]
- *Learning Security Strategies through Game Play and Optimal Stopping*[5]
- *An Online Framework for Adapting Security Policies in Dynamic IT Environments*[6]
- *Learning Near-Optimal Intrusion Responses Against Dynamic Attackers*[7]

---

[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)*. Izmir, Turkey, 2020.

[2] Kim Hammar and Rolf Stadler. "Learning Intrusion Prevention Policies through Optimal Stopping". In: *International Conference on Network and Service Management (CNSM 2021)*. http://dl.ifip.org/db/conf/cnsm/cnsm2021/1570732932.pdf. Izmir, Turkey, 2021.

[3] Kim Hammar and Rolf Stadler. "A System for Interactive Examination of Learned Security Policies". In: *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. 2022, pp. 1–3. DOI: 10.1109/NOMS54207.2022.9789707.

[4] Kim Hammar and Rolf Stadler. "Intrusion Prevention Through Optimal Stopping". In: *IEEE Transactions on Network and Service Management* 19.3 (2022), pp. 2333–2348. DOI: 10.1109/TNSM.2022.3176781.

[5] Kim Hammar and Rolf Stadler. "Learning Security Strategies through Game Play and Optimal Stopping". In: *Proceedings of the ML4Cyber workshop, ICML 2022, Baltimore, USA, July 17-23, 2022*. PMLR, 2022.

[6] Kim Hammar and Rolf Stadler. "An Online Framework for Adapting Security Policies in Dynamic IT Environments". In: *International Conference on Network and Service Management (CNSM 2022)*. Thessaloniki, Greece, 2022.

[7] Kim Hammar and Rolf Stadler. *Learning Near-Optimal Intrusion Responses Against Dynamic Attackers*. 2023. DOI: 10.48550/ARXIV.2301.06085. URL: https://arxiv.org/abs/2301.06085.

Conclusions

- We develop a framework for **automated security**.
- Our framework centers around a digital twin
- We use the digital twin to optimize security strategies through reinforcement learning, game theory, and control theory.
- Documentation of our framework: `limmen.dev/csle`.