Automated Security Response through Online Learning with Adaptive Conjectures[1]

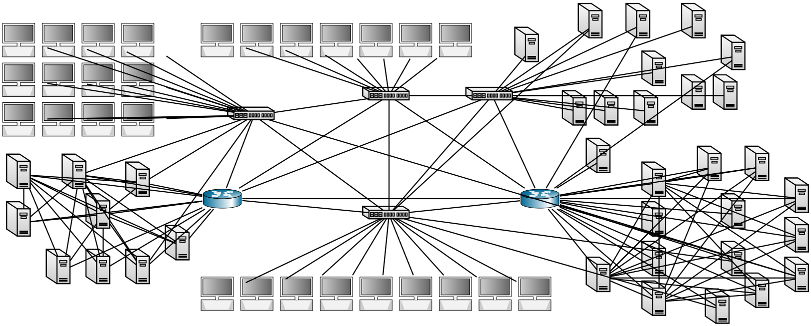
Kim Hammar, Tao Li, Rolf Stadler, & Quanyan Zhu

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

April 5, 2024

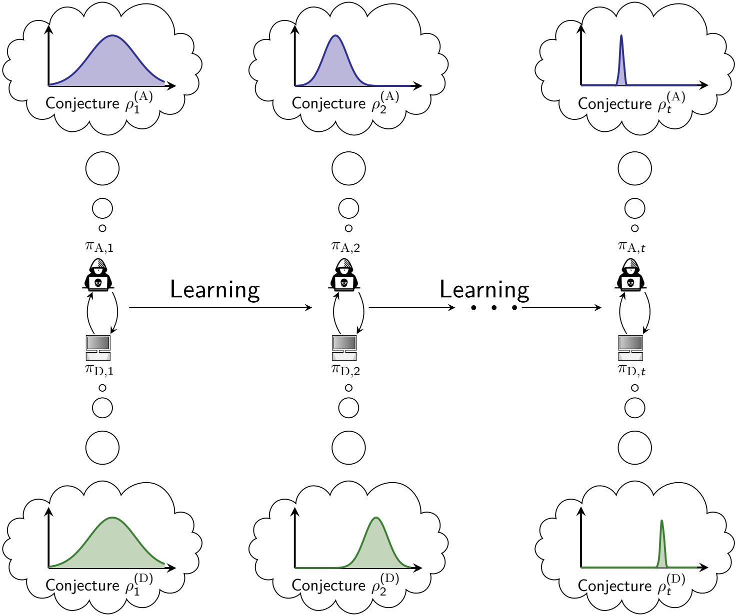
[1] Kim Hammar, Tao Li, Rolf Stadler, and Quanyan Zhu. *Automated Security Response through Online Learning with Adaptive Conjectures*. Submitted to the IEEE, https://arxiv.org/abs/2402.12499. 2024.

Challenge: IT Systems are Complex



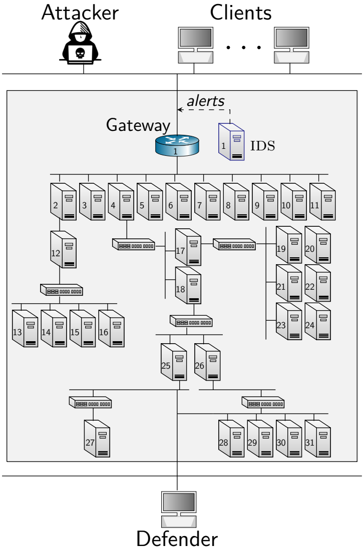
- It is not realistic that any model will capture all the details.
  - $\Longrightarrow$ We have to work with **approximate models**.
  - $\Longrightarrow$ model misspecification.
- How does misspecification affect optimality and convergence?

Our Contribution: **C**onjectural **O**nline **L**earning



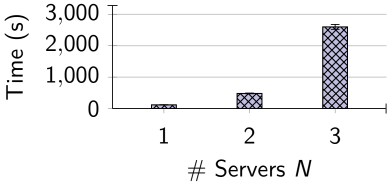
Use Case: Security Response

- A **defender** owns an infrastructure.
  - Defends the infrastructure by monitoring and response.
  - Has partial observability.
- An **attacker** seeks to intrude on the infrastructure.
  - Wants to compromise specific components.
  - Attacks by reconnaissance, exploitation and pivoting.

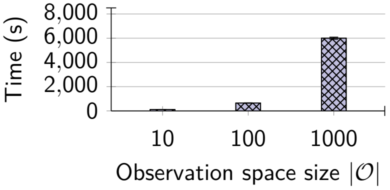


Prior Work

- Assumes a stationary model with no misspecification
  - Limitation: fails to capture many real-world systems.
- Focuses on offline computation of defender strategies
  - Limitation: computationally intractable for realistic models.



(a) $|\mathcal{O}| = 10$.



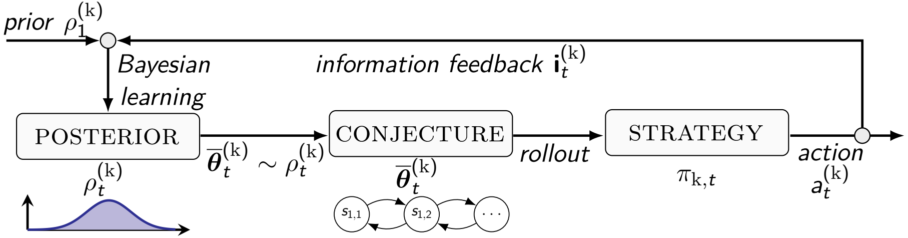
(b) $N = 1$.

Time required to compute a perfect Bayesian equilibrium with HSVI.

Problem

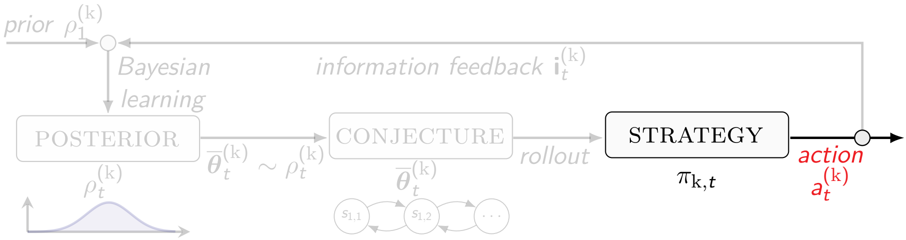
- **Partially-observed stochastic game $\Gamma_{\boldsymbol{\theta}_t}$.**
- $\Gamma_{\boldsymbol{\theta}_t}$ is parameterized by $\boldsymbol{\theta}_t$, which is hidden.
- Player k has a conjecture of $\boldsymbol{\theta}_t$, denoted by $\overline{\boldsymbol{\theta}}_t \in \Theta_{\mathrm{k}}$.
- The player is misspecified if $\boldsymbol{\theta}_t \notin \Theta_{\mathrm{k}}$.
- As $\boldsymbol{\theta}_t$ evolves, the player **adapts its conjecture**.
- The player uses the conjecture to update its strategy $\pi_{\mathrm{k},t}$.
- ***What is an effective method to update conjectures and strategies?***

Our Method: **C**onjectural **O**nline **L**earning



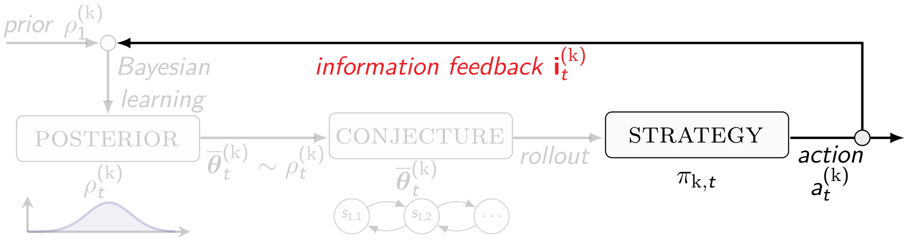
The conjecture distribution $\rho_t^{(k)}$ is calibrated through **Bayesian learning** and the strategy $\pi_{k,t}$ is updated through rollout.

Our Method: **C**onjectural **O**nline **L**earning



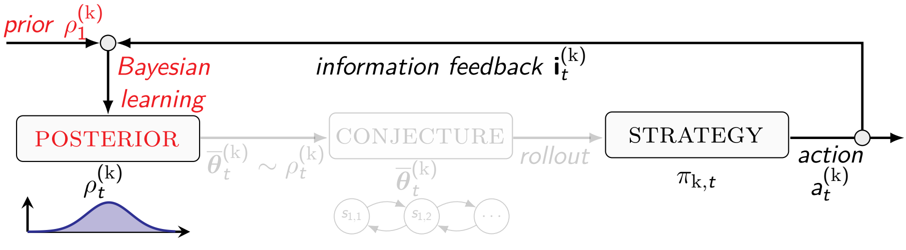
The conjecture distribution $\rho_t^{(\mathrm{k})}$ is calibrated through **Bayesian learning** and the strategy $\pi_{\mathrm{k},t}$ is updated through rollout.

Our Method: **C**onjectural **O**nline **L**earning



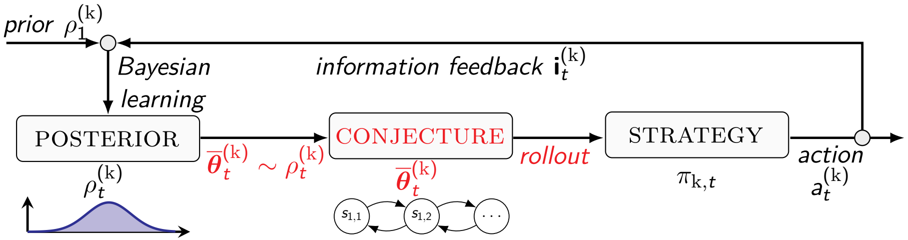
The conjecture distribution $\rho_t^{(\mathrm{k})}$ is calibrated through **Bayesian learning** and the strategy $\pi_{\mathrm{k},t}$ is updated through rollout.

Our Method: **C**onjectural **O**nline **L**earning



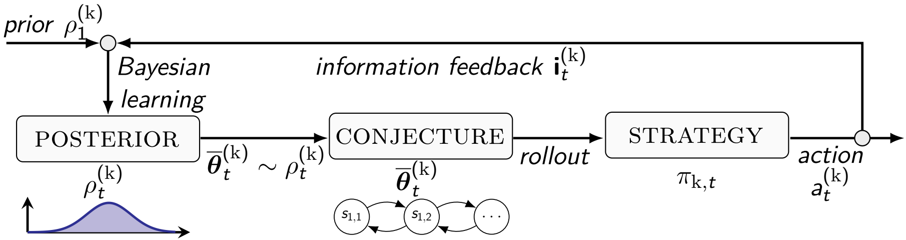
The conjecture distribution $\rho_t^{(\mathrm{k})}$ is calibrated through **Bayesian learning** and the strategy $\pi_{\mathrm{k},t}$ is updated through rollout.

Our Method: **C**onjectural **O**nline **L**earning



The conjecture distribution $\rho_t^{(k)}$ is calibrated through **Bayesian learning** and the strategy $\pi_{k,t}$ is updated through rollout.

Our Method: **C**onjectural **O**nline **L**earning

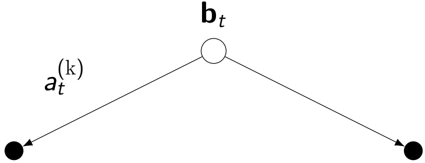


The conjecture distribution $\rho_t^{(\mathrm{k})}$ is calibrated through **Bayesian learning** and the strategy $\pi_{\mathrm{k},t}$ is updated through rollout.

Strategy Adaptation through Conjectural Rollout (1/2)

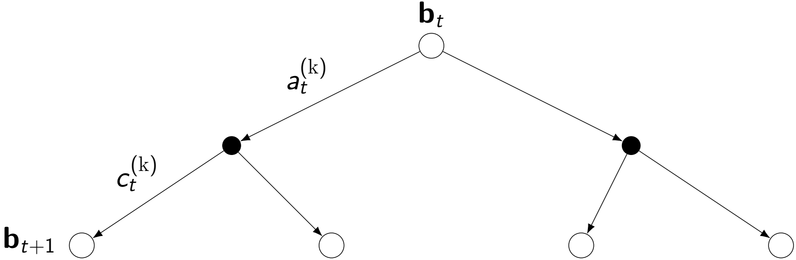
Strategy Adaptation through Conjectural Rollout (1/2)

Conjectured lookahead tree of player k.



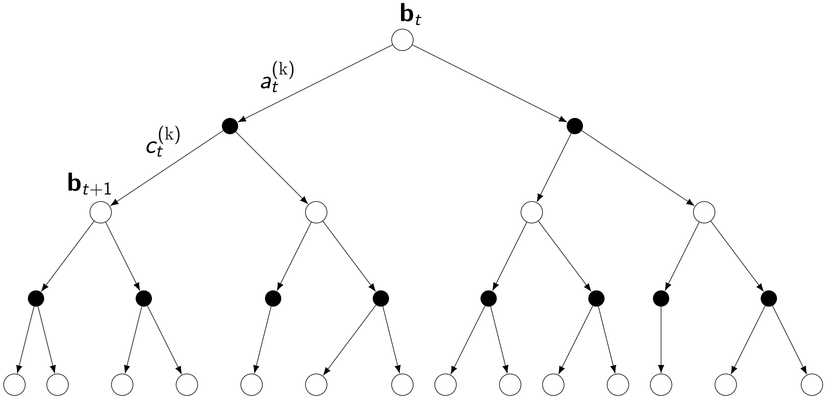
Strategy Adaptation through Conjectural Rollout (1/2)

Conjectured lookahead tree of player k.



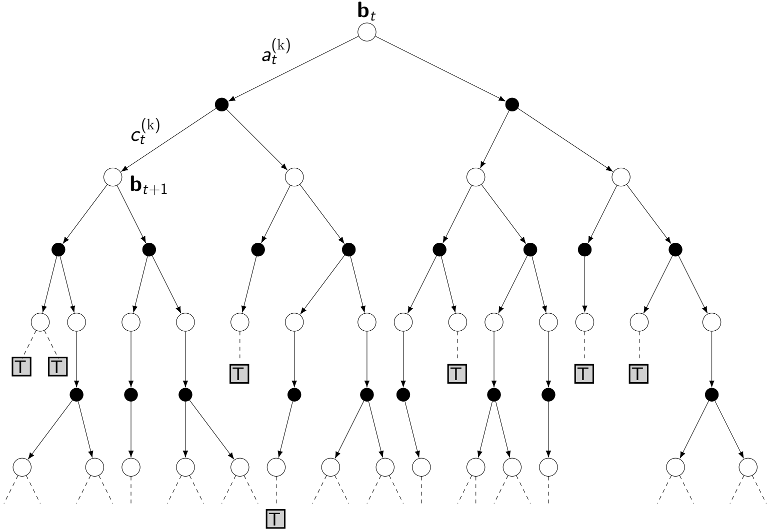
Strategy Adaptation through Conjectural Rollout (1/2)

Conjectured lookahead tree of player k.



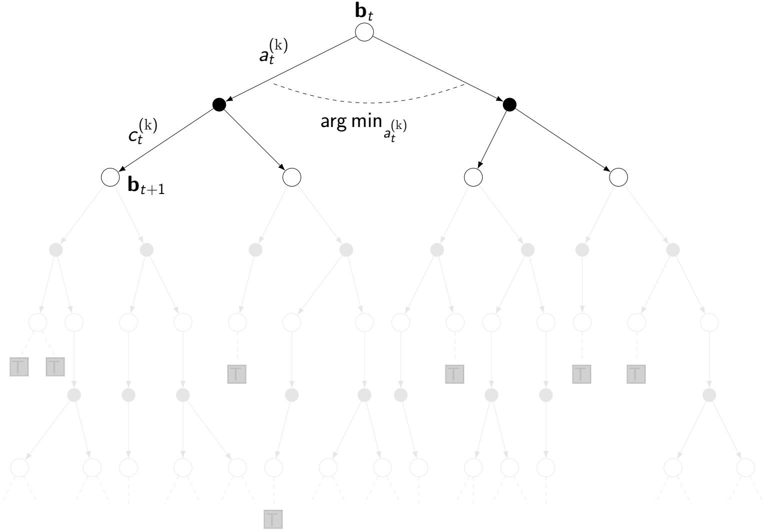
Strategy Adaptation through Conjectural Rollout (1/2)

Conjectured lookahead tree of player k



Strategy Adaptation through Conjectural Rollout (1/2)

ℓ_{k}-step rollout.



Strategy Adaptation through Conjectural Rollout (2/2)

ℓ_{k}-**step rollout** based on the conjectured model:

$$\pi_{\mathrm{k},t}(\mathbf{b}_t) \in \mathscr{R}(\overline{\boldsymbol{\theta}}_t^{(\mathrm{k})}, \mathbf{b}_t, \overline{J}_{\mathrm{k}}^{(\boldsymbol{\pi}_t)}, \ell_{\mathrm{k}}) \triangleq \underset{a_t^{(\mathrm{k})}, a_{t+1}^{(\mathrm{k})}, \ldots, a_{t+\ell_{\mathrm{k}}-1}^{(\mathrm{k})}}{\arg\min} \tag{1}$$

$$\mathbb{E}_{\boldsymbol{\pi}_t}\left[\sum_{j=t}^{t+\ell_{\mathrm{k}}-1} \gamma^{j-t} c_{\mathrm{k}}(S_j, A_j^{(\mathrm{D})}) + \gamma^{\ell_{\mathrm{k}}} \overline{J}_{\mathrm{k}}^{(\boldsymbol{\pi}_t)}(\mathbf{B}_{t+\ell_{\mathrm{k}}}) \mid \mathbf{b}_t\right].$$

- $\overline{\boldsymbol{\theta}}_t^{(\mathrm{k})}$ is the model conjecture.
- c_{k} is the cost function.
- $\overline{J}_{\mathrm{k}}^{\boldsymbol{\pi}_t}$ is the conjectured cost-to-go under strategy profile $\boldsymbol{\pi}_t$.
- $\mathbf{b}_t$ is the current belief state.

Performance Guarantees of Rollout (1/2)

Theorem

*The **conjectured cost** of player* k*'s rollout strategy* $\pi_{\mathrm{k},t}$ *satisfies*

$$\overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},t},\overline{\pi}_{-\mathrm{k},t})}(\mathbf{b}) \leq \overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},1},\overline{\pi}_{-\mathrm{k},t})}(\mathbf{b}) \qquad \forall \mathbf{b} \in \mathcal{B}. \qquad \text{(A)}$$

Intuition:

- The **rollout policy improves the base policy** in the conjectured model (A).

Performance Guarantees of Rollout (1/2)

Theorem

*The **conjectured cost** of player k's rollout strategy $\pi_{\mathrm{k},t}$ satisfies*

$$\overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},t},\overline{\pi}_{-\mathrm{k},t})}(\mathbf{b}) \leq \overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},1},\overline{\pi}_{-\mathrm{k},t})}(\mathbf{b}) \qquad \forall \mathbf{b} \in \mathcal{B}. \tag{A}$$

Assuming** $(\overline{\theta}_t^{(\mathrm{k})}, \overline{\ell}_{-\mathrm{k}})$ **predicts the game** ℓ_{k} **steps ahead, then

$$\|\overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},t},\overline{\pi}_{-\mathrm{k},t})} - J_{\mathrm{k}}^{\star}\| \leq \frac{2\gamma^{\ell_{\mathrm{k}}}}{1-\gamma}\|\overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},1},\overline{\pi}_{-\mathrm{k},t})} - J_{\mathrm{k}}^{\star}\|, \tag{B}$$

where $J_{\mathrm{k}}^{\star}$ is the optimal cost-to-go. $\|\cdot\|$ is the maximum norm $\|J\| = \max_x |J(x)|$.

Intuition:

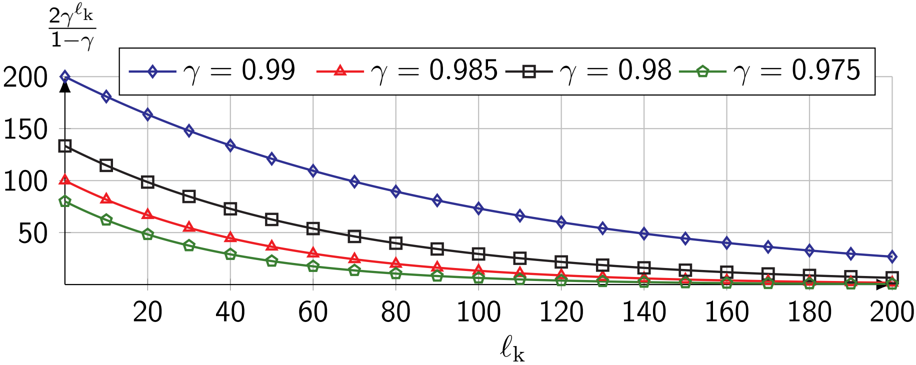
- The **rollout policy improves the base policy** in the conjectured model (A).
- If the conjectured model is wrong but can predict the next ℓ_{k} steps, then we **can bound the performance** (B).

Performance Guarantees of Rollout (2/2)

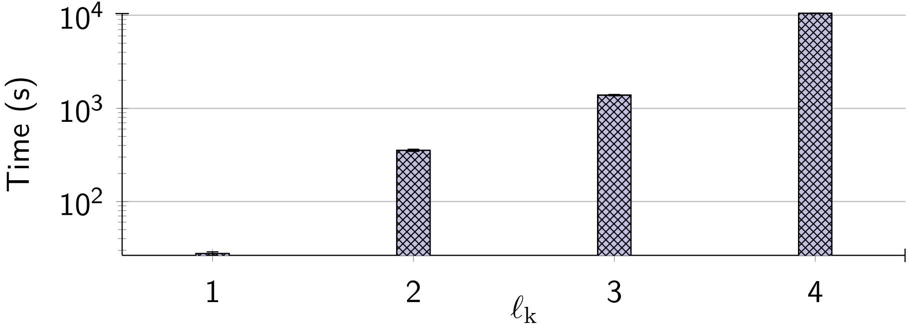
The performance bound

$$\|\overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},t},\overline{\pi}_{-\mathrm{k},t})} - J_{\mathrm{k}}^{\star}\| \leq \frac{2\gamma^{\ell_{\mathrm{k}}}}{1-\gamma}\|\overline{J}_{\mathrm{k}}^{(\pi_{\mathrm{k},1},\overline{\pi}_{-\mathrm{k},t})} - J_{\mathrm{k}}^{\star}\|, \qquad \text{(B)}$$

improves superlinearly with the lookahead horizon ℓ_{k}.

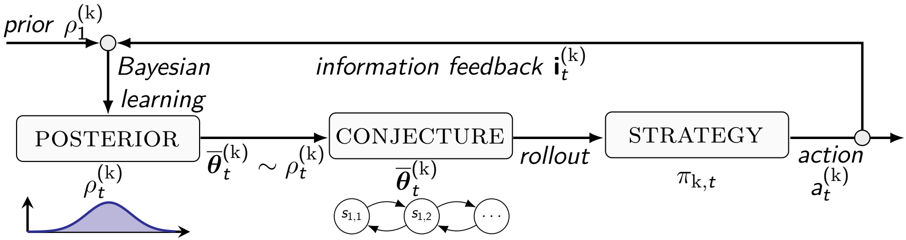


Compute Time of the Rollout Operator



Compute time of the rollout operator for varying lookahead horizons ℓ_k.

Our Method: **C**onjectural **O**nline **L**earning



The conjecture distribution $\rho_t^{(k)}$ is calibrated through **Bayesian learning** and the strategy $\pi_{k,t}$ is updated through rollout.

Bayesian Learning to Calibrate Conjectures

$\rho_t^{(\mathrm{k})}$ is calibrated through **Bayesian learning** as

$$\rho_t^{(\mathrm{k})}(\overline{\boldsymbol{\theta}}_t^{(\mathrm{k})}) \triangleq \frac{\mathbb{P}[\mathbf{i}_t^{(\mathrm{k})} \mid \overline{\boldsymbol{\theta}}_t^{(\mathrm{k})}, \mathbf{b}_{t-1}]\rho^{(\mathrm{k})}(\overline{\boldsymbol{\theta}}_{t-1}^{(\mathrm{k})})}{\int_{\Theta_\mathrm{k}} \mathbb{P}[\mathbf{i}_t^{(\mathrm{k})} \mid \overline{\boldsymbol{\theta}}_t^{(\mathrm{k})}, \mathbf{b}_{t-1}]\rho_{t-1}^{(\mathrm{k})}(\mathrm{d}\overline{\boldsymbol{\theta}}_t^{(\mathrm{k})})},$$

where $\mathbf{i}_t^{(\mathrm{k})}$ is the **information feedback** at time t.

- We want to characterize $\lim_{t\to\infty} \rho_t^{(\mathrm{k})}$.
  - Does the conjecture converge?
  - Is the conjecture consistent asymptotically?

Asymptotic Analysis of Bayesian Learning

- Let $\nu \in \Delta(\mathcal{B})$ be an occupancy measure over the belief space.
- We say that **a conjecture $\overline{\boldsymbol{\theta}}^{(\mathrm{k})}$ is consistent** if it minimizes the weighted KL-divergence:

$$K(\overline{\boldsymbol{\theta}}^{(\mathrm{k})}, \nu) \triangleq \mathbb{E}_{\mathbf{b}\sim\nu}\mathbb{E}_{\mathbf{I}^{(\mathrm{k})}}\left[\ln\left(\frac{\mathbb{P}[\mathbf{I}^{(\mathrm{k})} \mid \boldsymbol{\theta}, \mathbf{b}]}{\mathbb{P}[\mathbf{I}^{(\mathrm{k})} \mid \overline{\boldsymbol{\theta}}^{(\mathrm{k})}, \mathbf{b}]}\right) \mid \boldsymbol{\theta}, \mathbf{b}\right].$$

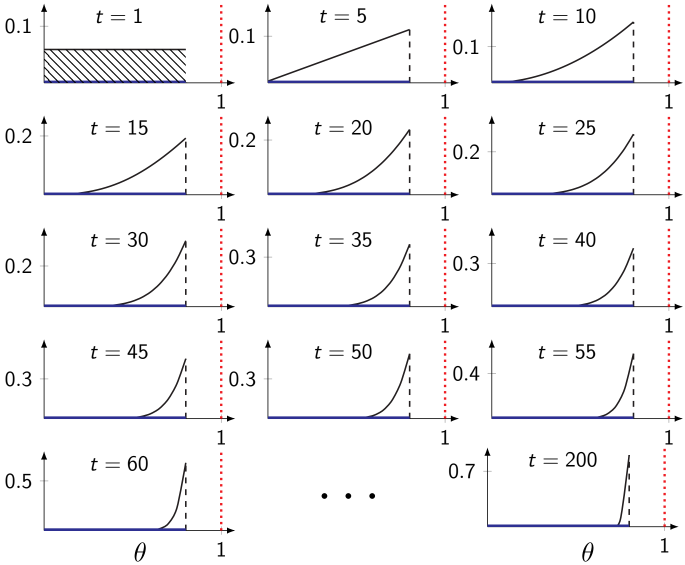
- Let $\Theta^{\star}_{\mathrm{k}}$ denote the set of consistent conjectures.

Remark

Due to misspecification, *$\overline{\boldsymbol{\theta}}^{(\mathrm{k})}_t \in \Theta^{\star}_{\mathrm{k}}$ does not imply that $\overline{\boldsymbol{\theta}}^{(\mathrm{k})}_t$ equals the true parameter vector $\boldsymbol{\theta}_t$.*

Bayesian Learning Converges to Consistent Conjectures

Intuitively, consistent conjectures are "closest" to the true model.



Conjecture $\rho_t^{(k)}$ --- Consistent ····· True model θ — Θ_k

Bayesian Learning is Consistent Asymptotically

As $t \to \infty$, **the conjecture distribution** $\rho_t^{(\mathrm{k})}$ **concentrates on the set of consistent conjectures.**

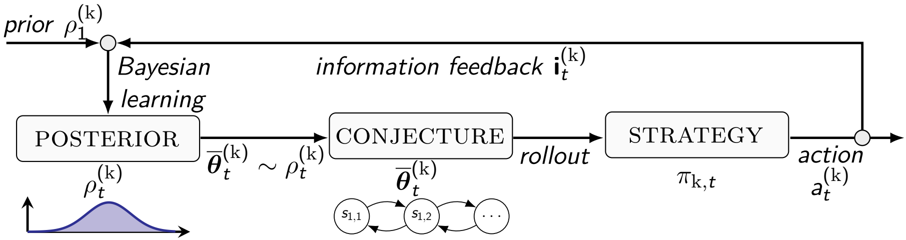
Theorem

Given certain regularity conditions, the following property is guaranteed by COL*.*

$$\lim_{t \to \infty} \int_{\Theta_{\mathrm{k}}} \left(K(\overline{\boldsymbol{\theta}}, \nu_t) - K^{\star}_{\Theta_{\mathrm{k}}}(\nu_t) \right) \rho_t^{(\mathrm{k})}(\mathrm{d}\overline{\boldsymbol{\theta}}) = 0$$

a.s.-$\mathbb{P}^{\mathscr{R}}$*, where* $K^{\star}_{\Theta_{\mathrm{k}}}$ *denotes the minimal weighted* KL*-divergence.*

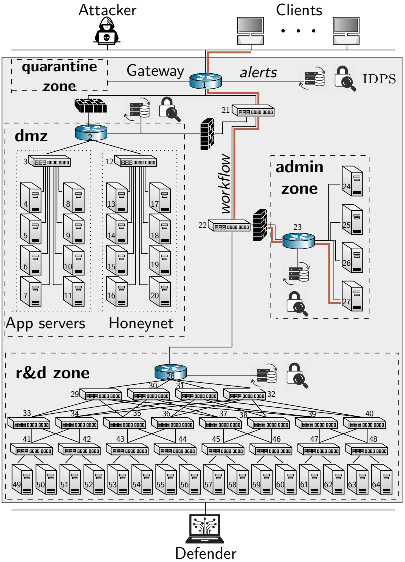
Our Method: **C**onjectural **O**nline **L**earning



The conjecture distribution $\rho_t^{(k)}$ is calibrated through **Bayesian learning** and the strategy $\pi_{k,t}$ is updated through rollout.

Evaluation - Target Infrastructure

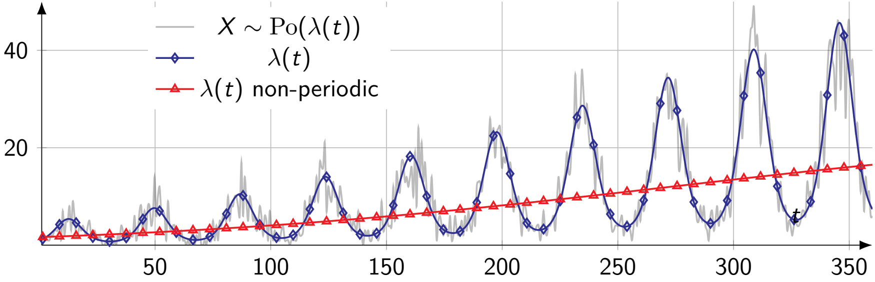
- **Target infrastructure** to the right.
- Defender monitors the infrastructure through IDS alerts.
- Attacker seeks to compromise servers.
- The *position of the attacker is unknown*.
- Defender can recover compromised servers at a cost.



Model Parameter

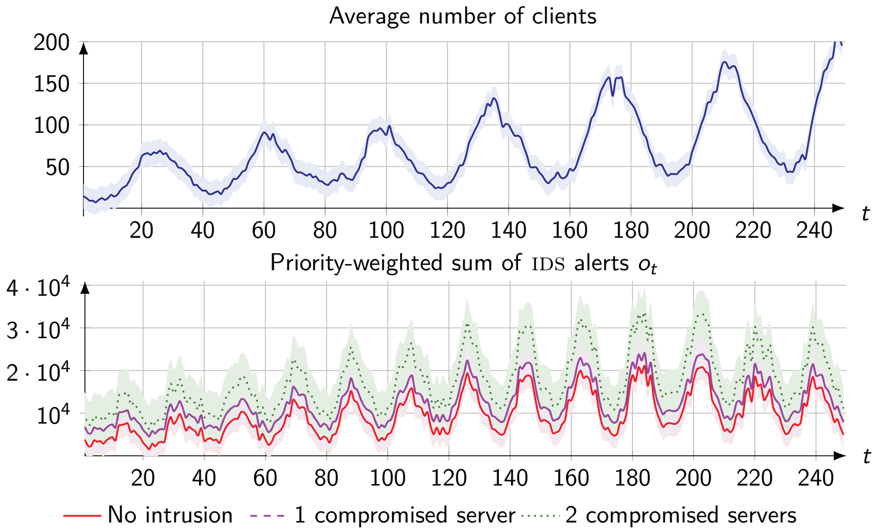
- Let $\boldsymbol{\theta}_t$ represent the number of clients.
- Clients arrive according to the **rate function**.

$$\lambda(t) = \exp\left(\underbrace{\sum_{i=1}^{\dim(\psi)} \psi_i t^i}_{\text{trend}} + \underbrace{\sum_{k=1}^{\dim(\chi)} \chi_k \sin(\omega_k t + \phi_k)}_{\text{periodic}}\right).$$

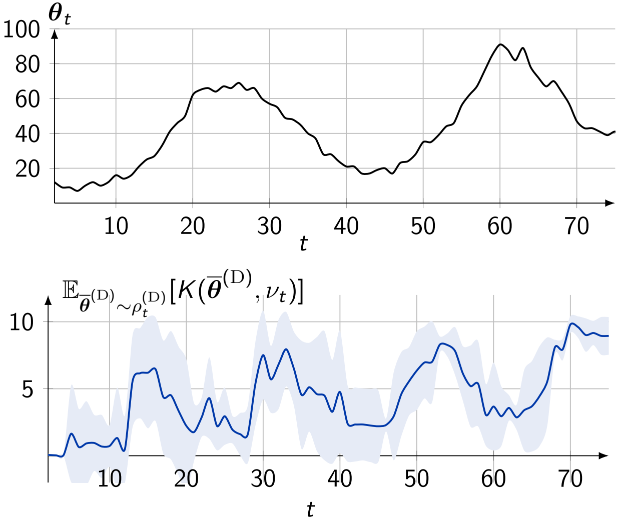


Correlation Between Observations and the Model

- We collect measurements from our testbed to estimate the distribution of IDS alerts.



Evaluation of COL (1/3)

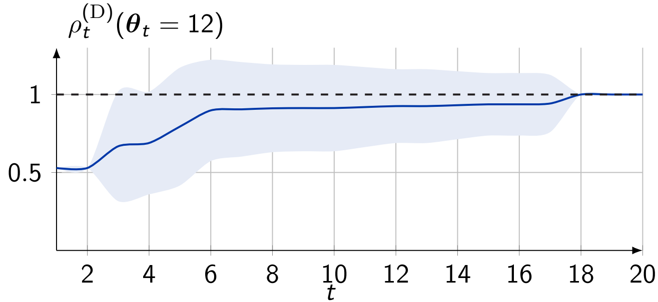


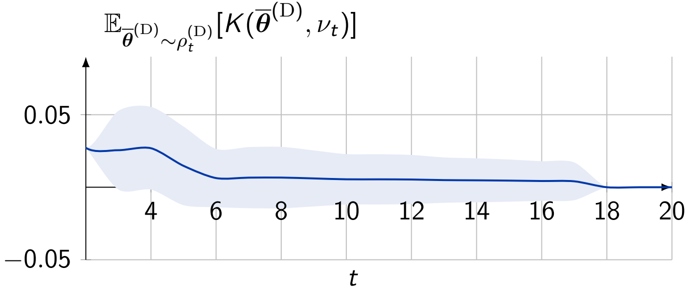
Remark

The conjectures do not converge if $\boldsymbol{\theta}_t$ keep changing.

Evaluation of COL (2/3)

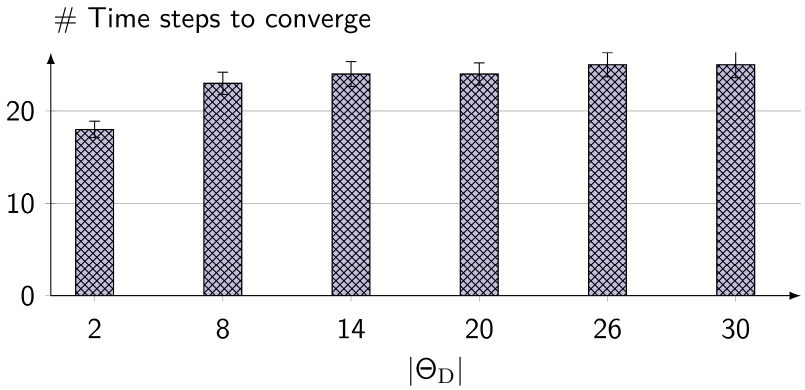
Fix the number of clients to be $\theta_t = 12$ for all t.





Evaluation of COL (3/3)

Fix the number of clients to be $\theta_t = 12$ for all t.[2][3]



[2] Kim Hammar, Tao Li, Rolf Stadler, and Quanyan Zhu. *Automated Security Response through Online Learning with Adaptive Conjectures*. Submitted to the IEEE, https://arxiv.org/abs/2402.12499. 2024.

[3] Further evaluations can be found in the paper.

Conclusion

- We introduce a **novel game-theoretic formulation** of automated security response where each player has a probabilistic conjecture about the game model.

- We present **C**onjectural **O**nline **L**earning, a theoretically-sound method for **online learning of security strategies** in non-stationary and uncertain environments.