Measuring Policy Similarity in a Cyber Attack Game

Pontus Johnson

30 January 2025

1 Introduction

Consider a cyber attack game with two agents (e.g., two adversaries or two defensive strategies). Each agent follows a stochastic policy that determines its next action based on a sequence of previous events (actions, targets, etc.). We wish to estimate how *similar* these two policies are using logged game data.

Each entry in the logs has the format:

$$\{\text{timestamp, principal, action, target}\},$$

for example:

{`2025-01-26 08:55:43.393`, `service_account_4324`, `list_bucket`, `storage_bucket_4587`}.

In this document, we provide a step-by-step method to estimate the policies and measure their similarity.

2 Modeling Each Agent's Policy

We treat each agent as having a (possibly k-th order) Markov policy:

$$\pi(a \mid s) = \Pr(\text{next action} = a \mid \text{state} = s),$$

where the state s may encapsulate the last k actions (plus any relevant contextual information, like targets or time intervals).

2.1 Defining the State Space

We assume each agent's policy depends on the last k events. For each agent, define:

$$s_t = (a_{t-1}, a_{t-2}, \dots, a_{t-k}),$$

where a_i denotes the i-th action in the sequence. Alternatively, you can extend s_t to include targets or time gaps if those features are relevant.

3 Estimating Each Policy from Logs

Given a set of logs for each agent, we construct an empirical estimate $\hat{\pi}(a \mid s)$ as follows:

1. **Parse the logs.** For each agent, extract the sequence of actions (and any auxiliary info) in chronological order.
2. **Form (state, action) pairs.** For each time t, treat

   $$\big(s_t,\ a_t\big),$$

   where s_t is the state derived from the last k actions.
3. **Count frequencies.** For each state s, and each action a, count how many times the pair (s, a) occurs:

   $$N_i(s,a) \ = \ \text{number of times agent } i \text{ was in state } s \text{ and took action } a.$$

   Also count

   $$N_i(s) \ = \ \text{number of times agent } i \text{ was in state } s.$$
4. **Compute empirical probabilities.** The estimated policy is

   $$\hat{\pi}_i(a \mid s) \ = \ \frac{N_i(s,a)}{N_i(s)}.$$

   In practice, smoothing techniques (like Laplace smoothing) may be needed if $N_i(s)$ is small or zero for some states.

4 Comparing Two Policies

Once we have $\hat{\pi}_1$ and $\hat{\pi}_2$, we compare them via a distance measure between the distributions over actions for *the same* state s. Common measures include:

Total Variation Distance (TV).

$$D_{\text{TV}}(p,q) \ = \ \frac{1}{2}\sum_{a\in\mathcal{A}}\big|\,p(a) - q(a)\big|.$$

Kullback–Leibler (KL) Divergence.

$$D_{\text{KL}}(p\,\|\,q) \ = \ \sum_{a\in\mathcal{A}} p(a)\,\log\frac{p(a)}{q(a)},$$

which is not symmetric in p and q, and requires $q(a) > 0$ whenever $p(a) > 0$.

Jensen–Shannon Divergence (JS).

$$D_{\text{JS}}(p,q) \ = \ \frac{1}{2}D_{\text{KL}}\Big(p\,\|\,\frac{p+q}{2}\Big) \ + \ \frac{1}{2}D_{\text{KL}}\Big(q\,\|\,\frac{p+q}{2}\Big).$$

This is symmetric and always finite (since it includes smoothing by $\frac{p+q}{2}$).

4.1 Aggregating Over States

Let $d(\hat{\pi}_1(\cdot \mid s), \hat{\pi}_2(\cdot \mid s))$ be any of the above local distances or divergences. We then obtain an *overall* distance by taking a weighted sum (or expectation) across all states. For instance:

$$D(\hat{\pi}_1, \hat{\pi}_2) \;=\; \sum_{s \in \mathcal{S}} \Pr_{\hat{\pi}_1 \cup \hat{\pi}_2}(s)\; d\big(\hat{\pi}_1(\cdot \mid s),\, \hat{\pi}_2(\cdot \mid s)\big),$$

where $\Pr_{\hat{\pi}_1 \cup \hat{\pi}_2}(s)$ is an empirical frequency of state s based on one or both agents' logs (e.g., normalized to sum to 1 over all states that appear in either log).

5 Practical Considerations

- **Data sparsity:** Many states may be rarely observed. Consider smoothing or ignoring states below a certain frequency threshold.
- **Unseen actions or states:** If one policy exhibits states or actions never seen in the other, decide whether to ignore them or assign small probabilities via smoothing.
- **Choice of Markov order k:** Higher k captures more context but leads to fewer samples per state. In practice, choose k to balance modeling power and data availability.
- **Temporal aspects vs. pure sequences:** If timing is relevant (e.g., response latency, intervals between actions), you can add time-related features to the state.

6 Conclusion

To measure how similarly two agents behave in a cyber attack game, we:

1. Represent each agent's game log as an estimated Markov (k-gram) model;
2. Compute $\hat{\pi}_1(a \mid s)$ and $\hat{\pi}_2(a \mid s)$ for each state s;
3. Compare the two distributions via a suitable distance measure (TV, KL, JS, etc.);
4. Weight by the frequency of states to get an overall similarity or distance score.

This procedure effectively captures the degree to which the two agents choose the same (or similar) actions in the same or similar contexts, providing a principled, data-driven measure of policy similarity.