AI Observability for Detecting Security and Reliability Threats in Agentic AI Systems

Motivation

Agentic AI systems increasingly interact with external tools, databases, APIs, and other services on behalf of users. This creates new challenges for **security, privacy, and reliability**, because harmful or incorrect behavior may occur across multiple stages of an agent's execution. For example, a **prompt-injection attack** may manipulate an agent into ignoring its intended instructions and performing unauthorized actions. **Hallucination** may cause an agent to generate unsupported or incorrect information. In addition, **sensitive information** such as personal data, credentials, confidential documents, system prompts, or other protected information may enter the agent through user inputs, retrieved data, tool results, or other application components and may subsequently appear in LLM outputs or tool requests. Traditional application logging is often insufficient to understand these behaviors because a single agent request may involve multiple LLM calls, retrieval operations, database queries, tool invocations, and intermediate decisions. **AI observability** can provide a structured view of this execution by capturing traces, events, logs, and metrics associated with the complete agent workflow. This project will therefore investigate how observability can be integrated into an agentic AI pipeline and whether the collected telemetry can provide useful signals for detecting **prompt injection, hallucination, and sensitive information exposure**. Existing observability technologies will be used rather than implementing an observability infrastructure from scratch.

Project Objective

The objective is to **design and prototype an observable agentic AI pipeline that collects and analyzes runtime telemetry to support the detection of security, privacy, and reliability threats**.

The prototype will investigate the complementary roles of:

- **OpenTelemetry (OTel)** as the vendor-neutral instrumentation and telemetry layer for collecting and transporting standardized traces, metrics, and logs/events.
- **Langfuse** as an LLM/agent observability platform for storing, visualizing, and inspecting LLM and agent traces, including inputs/outputs, token usage, latency, sessions, users, tool interactions, and custom evaluations.
- **Visualization/monitoring tools**, such as Grafana, where appropriate, for dashboards and aggregated operational metrics.
- **OPA or a lightweight policy engine** as an optional control layer for making simple policy decisions based on detected conditions.

The prototype will implement a simple agentic AI application containing an LLM and selected external capabilities such as a SQL database and/or MCP-based tools.

Main Goals

1. **Build a minimal agentic AI prototype** containing an LLM and selected tools such as SQL and/or MCP.
2. **Instrument the application using OpenTelemetry / Langfuse** to capture runtime telemetry across the agent workflow.
3. **Integrate Langfuse** to visualize and inspect LLM/agent traces, inputs, outputs, tool interactions, latency, token usage, sessions, users, and evaluations.
4. **Define application-specific security and reliability evaluations** for:
    - prompt-injection detection;
    - hallucination/groundedness;
    - sensitive-information detection.
5. **Investigate relationships between telemetry signals and threats**, such as whether unexpected tool calls, abnormal execution paths, evidence/output inconsistencies, or sensitive-information indicators provide useful detection signals.
6. **Investigate telemetry privacy**, including which information should be collected, which information should be redacted or masked, and whether sensitive information should be stored in raw form at all.
7. **Prototype a lightweight policy layer using OPA or a comparable policy engine** to demonstrate actions such as:
    - allow;
    - block;
    - require human approval;
    - redact;
    - flag for investigation.
8. **Evaluate the prototype** using benign scenarios and intentionally constructed attack/failure scenarios, measuring detection capability, false positives, observability coverage, and practical limitations.

Threat Investigation

The project investigates three threat categories using observability signals: **prompt injection** (via suspicious instructions, unexpected tool selection/arguments, policy violations, abnormal call sequences), **hallucination** (via consistency between LLM output and retrieved evidence/context, groundedness scores, user feedback), and **sensitive information exposure** (tracking sensitive data input / outputs as it propagates).