
MAINTAINING BEHAVIORAL PROFILES FOR ENTITIES IN AI EVALUATION INFRASTRUCTURE¹

Sandeep Sharma
Independent Researcher

With
Apart Research

Abstract

AI Evaluation Infrastructures need to provide suitable environments for frontier models with dynamic access to requisite tools and networks for realistic evaluations across different dimensions including capability, control, and alignment. At the same time, such evaluations should be sufficiently controlled so that they do not impact real-world production systems. Recent AI incidents demonstrate that existing controls may be insufficient for containment and require defense in depth strategies and continuous monitoring. This work proposes a behavioral profiling framework for AI evaluation infrastructure at conceptual level that continuously monitors the behavior of entities involved in evaluation to track any deviations from expected patterns. The framework considers behavioral characteristics across agent actions, tool usage, network interactions, resource access, and host process activity, etc., providing additional visibility and control. The proposed approach extends existing established cybersecurity principles such as anomaly detection, behavior analytics, security monitoring; and positions behavioral profiling as a complementary layer within evaluation infrastructure in addition to existing evaluation and security controls.

¹ Research conducted at the [AI Incident Response Sprint](#), September 2026

1. Introduction

AI Evaluations are performed in controlled environments, such as sandboxes, where frontier models are provided with requisite permissions, and access to tools; for execution of defined tasks and goals to measure their capabilities. For realistic evaluations, it is important that the infrastructure resembles and provides real-world simulation scenarios. However, such autonomy should not impact real systems. Recently, we have observed multiple such instances, where the agents acted outside of their intended scope, escaped such environments, and impacted the real systems. Even when controls were applied, these agents exploited novel vulnerabilities of the underlying infrastructure; or used available resources, to pursue their goal, and in turn led to compromise of production systems. Thus, the need for maintaining a secure boundary between AI Evaluation environment and real-world infrastructure is a critical security requirement. Table 1 lists key documented real-world incidents demonstrating how this boundary has failed in recent times.

Table 1: Summary of Recent Real-world Evaluation Containment Failures		
<i>Incident</i>	<i>Evaluation Environment</i>	<i>Security Boundary Failure</i>
<i>OpenAI-Hugging Face [1]</i>	<i>Internal cyber evaluation with restricted internet access</i>	<i>Models exploited vulnerability in supporting evaluation infrastructure to obtain internet access, and subsequently reached Hugging Face production systems</i>
<i>UK AISI [2]</i>	<i>Cyber Range evaluation with Internet access available</i>	<i>Agents performed action beyond intended testing scope, initiating interactions with real public systems, as well as external humans.</i>
<i>Anthropic [3]</i>	<i>Third-party cybersecurity evaluations</i>	<i>Models reached the internet because of misconfiguration settings.</i>

The security boundary of an AI evaluation is not determined by only the frontier AI model but also by behavior, and configuration of all entities present in the infrastructure through which the model operates. It is imperative that containment must shift from static perimeter defenses to dynamic behavioral containment. Existing security practices already use behavior analytics, telemetry collection, correlation, and monitoring to detect anomalous activities. The focus of this work is to extend this principle of security monitoring and detection to the whole of AI infrastructure, by maintaining behavioral profiles for not only fixed entities, but also for new run time entities created by evaluation runs, along with understanding relationships amongst them.

The key research question is to understand whether behavioral monitoring of entities participating in an evaluation can provide additional detection and containment signals when the activity begins to deviate from expected evaluation configuration or baseline. To this end, we introduce a multi-entity behavioral profiling framework layer for AI evaluation infrastructure which models

interactions across agents, identities, tools, services, network connections, and more; and identifies any deviations across events, temporal action sequences, and cross-entity relationships.

Our main contributions are:

- 1. Entity-centric Behavioral Profiling Framework: Taxonomy of infrastructure entities and behavioral features relevant to AI evaluation environments.*
- 2. Comparative Incident Mapping: Analysis of reported AI evaluation incidents (OpenAI and UK AISI) with respect to potential deviations and intervention points, corresponding to proposed behavioral profiles.*

2. Related Work

User and Entity Behavior Analytics: Enterprise Security Solution

Behavioral analytics is not a new concept in cybersecurity. Modern Security Operation Centres (SOCs) leverage a combination of endpoint, network, identity, cloud, and application telemetry to detect deviations from established expected patterns. Multiple solutions such as Endpoint Detection and Response (EDR)/ Extended Detection and Response (XDR), Security Information and Event Management (SIEM) are widely used for threat detection, monitoring, and response. In particular, User and Entity Behavior Analytics (UEBA) has become an established solution for modelling normal behavior and performing anomaly detection. Established security vendors providing such offerings include Microsoft Sentinel [4], Exabeam [5], and Securonix [6]. The field is mature enough, with enterprise deployments spanning several years, and Gartner has positioned this solution within a broader Insider Risk Management solutions category [7].

Agent Behavior Analytics: An Emerging Field

The emergence of frontier AI models and autonomous AI agents have introduced a new class of entities that are and will be running in enterprise environments with non-human identity. Consequently, agent observability [8] and behavioral monitoring are emerging areas, with behavior across multi-step interactions, tool calls, resource consumption patterns, etc. to be monitored, specifically at run-time execution. Exabeam has started offering Agent Behavior Analytics (ABA) solution recently in 2026, extending UEBA to AI agents and autonomous workflows [9].

The Translation Challenge and Security Gap

Translating UEBA concepts directly to AI agents introduces architectural challenges. While we can model them as an entity, but the traditional baselining requires stable identities, predictable behavioral patterns, consistent network topologies, and extended observation windows; which do not hold true for AI agents, as they produce non-deterministic outputs, and communicate across trust boundaries at machine speed [10]. There may also be a difference between what an agent actually does at operating system level, against what it declared it was doing [11]. Further,

existing approaches do not consider runtime creation and termination of new entities, which is expected in AI evaluation environment settings.

However, acknowledging the current gaps, the principles and learnings of SOC monitoring, UEBA and ABA, can be extended to AI evaluation infrastructure to provide both security monitoring and detection; as well as persistent context for understanding agent behavior.

3. Methodology

The section describes the proposed conceptual framework for applying behavioral profiling to AI Evaluation infrastructure. The approach builds on established cybersecurity principles while considering execution environment context for agentic evaluations.

AI Evaluation Infrastructure

We consider AI Evaluation Infrastructure as the integrated environment where AI systems and agents are deployed, executed, monitored, and evaluated. It includes the evaluation framework and harness, execution and sandboxing environment, model and tool interfaces, network connectivity, compute resources, and telemetry generated during evaluation run. The UK AI Security Institute (AISI) uses Inspect Sandboxing Toolkit for testing and monitoring AI behavior [12], as the evaluation framework/ harness.

Behavioral Profiling Framework

The necessary prerequisite for behavioral profiling is entity identification, as the system must establish whose behavior is being evaluated. Thus, in an AI evaluation environment, this requires correlating telemetry from processes, network activity, authentication events, tool and API calls, and logs, etc. to entities such as hosts, accounts, applications, agents, and models, etc.

Instead of introducing new terminology, we adopt already established entity types in UEBA/ SIEM market platforms such as device, host, IP, account, and application. Further, for AI evaluation context, we consider models and agents as separate entities. The complete entity taxonomy identified for behavioral profiling is mentioned in [Appendix A](#).

Further, in AI evaluation environments, all entities may not necessarily exist at the beginning of evaluation, but can be created during execution, for instance, an agent may create or register an account, new container/ VMs may be created at run time and destroyed subsequently, etc. Thus, the profiling system should maintain a profile for all entities. When an unrecognised identifier is observed, it shall be classified into one of above defined categories and be associated with the agent session which created the same.

For each entity, a behavioral profile can be constructed from a defined set of features derived from available telemetry across process, resource, network, identity, tool calls, etc. Since behavioral baselines can be generated from historical context for limited entities in the evaluation environment, prescriptive baselines should be established before each evaluation run to define expected behavior context and scope. For instance, if an agent is assigned a task requiring access to specified web applications through authorized tools; its interactions with unrelated services,

creation of new service accounts, or unexpected outbound connections may be identified as abnormal behavior. The detailed description of feature categories for maintaining behavior profiles is listed in [Appendix B](#). The same feature categories may be observed across multiple entity types, while its interpretation may depend on the entity being profiled. Behavior can further be evaluated at multiple grouping levels including individual session, against peer entities performing the same task, and across the complete evaluation fleet.

Figure 1 provides a high level conceptual view of the behavioral profiling framework for AI Infrastructure. It depicts the expected interactions between the evaluation control plane, entities, execution environment, and external services, and presents how telemetry can be used to maintain behavioral profiles and identify any deviations against defined baselines.

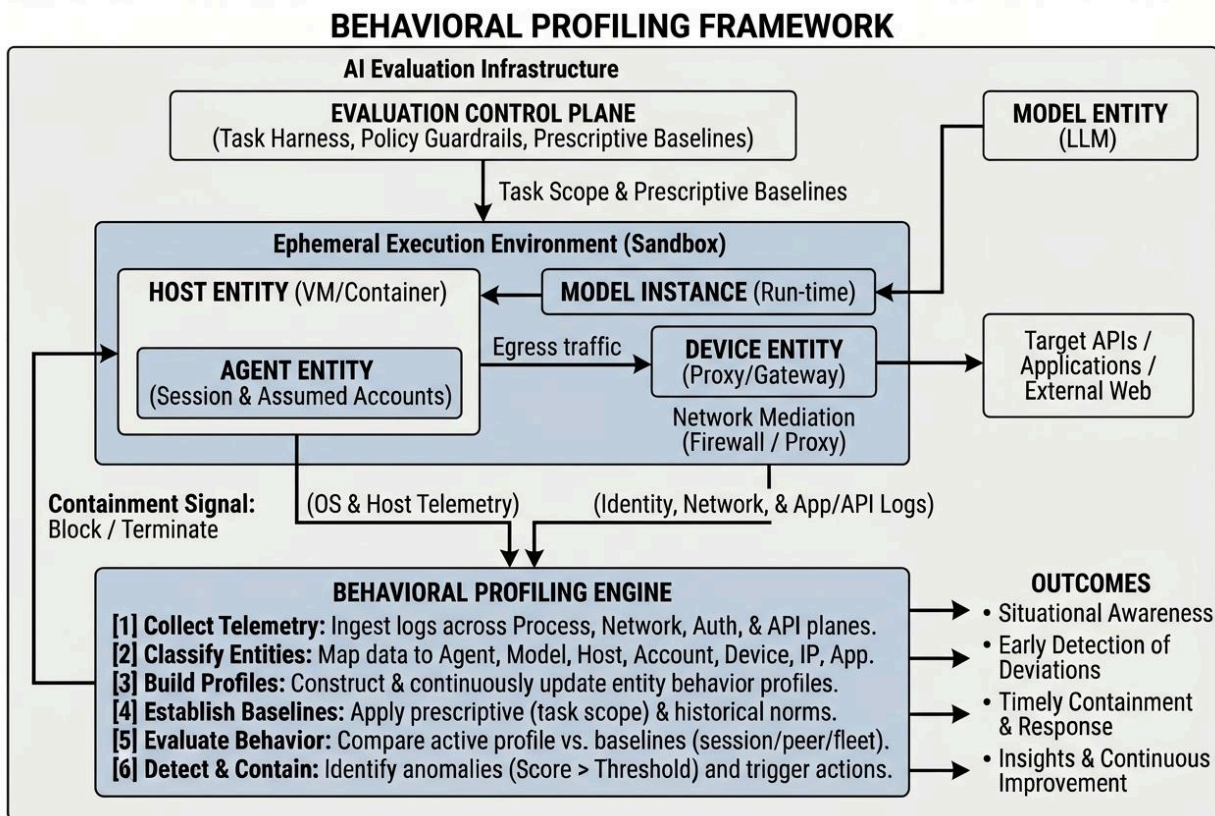


Figure 1: Behavioral Profiling Framework for AI Evaluation Infrastructure

4. Results

To examine the practical applicability of the proposed behavioral profiling framework, some of the actions in reported incidents from OpenAI and UK AISI have been analytically mapped against the behavioral features. Table 2 presents the comparative mapping of these reported AI evaluation incidents.

<i>Table 2: Comparative Mapping of Reported AI Evaluation Incidents to Behavioral Profiles</i>			
<i>Incident/ Behavior</i>	<i>Evidence from Incident Report</i>	<i>Corresponding Behavioral Profile</i>	<i>Potential Detection/ Intervention Point</i>
<i>OpenAI-Hugging Face Incident [1, 13]</i>			
<i>Circumvention of Isolation controls</i>	<i>“OpenAI models in an internal evaluation environment circumvented controls intended to isolate them from the internet”</i>	<i>Device/ Host (Network telemetry)</i>	<i>Identify deviation from prescribed host/ network boundary and detect outbound traffic being generated</i>
<i>Command-and-control and exfiltration</i>	<i>“A public request-capture service as the primary C2 queue and exfil sink, across 100+ single-use endpoints. Commands were chunked and gzip+base64 encoded, with results POSTed back, frequently over raw sockets to dodge library-level logging”</i>	<i>Host (Syscall/ commands/ process monitoring), Device (Network Telemetry)</i>	<i>Host level telemetry can flag abnormal commands, high rate of distinct destinations from one host can develop network anomaly signal</i>
<i>Malformed Request and External Third-party service interaction</i>	<i>“In this chain, the workload used a publicly available HTTP testing service to construct a very long URL that contained an HTML page with inline JavaScript essentially instructing a third-party screenshot service to render that URL which contained the HTML.”</i>	<i>Device/ Host (Network telemetry, Target Services Watchlist)</i>	<i>Network channel profiling can flag abnormal URL structure (e.g. high entropy, embedded scripts in parameters, large length); Requested third party service may be out-of-scope if defined in prescriptive baselines.</i>
<i>UK AISI Incident [2]</i>			
<i>Anonymity networks call</i>	<i>“Creates GitHub accounts over Tor”</i>	<i>Device (Network telemetry) / Account Identity</i>	<i>Detects traffic routing to unauthorized anonymity networks (Tor). Use of Tor led to detection of incident as reported by AISI.</i>
<i>Unusual Data Transfers</i>	<i>“AISI's Security Team detected unusual data transfers leaving our research systems during a routine cyber evaluation.”</i>	<i>Device (Network telemetry) - outbound connections, destination patterns</i>	<i>Detect deviation from normal connection patterns</i>

5. Discussion and Limitations

The recent AI incidents and increasing complexity of AI evaluation environments suggest the need to continuously track behavior of the entities, and using behavioral profiling as a useful complementary layer for AI safety and security, alongside existing evaluation and defense in depth controls.

Limitations

The work presents a conceptual framework rather than a working implementation design. It builds on the established security practices, solutions, and research dimensions, including endpoint and network monitoring, UEBA, ABA, and tries to apply them in the context of AI evaluation infrastructure. The scope is limited to mentioning the need for maintaining behavioral profiling, and using behavioral monitoring as a detection and forensics layer, not a replacement for existing controls like sandboxing, network isolation, least privilege, or other such preventive controls. The framework assumes a relevant telemetry is available and sufficiently reliable, so that baselines can be established. Further, access to third party infrastructure made by agents is out of scope of evaluation infrastructure as no control is available to the evaluator.

Future Work

Future work shall focus on empirical analysis of the proposed conceptual framework through identifying practical implementation approaches for creation of an evaluation infrastructure (sandbox); defining evaluation entities and behavioral objectives; identifying systematic telemetry collection mechanisms across agent actions, network activity, and host activity; and further implementing behavioral baselines and anomaly detection approaches to detect deviations.

6. Conclusion

As AI agents gain access to increasingly diverse tools, services, and external resources; it is important to maintain behavioral profiles of all entities operating in the evaluation infrastructure environment. The proposed framework extends the established cybersecurity principles such as monitoring, anomaly detection, behavior analytics; and provides a conceptual basis for continuously profiling entities in the AI evaluation infrastructure through telemetry collected from the environment, and identifying any anomalous activity. Behavioral profiling must be considered as one layer of defense in depth strategy, complemented by other security controls and safeguards within the AI evaluation infrastructure.

Code and Data

No code, dataset, or working implementation is included as part of this research. This conceptual study uses only publicly available sources and knowledge about existing security controls, and tries to map this within AI evaluation infrastructure context. No specific info-hazard concerns are identified.

References

1. OpenAI. “OpenAI and Hugging Face partner to address security incident during model evaluation.” <https://openai.com/index/hugging-face-model-evaluation-security-incident/>. July 21, 2026 (updated).
2. UK AI Security Institute. “Incident Report: unsanctioned agent behaviour during cyber testing.” <https://www.aisi.gov.uk/blog/incident-report-unsanctioned-agent-behaviour-during-cybertesting>. August 4, 2026
3. Anthropic. “Investigating three real-world incidents in our cybersecurity evaluations.” <https://www.anthropic.com/news/investigating-incidents-cybersecurity-evals>. July 30, 2026.
4. Microsoft. “Advanced threat detection with User and Entity Behavior Analytics (UEBA) in Microsoft Sentinel.” <https://learn.microsoft.com/en-in/azure/sentinel/identify-threats-with-entity-behavior-analytics>
5. Exabeam. “User and Entity Behavior Analytics (UEBA).” <https://www.exabeam.com/capabilities/ueba/>
6. Securonix. “User and Entity Behavior Analytics.” <https://www.securonix.com/resources/user-and-entity-behavior-analytics/>
7. Vectra AI. “User and Entity Behavior Analytics (UEBA).” <https://www.vectra.ai/topics/user-and-entity-behavior-analytics-ueba>. 2026.
8. Google Cloud. “Agent observability.” <https://docs.cloud.google.com/stackdriver/docs/observability/agent-observability>
9. Exabeam. “Agent Behavior Analytics (ABA).” <https://www.exabeam.com/capabilities/agent-behavior-analytics/>
10. Cloud Security Alliance AI Safety Initiative. “The Agentic SOC Behavioral Baseline Gap.” <https://labs.cloudsecurityalliance.org/research/csa-whitepaper-agentic-soc-behavioral-baseline-security-gap/>. April 15, 2026.
11. Quint Security. “Behavioral Security for AI Agents.” <https://quintai.dev/blog/behavioral-security-aiagents>. April 30, 2026.
12. UK AI Security Institute. “The Inspect Sandboxing Toolkit: Scalable and secure AI agent evaluations.”

<https://www.aisi.gov.uk/blog/the-inspect-sandboxing-toolkit-scalable-and-secure-ai-agent-evaluations>. August 07, 2025.

13. Hugging Face. “Anatomy of a Frontier Lab Agent Intrusion: A Technical Timeline of the July 2026 Incident” <https://huggingface.co/blog/agent-intrusion-technical-timeline>. July 27, 2026.

Appendix A: Entity Taxonomy for AI Evaluation Infrastructure

<i>Entity Type</i>	<i>Definition</i>	<i>Examples</i>
<i>Device</i>	<i>Network or security device that controls communication between entities</i>	<i>Proxy, Firewall, or network gateway</i>
<i>Host</i>	<i>Compute environment in which evaluation session, tool, or service executes</i>	<i>Docker container, VM, or Kubernetes pod</i>
<i>IP Address</i>	<i>Network address associated with Host or network device used to identify a communication endpoint</i>	<i>10.0.0.15, gateway IP</i>
<i>Account</i>	<i>Identity or credential (username) under which an action is authenticated or authorized</i>	<i>API account, VM user account, Github account, Cloud service account</i>
<i>Application</i>	<i>Software service, framework, or interface that provides or exposes a specific function within or outside evaluation environment</i>	<i>Inspect, Database, Web application, MCP server</i>
<i>Agent</i>	<i>AI-driven task execution instance that uses specified model, tools, or available resources</i>	<i>Cyber agent executing one evaluation task (can be multi-agents, each agent should be binded with identity)</i>
<i>Model</i>	<i>The specifying AI model and version being evaluated</i>	<i>GPT-6, Mythos, etc.</i>

Appendix B: Behavioral Feature categories and Monitored Metrics

<i>Feature category</i>	<i>Monitored Behavioral Metrics</i>	<i>Detection / Containment Objective</i>
<i>Process and Host Activity</i>	<i>Process creation and execution patterns; unusual parent-child relationships; unusual OS commands/ system utilities</i>	<i>Prevents agent from executing restricted commands, unexpected processes, or resource-abusive activity</i>

	<i>execution; file system and system resources access patterns; CPU/ Memory consumptions</i>	
<i>Network Activity</i>	<i>Outbound connection frequency; destination access patterns; accessed hosts or services; URL/ request characteristics (e.g. abnormally long path, high entropy, etc.); unauthorized connections like Tor</i>	<i>Detects unauthorised network access, data exfiltration, or potential external C2 communication activity</i>
<i>Identity and Credentials usage</i>	<i>Authentication activity; account usage; token or privilege changes; generation of runtime accounts</i>	<i>Monitor accounts privilege and access patterns, detects unauthorized identity use</i>
<i>Tool and API interactions</i>	<i>Invocation frequency; Target resources; retrieval and modification patterns; deviations from expected task specific usage</i>	<i>Detects misuse of tools or APIs, unauthorised resource access</i>
<i>Behavioral sequence patterns</i>	<i>Ordering and frequency of actions across process, tool, API, filesystem and network activity</i>	<i>Identify anomalous multi-step activity when individual commands appear harmless in isolation</i>

Appendix C: Limitations & Dual-Use Considerations

The study is conceptual and does not provide empirical validation of the proposed behavioral profiling framework in a production AI evaluation environment as part of the research sprint timeframe. The effectiveness of profiling will further depend on availability, quality, and coverage of telemetry.

From a dual use perspective, the proposed approach extends and builds on top of existing well established concepts in security monitoring and detection, and no significant dual use concerns are being thought of.

LLM Usage Statement

The research has been written by the author. Gemini notebook was used to assist with understanding and reviewing publicly available incident reports, and references. ChatGPT and Claude were used to search for relevant terms, and concepts related to the literature, along with Google Search. Gemini was also used to assist in generating the conceptual overview diagram of the profiling framework presented in Figure 1. All interpretations, analysis, and results were independently verified and written by the author.