Automated Security with a Foundation Model

Visit to the City University of Hong Kong

October 20, 2025

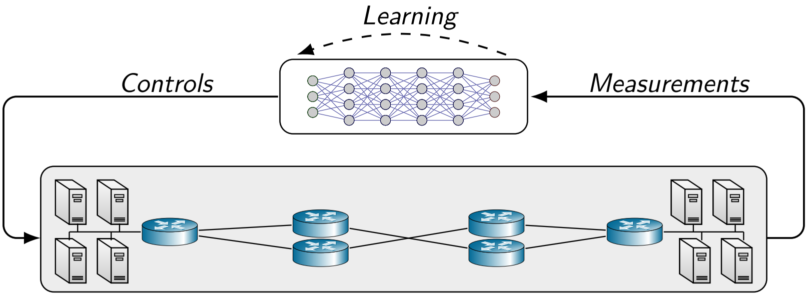
Dr. Kim Hammar

kim.hammar@unimelb.edu.au

POSTERA CRESCAM LAUDE

THE UNIVERSITY OF MELBOURNE

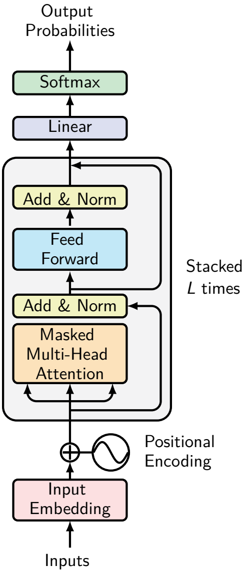
Next Generation of Security Systems



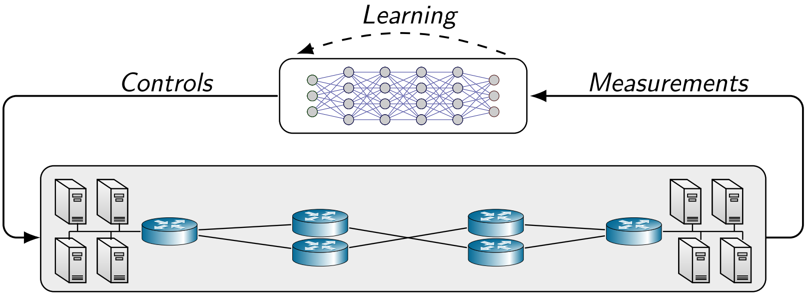
- What role will **foundation models** play in the next generation of security systems?

Different Types of **Foundation Models**

- Based on the transformer architecture.
- Trained on **vast datasets**.
- Billions of parameters.
- Examples:
  - Large language models (e.g., DeepSeek).
  - Time series models (e.g., Chronos).
  - Speech and audio models (e.g., Whisper).
  - Multi-modal models (e.g., Sora).

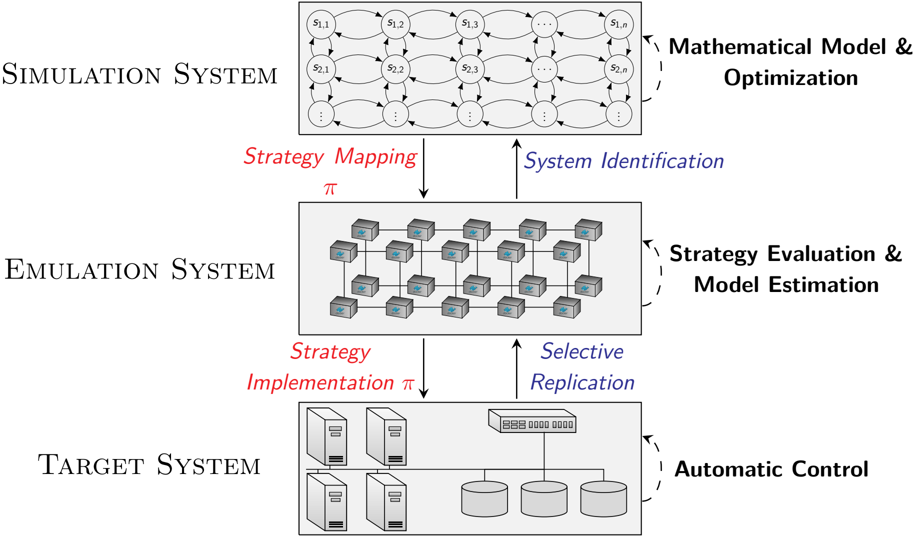


Autonomous Security Systems

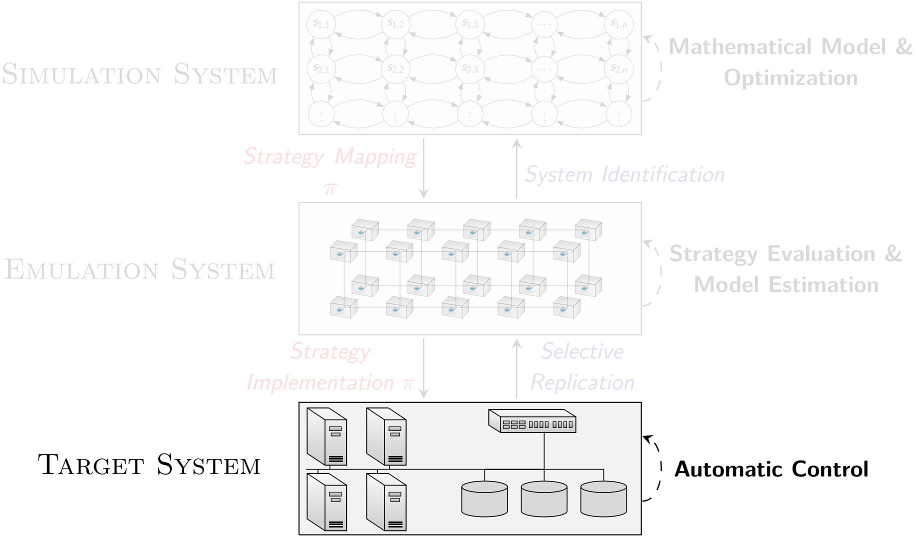


- Systems with **high automation** that adapt and learn.
- Responds to threats and incidents autonomously.
- Longstanding goal in network and systems engineering.

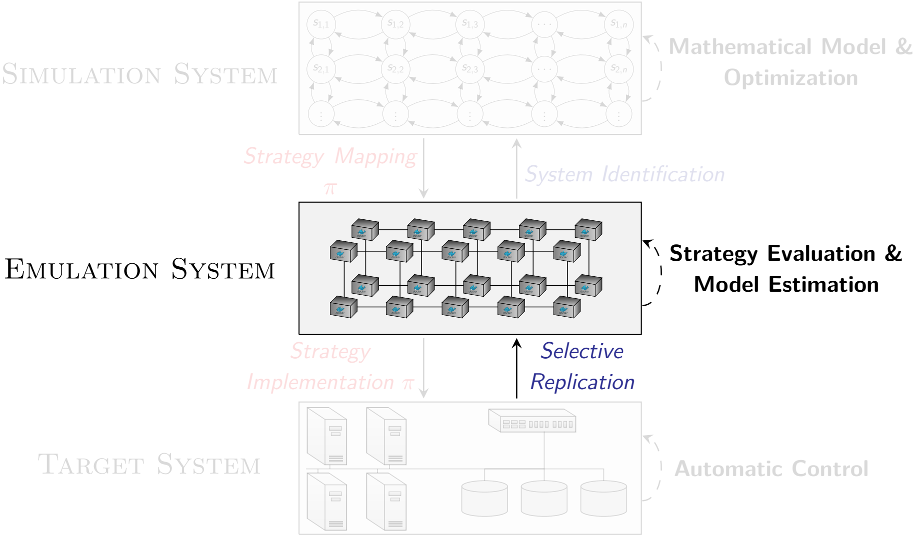
Methodology for Building Autonomous Security Systems



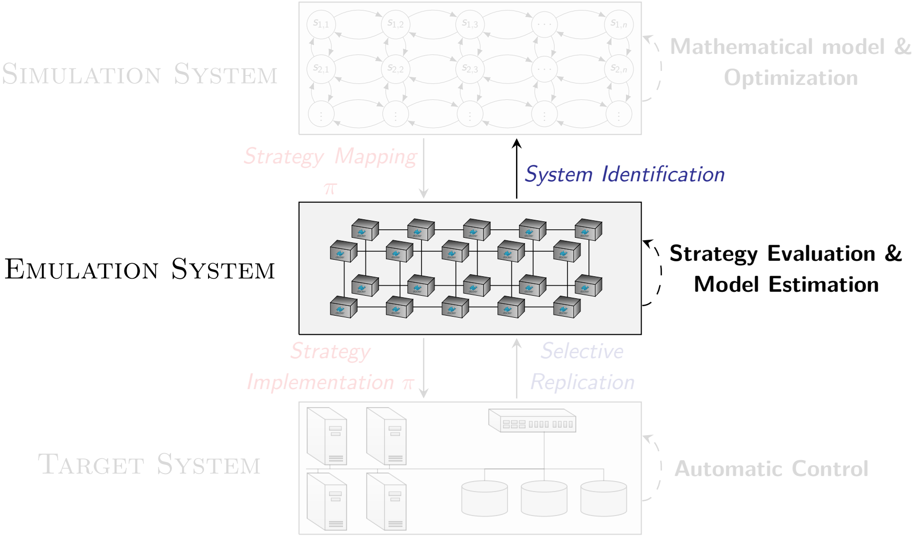
Methodology for Building Autonomous Security Systems



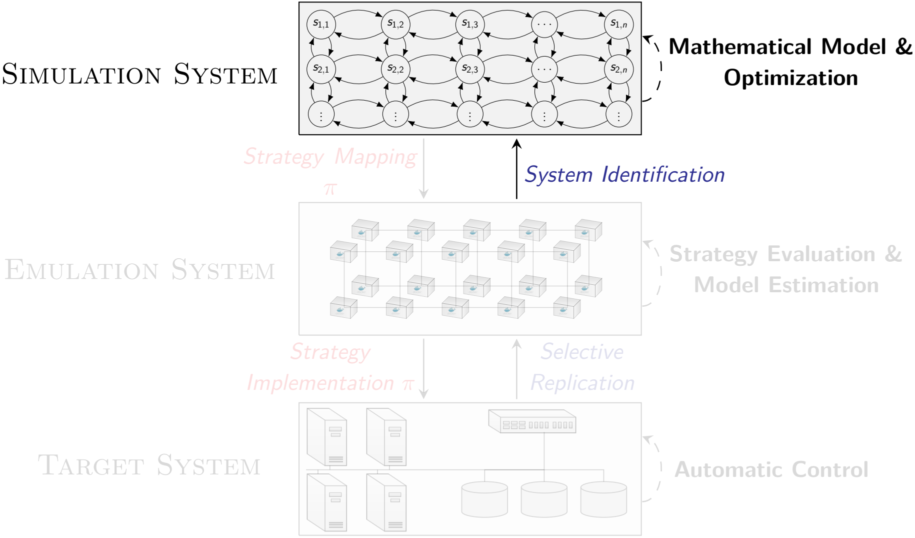
Methodology for Building Autonomous Security Systems



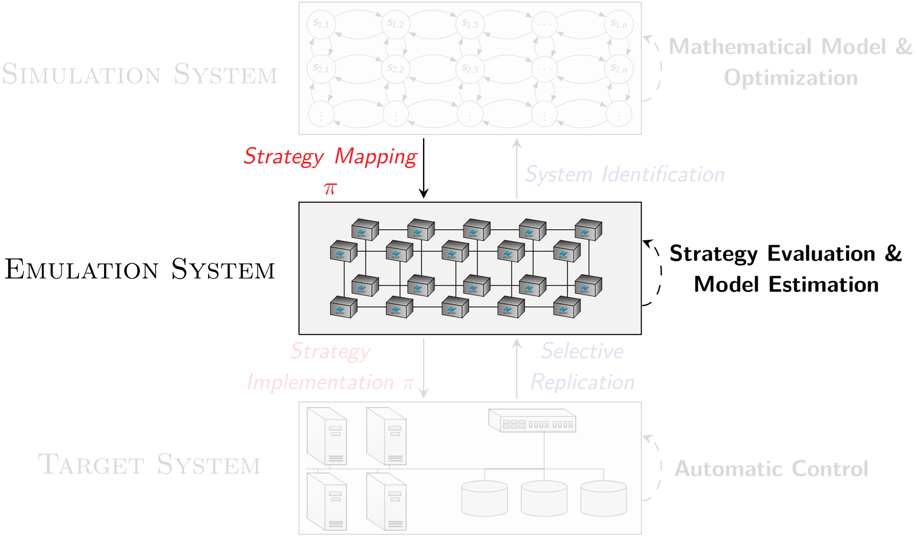
Methodology for Building Autonomous Security Systems



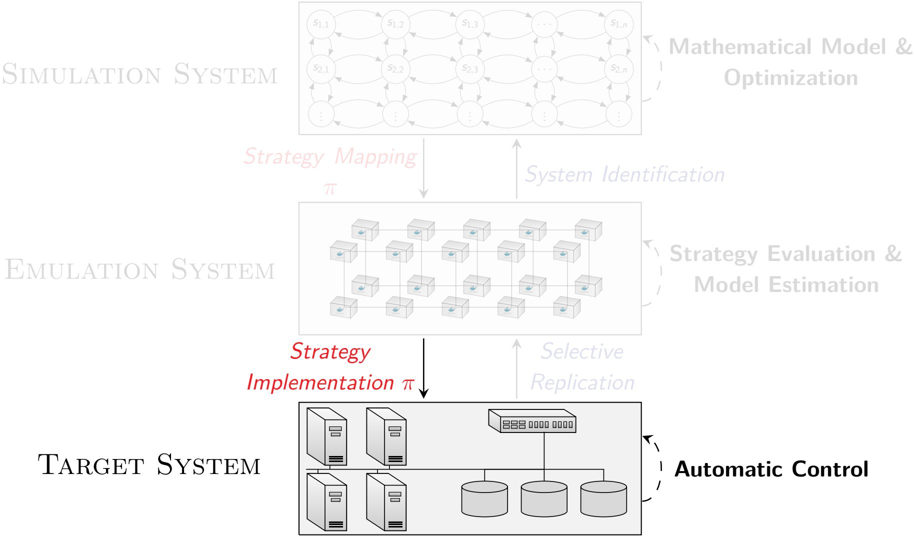
Methodology for Building Autonomous Security Systems



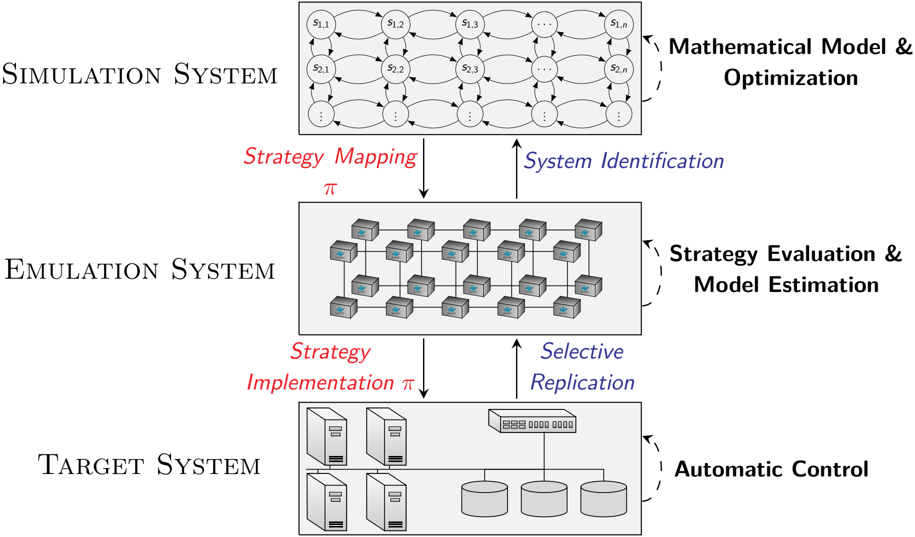
Methodology for Building Autonomous Security Systems



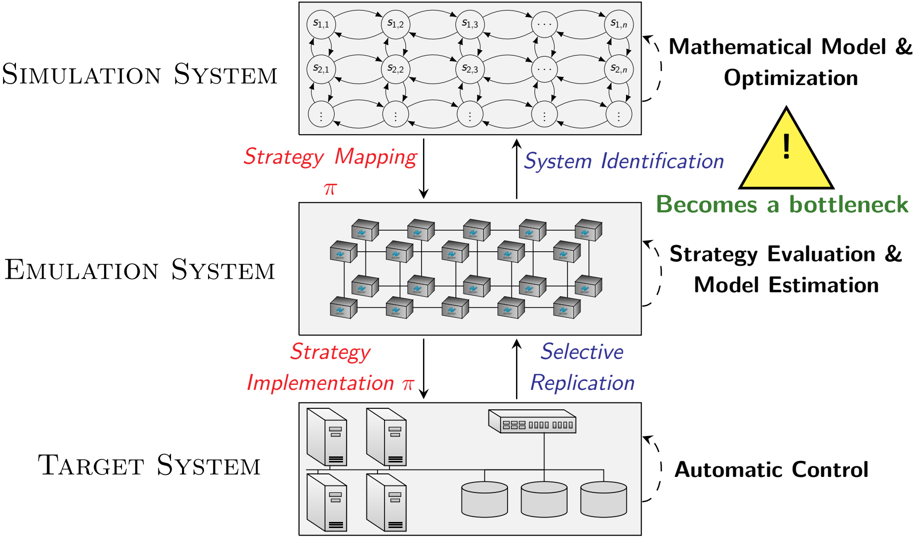
Methodology for Building Autonomous Security Systems



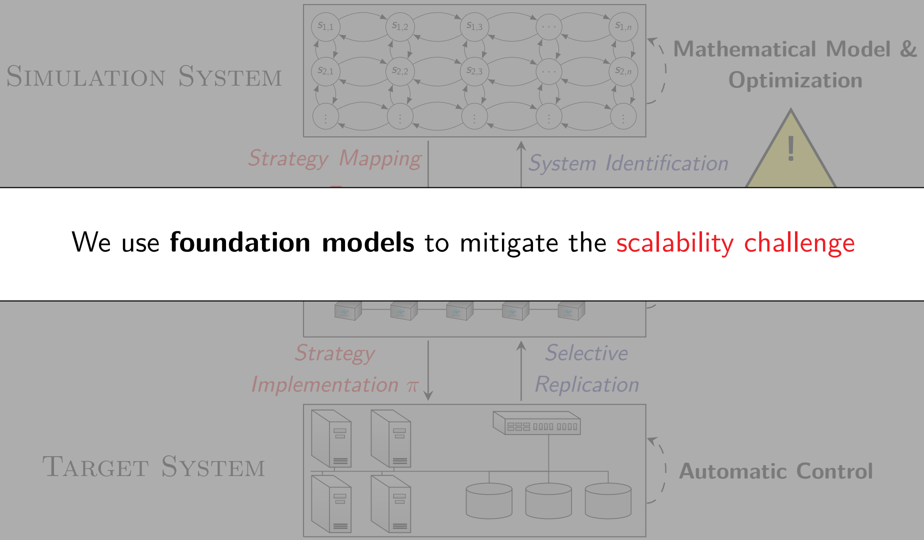
Methodology for Building Autonomous Security Systems



Methodology for Building Autonomous Security Systems



Methodology for Building Autonomous Security Systems



We use **foundation models** to mitigate the scalability challenge

Outline

- **Automated security with a foundation model.**
  - *Overview of our framework.*
- **Theoretical analysis.**
  - *Controlling the hallucination bound.*
  - *Regret bound.*
- **Case study: Incident Response.**
  - *Comparison with frontier models.*

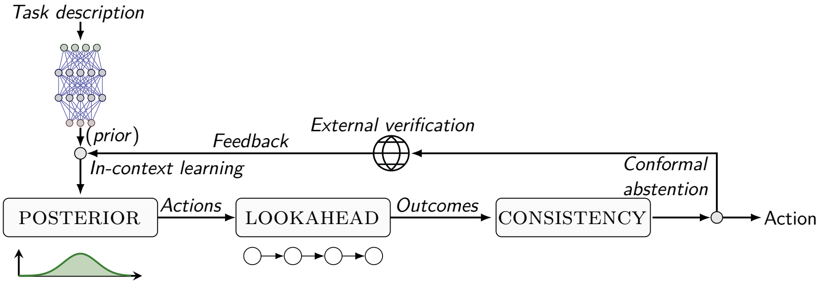
Outline

- **Automated security with a foundation model**
  - *Overview of our framework.*
- **Theoretical analysis**
  - *Controlling the hallucination bound.*
  - *Regret bound.*
- **Case study: Incident Response**
  - *Comparison with frontier models.*

Outline

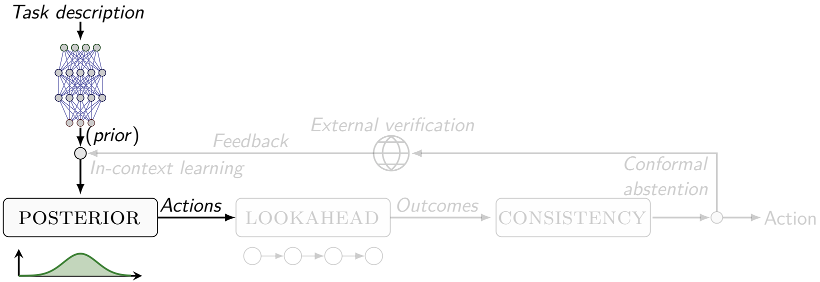
- **Automated security with a foundation model**
  - *Overview of our framework.*
- **Theoretical analysis**
  - *Controlling the hallucination bound.*
  - *Regret bound.*
- **Case study: Incident Response**
  - *Comparison with frontier models.*

Automated Security with a **Foundation Model**



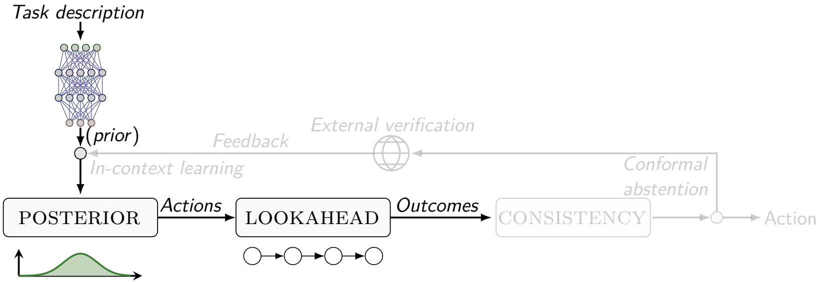
- We use the **model to generate candidate actions.**
- We evaluate actions through lookahead.
- We detect likely hallucinations by evaluating consistency.
- Abstain from actions with low consistency.
- Refine actions via **in-context learning** from feedback.

Automated Security with a **Foundation Model**



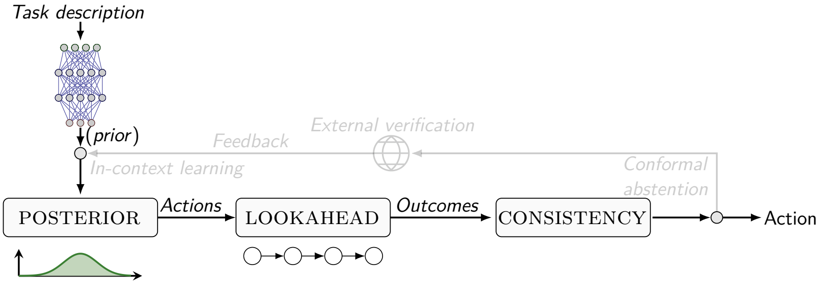
- We use the **model to generate candidate actions.**
- We evaluate actions through lookahead.
- We detect likely hallucinations by evaluating consistency.
- Abstain from actions with low consistency.
- Refine actions via **in-context learning** from feedback.

Automated Security with a **Foundation Model**



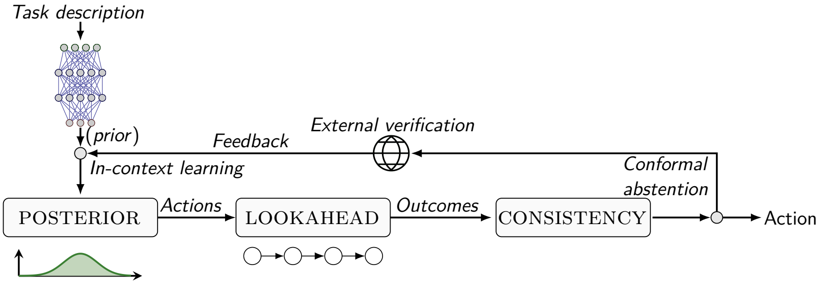
- We use the **model to generate candidate actions.**
- We evaluate actions through lookahead.
- We detect likely hallucinations by evaluating consistency.
- Abstain from actions with low consistency.
- Refine actions via **in-context learning** from feedback.

Automated Security with a **Foundation Model**



- We use the **model to generate candidate actions.**
- We evaluate actions through lookahead.
- We detect likely hallucinations by evaluating consistency.
- Abstain from actions with low consistency.
- Refine actions via **in-context learning** from feedback.

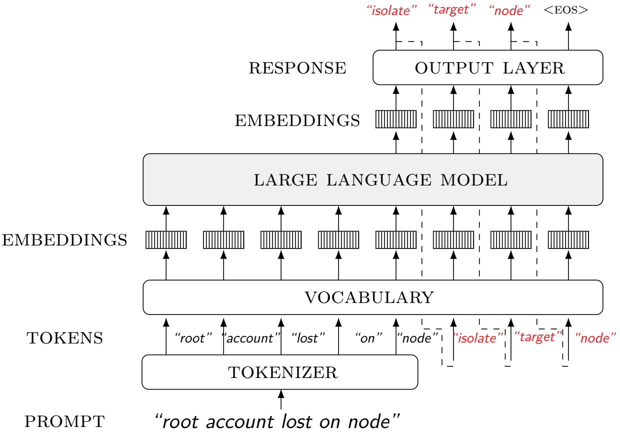
Automated Security with a **Foundation Model**



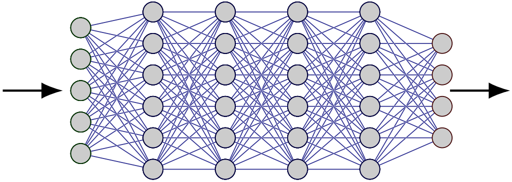
- We use the **model to generate candidate actions.**
- We evaluate actions through lookahead.
- We detect likely hallucinations by evaluating consistency.
- Abstain from actions with low consistency.
- Refine actions via **in-context learning** from feedback.

Generating Candidate Actions

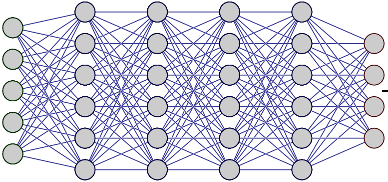
- Generate N candidate actions via **auto-regressive sampling.**
- Can think of the LLM as a base strategy.



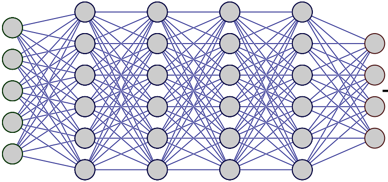
Lookahead Simulation with the LLM

$\mathbf{s}_0$  $\mathbf{a}_0$

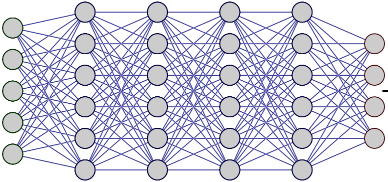
Lookahead Simulation with the LLM

$\mathbf{s}_0, \mathbf{a}_0 \longrightarrow$  $\longrightarrow \mathbf{s}_1$

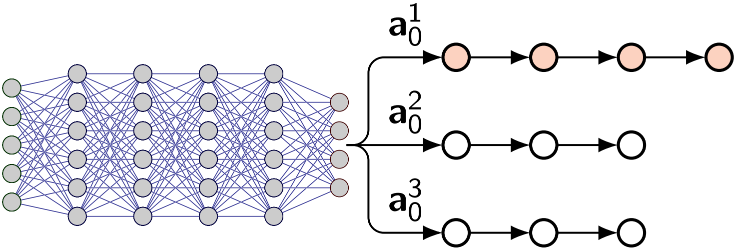
Lookahead Simulation with the LLM

$\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1 \longrightarrow$  $\longrightarrow \mathbf{a}_1$

Lookahead Simulation with the LLM

$\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1 \longrightarrow$  $\longrightarrow \mathbf{s}_2$

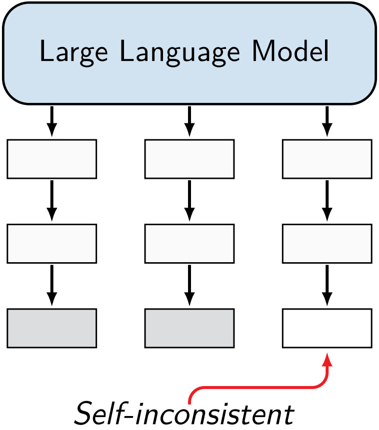
Lookahead Simulation with the LLM



- For each candidate action $\mathbf{a}_t^i$, we use the LLM to predict the subsequent states and actions.
- We select the action with the best outcome.

Evaluating the **Consistency** of Actions

- We use **inconsistency** as an indication of hallucination.



Abstaining from Inconsistent Actions

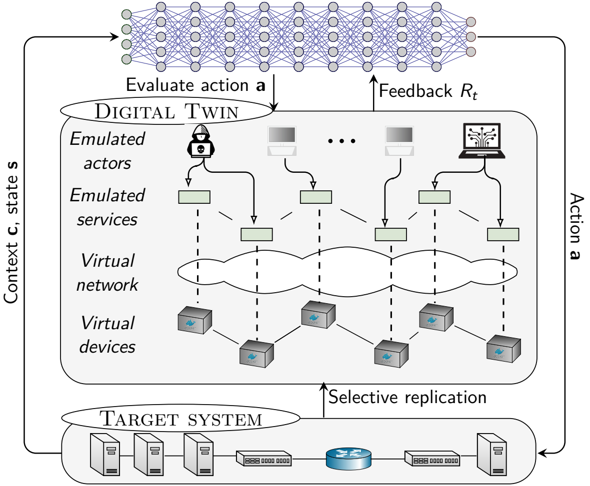
- Let $\lambda(\mathbf{a}) \in [0, 1]$ be a function that evaluates the consistency of a given action $\mathbf{a}$.
- We use this function to abstain from actions with low consistency, as expressed by the following decision rule:

$$\rho_\gamma(\mathbf{a}_t) = \begin{cases} 1 \text{ (abstain)}, & \text{if } \lambda(\mathbf{a}_t) \leq \gamma, \\ 0 \text{ (not abstain)}, & \text{if } \lambda(\mathbf{a}_t) > \gamma, \end{cases}$$

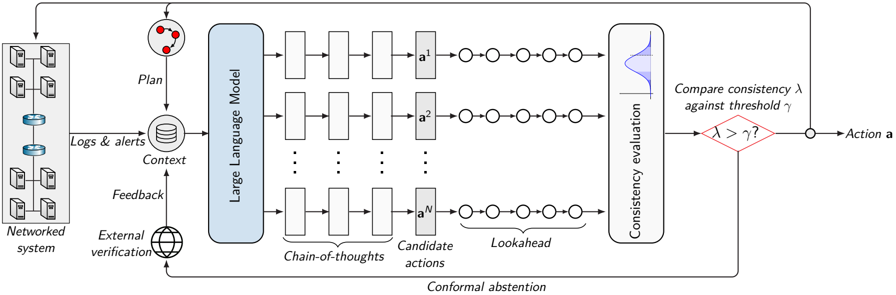
where $\gamma \in [0, 1]$ is a **consistency threshold**.

In-Context Learning from Feedback

If an action does not meet the **consistency threshold**, we abstain from it, collect external feedback (e.g., from a digital twin), and select a new action through in-context learning.



Summary of Our Framework



Outline

- **Automated security with a foundation model**
  - *Overview of our framework.*
- **Theoretical analysis**
  - *Controlling the hallucination bound.*
  - *Regret bound.*
- **Case study: Incident Response**
  - *Comparison with frontier models.*

Conformal Abstention

Let $\{\mathbf{a}_i\}_{i=1}^n$ be a *calibration dataset* of hallucinated actions.

Proposition 1

- *Assume the actions in the calibration dataset* $\{\mathbf{a}_i\}_{i=1}^n$ *are i.i.d.*
- *Let* $\tilde{\mathbf{a}}$ *be an hallucinated action from the same distribution.*
- *Let* $\kappa \in (0, 1]$ *be a desirable upper bound on the hallucination probability.*

Define the threshold

$$\tilde{\gamma} = \inf \left\{ \gamma \;\middle|\; \frac{|\{i \mid \lambda(\mathbf{a}_i) \leq \gamma\}|}{n} \geq \frac{\lceil (n+1)(1-\kappa) \rceil}{n} \right\},$$

where $\lceil \cdot \rceil$ *is the ceiling function. We have*

$$P\left(\textit{not abstain from } \tilde{\mathbf{a}}\right) \leq \kappa.$$

Regret Bound for In-Context Learning

Proposition 2 (Informal)

- *Let $\mathcal{R}_K$ denote the* ***Bayesian regret.***
- *Assume that the LLM's output distribution is aligned with the posterior given the context.*
- *Assume bandit feedback.*

We have

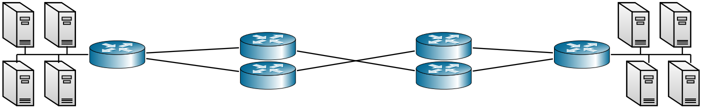
$$\mathcal{R}_K \leq C\sqrt{|\mathcal{A}| K \ln K},$$

where $C > 0$ is a universal constant, $\mathcal{A}$ is the set of actions, and K is the number of ICL iterations.

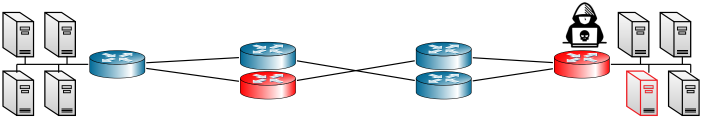
Outline

- **Automated security with a foundation model**
  - *Overview of our framework.*
- **Theoretical analysis**
  - *Controlling the hallucination bound.*
  - *Regret bound.*
- **Case study: Incident Response**
  - *Comparison with frontier models.*

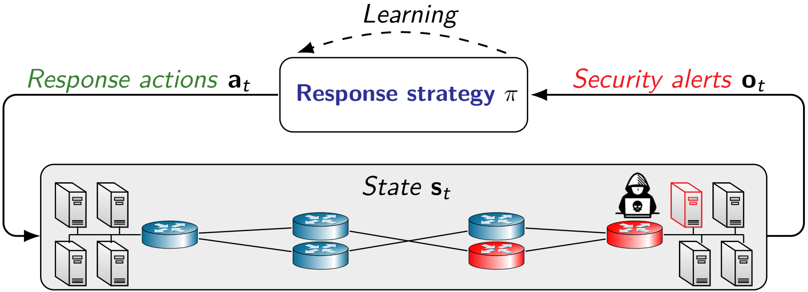
Use Case: Incident Response



Use Case: Incident Response

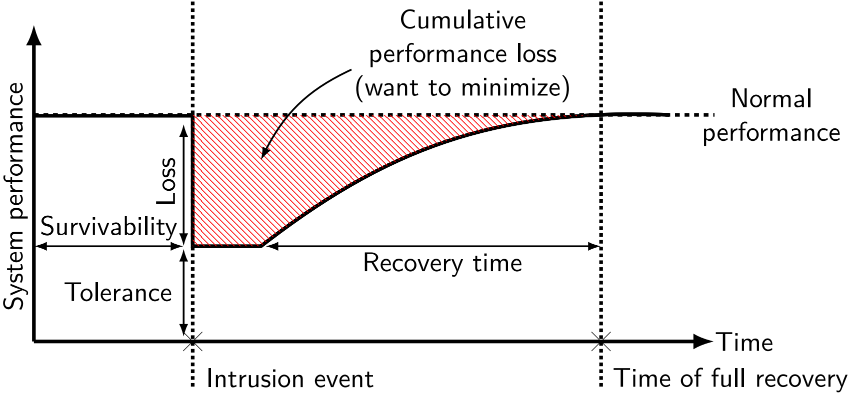


Use Case: Incident Response



- **Problem:** select actions $\mathbf{a}_0, \mathbf{a}_1, \ldots$ that drives the system to a secure and operational state after a cyberattack.

Response Objective



Challenges

Challenge 1: Partial observability.

The operator has to select response actions based on partial indicators of compromise, such as alerts and logs.

Challenges

Challenge 1: Partial observability.

The operator has to select response actions based on partial indicators of compromise, such as alerts and logs.

Challenge 2: Large and unstructured action space.

Actions have to be tailored to the specific incident.

Challenges

Challenge 1: Partial observability.

The operator has to select response actions based on partial indicators of compromise, such as alerts and logs.

Challenge 2: Large and unstructured action space.

Actions have to be tailored to the specific incident.

Challenge 3: Time-sensitive.

Delays in initiating the response can lead to costs.

Current Practice

- Incident response is **managed by security experts**.
- We have a global shortage of more than 4 million experts.
- Pressing need for new decision support systems!

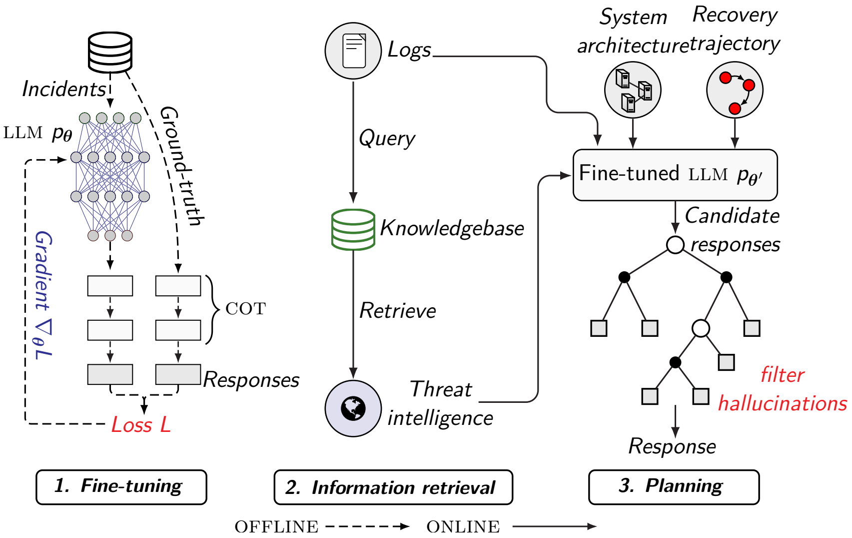
Current Practice

- Incident response is **managed by security experts**.
- We have a global shortage of more than 4 million experts.
- Pressing need for new decision support systems!

Current Practice

- Incident response is **managed by security experts**.
- We have a global shortage of more than 4 million experts.
- Pressing need for new decision support systems!

Experiment Setup

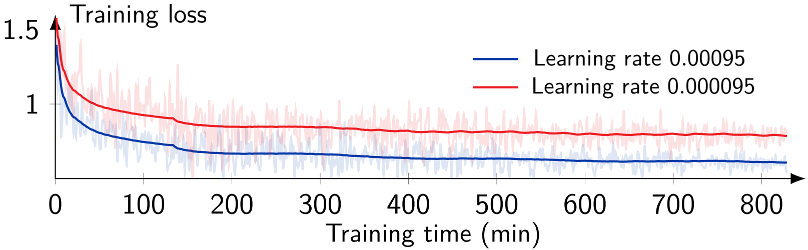


Instruction Fine-Tuning

- We fine-tune the DEEPSEEK-R1-14B LLM on a dataset of $68,000$ incidents $\mathbf{x}$ and responses $\mathbf{y}$.
- Minimize the **cross-entropy loss:**

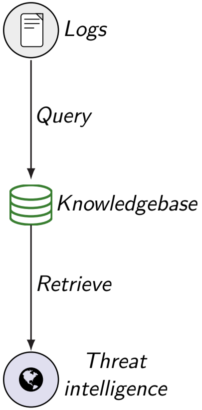
$$L = -\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{m_i} \ln p_{\theta}\left(\mathbf{y}_k^i \mid \mathbf{x}^i, \mathbf{y}_1^i, \ldots, \mathbf{y}_{k-1}^i\right),$$

where m_i is the length of the vector $\mathbf{y}^i$.



Retrieval-Augmented Generation (RAG)

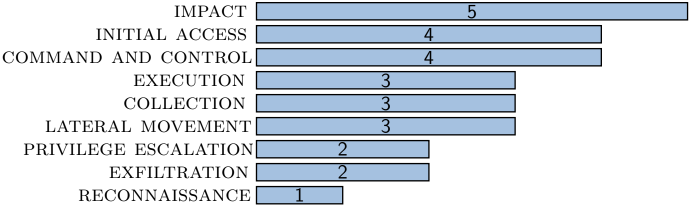
- We use regular expressions to extract indicators of compromise (IOC) from logs.
  - e.g., IP addresses, vulnerability identifiers, etc.
- We use the IOCs to **retrieve information about the incident** from public threat intelligence APIs, e.g., OTX.
- We include the retrieved information in the context of the LLM.



Experimental Evaluation

- We evaluate our system on 4 public datasets.

Dataset	*System*	*Attacks*
CTU-Malware-2014	Windows xp sp2 servers	Various malwares and ransomwares.
CIC-IDS-2017	Windows and Linux servers	Denial-of-service, web attacks, SQL injection, etc.
AIT-IDS-V2-2022	Linux and Windows servers	Multi-stage attack with reconnaissance, cracking, and escalation.
CSLE-IDS-2024	Linux servers	SambaCry, Shellshock, exploit of CVE-2015-1427, etc.



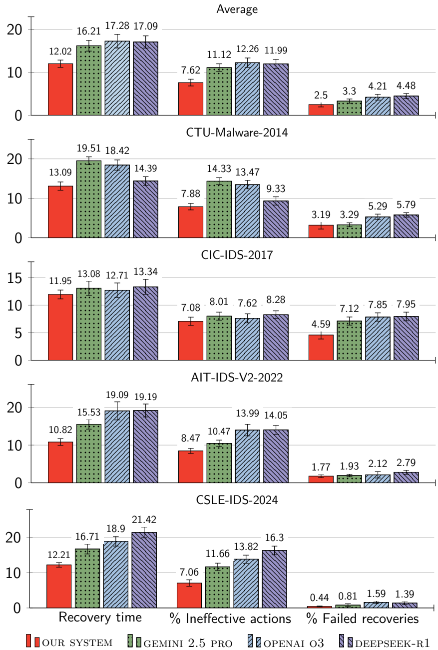
Distribution of MITRE ATT&CK tactics in the evaluation datasets.

Baselines

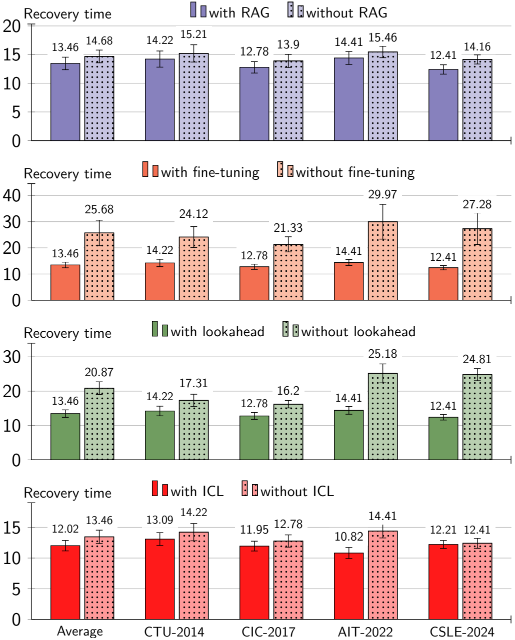
- We compare our system against **frontier LLMs.**
- Compared to the frontier models, our system is lightweight.

System	*Number of parameters*	*Context window size*
OUR SYSTEM	14 billion	128,000
DEEPSEEK-R1	671 billion	128,000
GEMINI 2.5 PRO	unknown ($\geq$ 100 billion)	1 million
OPENAI O3	unknown ($\geq$ 100 billion)	200,000

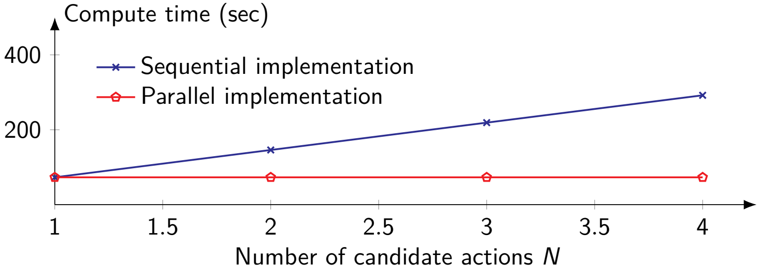
Evaluation Results



Ablation Study



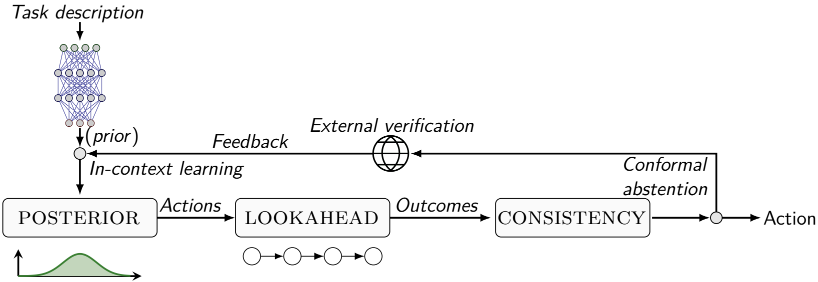
Scalability



- The lookahead optimization is computationally intensive since it requires making multiple inferences with the LLM.
- The computation can be parallelized across multiple GPU.

Conclusion

- **Foundation models will play a key role in cybersecurity.**
  - Effective at tackling the scalability challenge.
  - Remarkable knowledge management capabilities.
- We present a framework for security planning.
  - Allows to control the hallucination probability.
  - Significantly outperforms frontier LLMs.



References

- **Paper**
  - https://arxiv.org/abs/2508.05188
  - (A new paper will be released soon.)
- **Code**
  - https://github.com/Limmen/csle
- **Demonstration**
  - https://www.youtube.com/watch?v=XXo4Y6LCWk4
- **Data & Weights**
  - https://huggingface.co/datasets/kimhammar/CSLE-IncidentResponse-V1
  - https://huggingface.co/kimhammar/LLMIncidentResponse