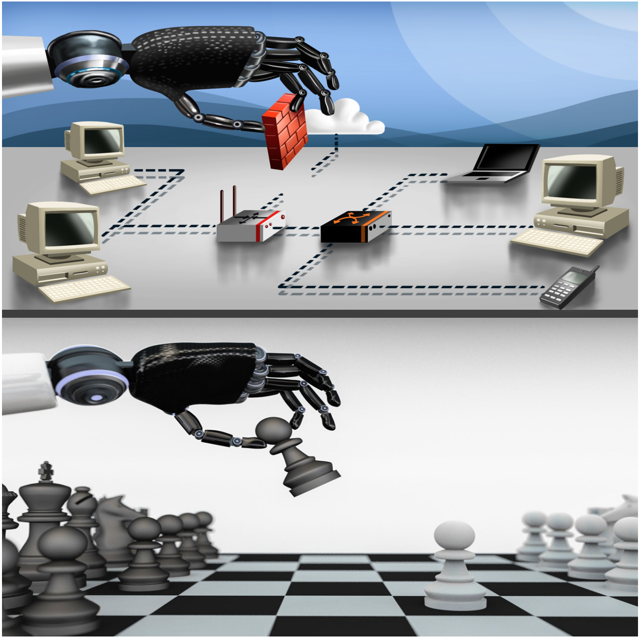
Automated Intrusion Response

Kim Hammar

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

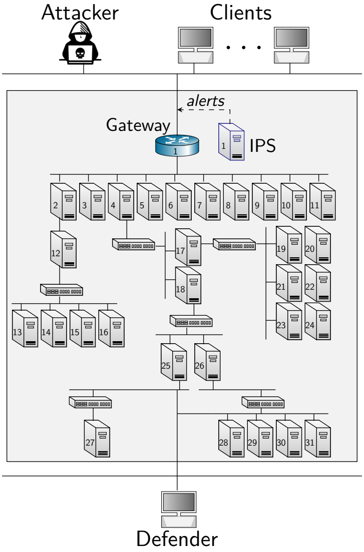
May 31, 2024

Automated Intrusion Response

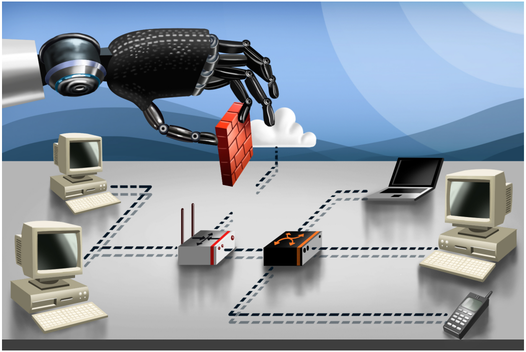


Use Case: Intrusion Response

- 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

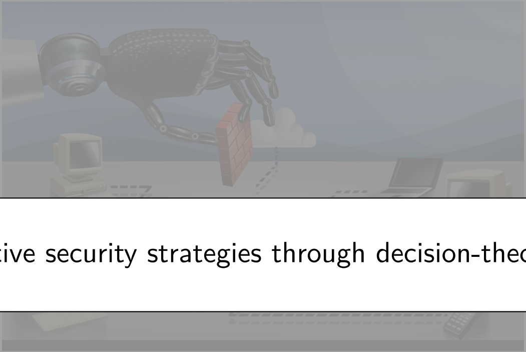
No automation.
Manual detection.
Manual prevention.
No alerts.
No automatic responses.
Lack of tools.

Operator assistance.
Manual detection.
Manual prevention.
Audit logs.
Security tools.

Partial automation.
System has automated functions for detection/prevention but requires manual updating and configuration.
Intrusion detection systems.
Intrusion prevention systems.

High automation.
System automatically updates itself.
Automated attack detection.
Automated attack mitigation.

Automated Intrusion Response



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

Levels of security automation

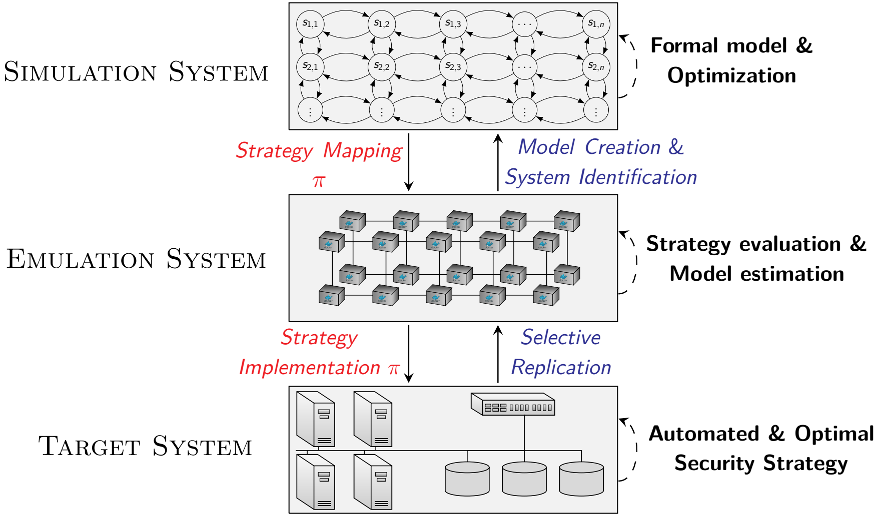
No automation.
Manual detection.
Manual prevention.
No alerts.
No automatic responses.
Lack of tools.

Operator assistance.
Manual detection.
Manual prevention.
Audit logs.
Security tools.

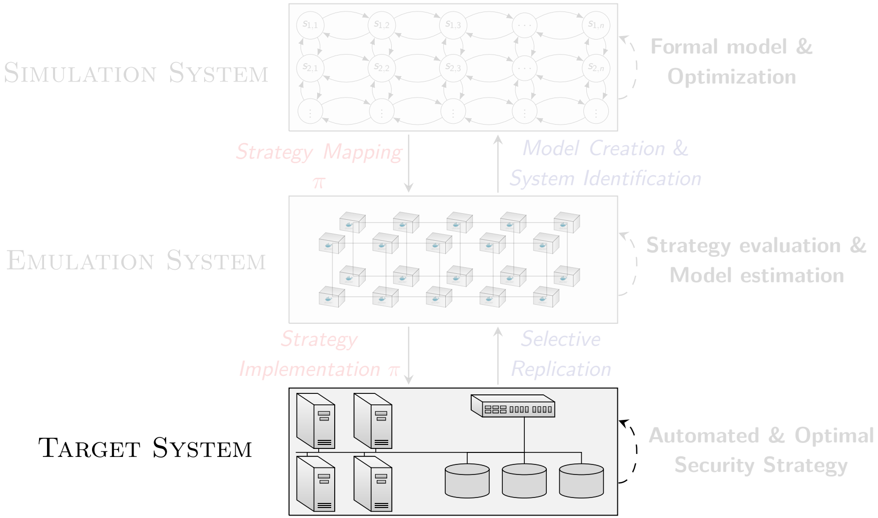
Partial automation.
System has automated functions for detection/prevention but requires manual updating and configuration.
Intrusion detection systems.
Intrusion prevention systems.

High automation.
System automatically updates itself.
Automated attack detection.
Automated attack mitigation.

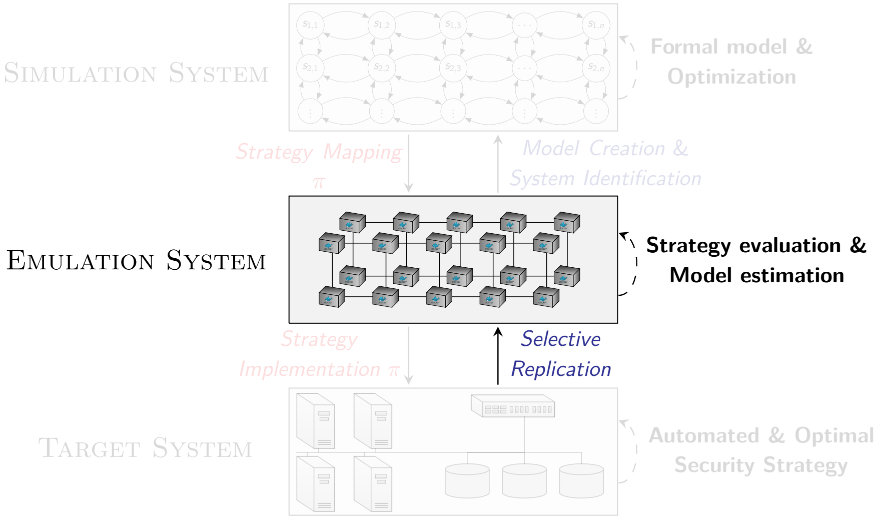
Our Framework for Automated Intrusion Response



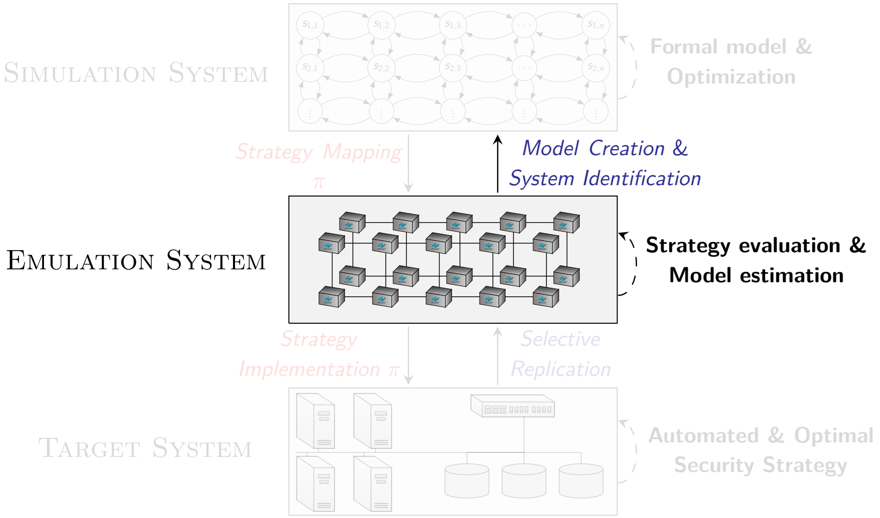
Our Framework for Automated Intrusion Response



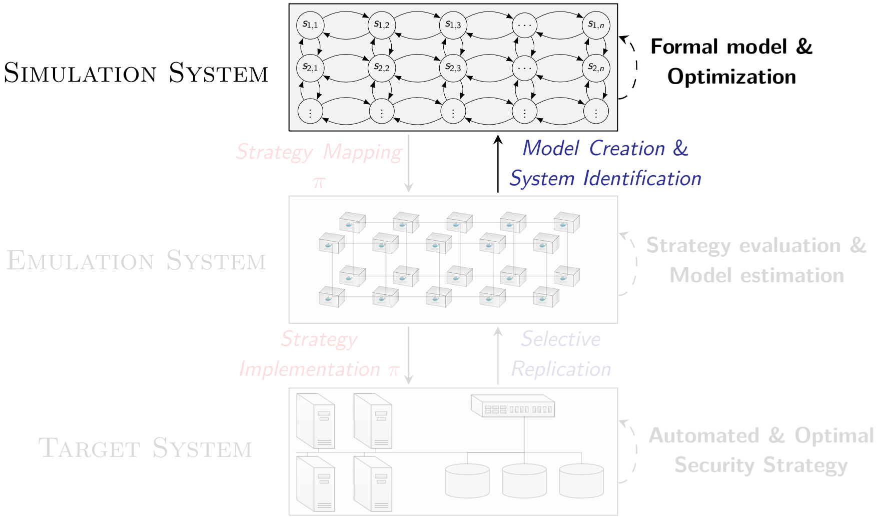
Our Framework for Automated Intrusion Response



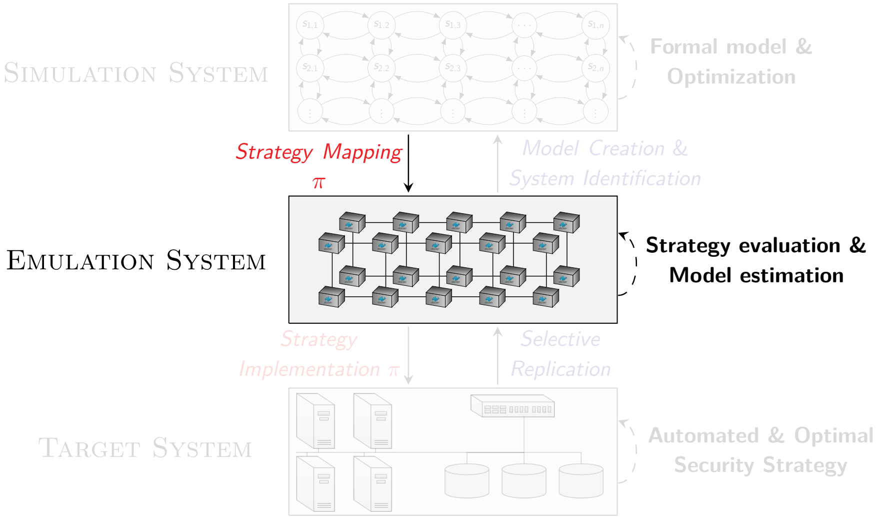
Our Framework for Automated Intrusion Response



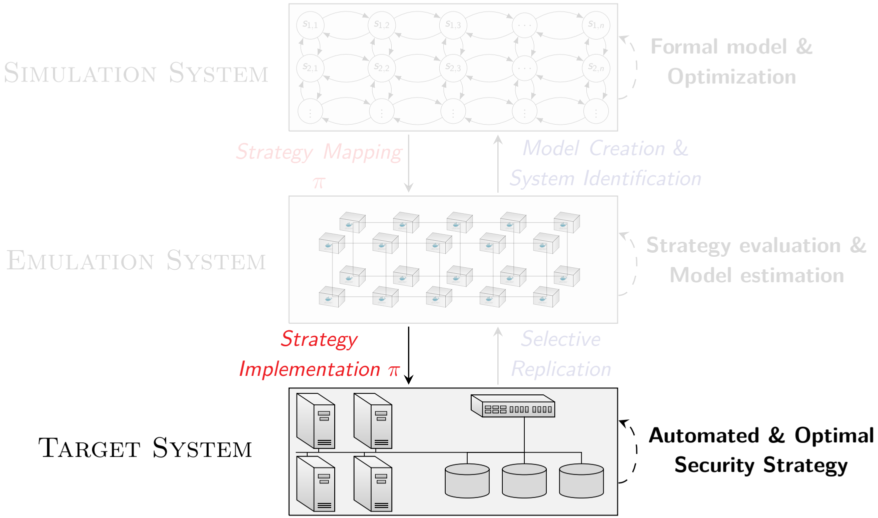
Our Framework for Automated Intrusion Response



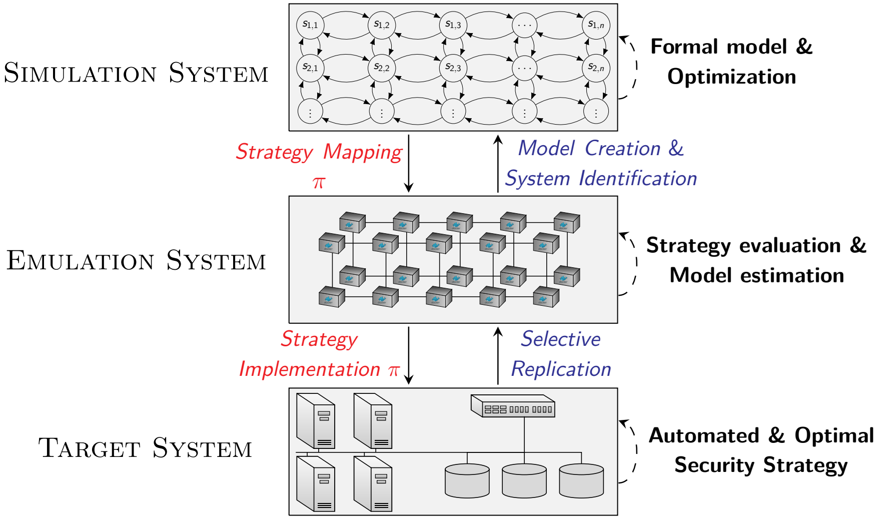
Our Framework for Automated Intrusion Response



Our Framework for Automated Intrusion Response

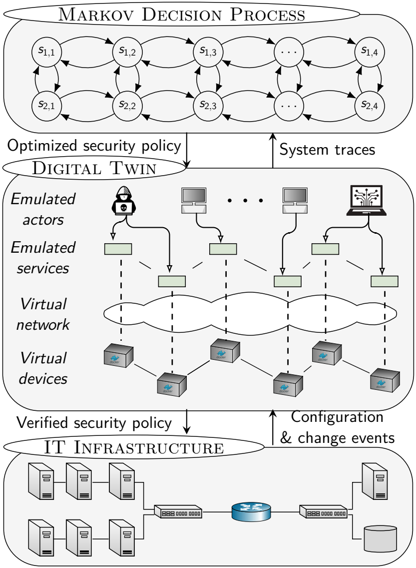


Our Framework for Automated Intrusion Response

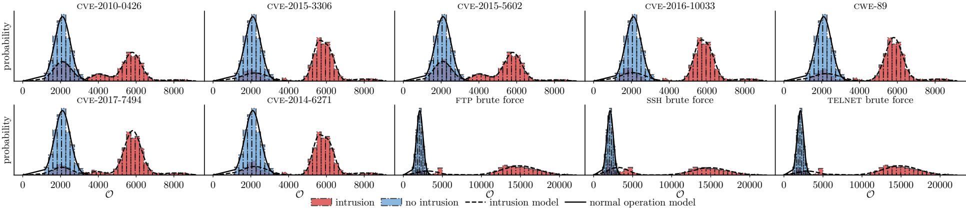


Step 1: Emulation

- Emulate servers using **virtual containers**.
- Emulate connectivity using virtual networks.
- Emulate clients using traffic generators.
- Emulate attacker/defender using **automation API**.
- Source code: `https://github.com/Limmen/csle`



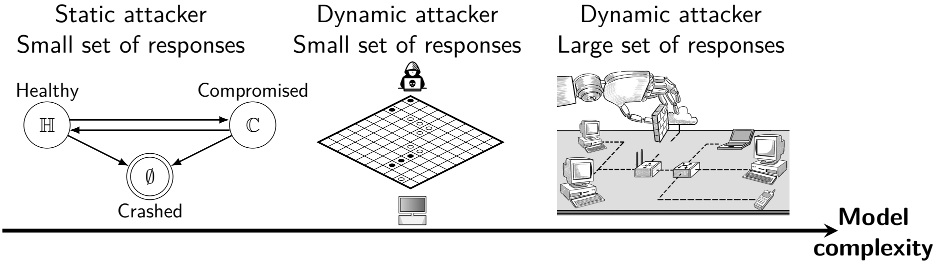
Step 2: Data Collection



Distributions of IDS alarms during different types of intrusions.

- The first step in our framework is to collect data from the emulation system.
- We collect data both during normal operation and during attacks.

Step 3: Modeling

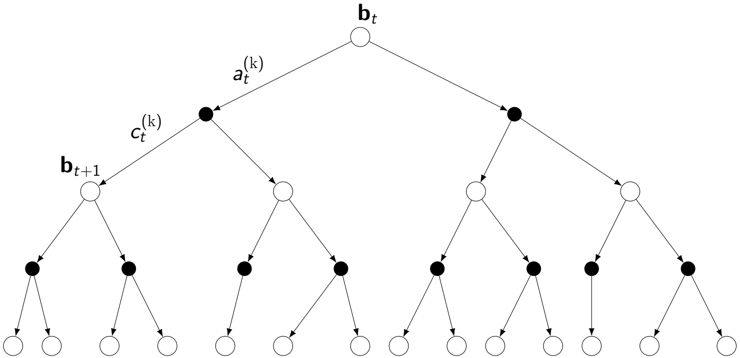


- Intrusion response can be **modeled in many ways**
  - As a *parametric optimization problem*
  - As an *optimal stopping problem*
  - As a *dynamic program*
  - As a *game*
  - etc.

Step 4: Optimization

- Different optimization techniques:
  - Dynamic programming
  - Reinforcement learning
  - Stochastic approximation
  - Regret minimization
  - Evolutionary computation
  - etc.

Lookahead tree.



Conclusions

- We develop a *framework* to automatically learn security strategies.
- We apply the framework to an **intrusion response use case**.
- References and videos are available at: https://www.kth.se/cdis