Independent Submission
Internet-Draft
Intended status: Informational
Expires: 17 October 2026

R. Sharif
CyberSecAI Ltd

15 April 2026

# Agent Event Behaviour Analysis (AEBA): A Framework for Behavioural Security Monitoring of Autonomous AI Agents
draft-sharif-aeba-00

## Abstract

This document specifies Agent Event Behaviour Analysis (AEBA), a framework for collecting, signing, exchanging, and analysing behavioural events produced by autonomous AI agents. AEBA is the agent-domain equivalent of User and Entity Behaviour Analytics (UEBA) as commonly deployed in enterprise Security Operations Centres. It defines a canonical event schema, signature binding to agent identity, baseline and peer-group exchange protocols, deviation signalling, detection rule structure, revocation mechanisms, and interoperability bindings for existing Security Information and Event Management (SIEM) event formats (syslog, CEF, LEEF). The framework is designed to compose with existing cryptographic primitives for agent identity, payment, and transport security, and to support cross-framework deployments in which agents produced by different runtimes must share a common behavioural observability surface.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 17 October 2026.

## Copyright Notice

Table of Contents

## 1. Introduction

Autonomous AI agents are being deployed into production in increasing numbers across payments, customer service, software engineering, healthcare, and industrial control. Enterprise Security Operations Centres have mature frameworks for monitoring the behaviour of human users through User and Entity Behaviour Analytics (UEBA) systems [UEBA-Gartner], but no equivalent standard exists for agent entities. This gap is becoming acute: agents operate at one to three orders of magnitude higher event rates than human users, carry cryptographic rather than password-based identities, and are subject to novel attack classes including prompt injection, delegation-chain abuse, and model-output-driven drift that have no human analogue.

This document defines a framework called Agent Event Behaviour Analysis (AEBA). AEBA specifies:

* A canonical, extensible event schema for agent behavioural telemetry.

* Binding of each event to a verifiable agent identity using existing cryptographic primitives.

* A baseline-exchange protocol enabling hosts and observers to share per-agent normal-behaviour models.

* A deviation-signalling mechanism producing per-event and per-sequence risk scores.

* A detection-rule structure aligned with kill-chain pattern matching as practised in UEBA.

* A revocation protocol for publishing cryptographic signals that a specific agent has been confirmed as misbehaving.

* Interoperability bindings to syslog [RFC5424], CEF [CEF], and LEEF [LEEF], enabling direct ingestion by existing SIEM products.

AEBA does not specify a complete SIEM or replace existing commercial offerings. It defines the common contract above which multiple vendors, open-source projects, and enterprise deployments can interoperate.

## 1.1. Relation to other drafts

AEBA is designed to compose with, not replace, existing work on agent security:

* Agent identity and key binding are specified by [I-D.sharif-agent-identity-framework].

* Event signing uses the same canonical-signing-string construction as [I-D.sharif-mcps-secure-mcp] and [I-D.sharif-agent-payment-trust].

* Per-message audit log format is specified by [I-D.sharif-agent-audit-trail]; AEBA events are a specific subtype suitable for behavioural analysis.

* Runtime attestation of agent state is specified by [I-D.sharif-attp-agent-trust-transport] and provides supplementary input to AEBA detection.

AEBA is complementary to, not dependent on, OWASP [OWASP-AST10] and [OWASP-MCP10] risk taxonomies, and provides a machine-readable observability layer on which those risk categories can be monitored in production.

## 2. Terminology and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are used throughout this document:

Agent: An autonomous or semi-autonomous software entity executing under a runtime (for example, a Large Language Model with attached tools) that performs actions on behalf of a user, organisation, or other agent.

AEBA Event: A structured, cryptographically signed record describing a single observable action taken by or on behalf of an Agent. Defined in Section 5.

Baseline: A statistical model of the normal behaviour of an Agent or a peer-group of Agents over a specified time window. Defined in Section 7.

Deviation Signal: A per-event or per-sequence risk score indicating the extent to which observed behaviour deviates from the applicable Baseline.

Detection Rule: A declarative specification matching one or more AEBA Events against a known malicious pattern. Defined in Section 9.

Peer Group: A set of Agents sharing a common purpose, deployment, framework, or role, against which individual Agent behaviour is compared.

Host Runtime: The environment in which an Agent executes and which is responsible for emitting AEBA Events describing Agent actions.

Collector: An AEBA role that receives AEBA Events from one or more Host Runtimes, verifies signatures, deduplicates, and forwards to one or more Analysers or SIEM systems.

Analyser: An AEBA role that computes Baselines, applies Detection

Rules, and emits alerts.

Trust Registry: An external authoritative endpoint that publishes and receives revocation signals for Agent identities, as specified by a referenced registry protocol.

Consumer: A downstream recipient of AEBA alerts or raw events, typically a Security Operations Centre, an incident-response workflow, or a policy-enforcement engine.

## 3. Threat Model

This section enumerates the threats AEBA is designed to detect or mitigate. Each threat is mapped to the AEBA mechanism addressing it.

### 3.1. In-scope threats

#### 3.1.1. T1. Identity compromise

An adversary obtains an Agent’s signing key and emits actions indistinguishable from the legitimate Agent.

AEBA mitigation: Baseline deviation. A compromised identity performing actions outside the legitimate Agent’s behavioural envelope triggers Deviation Signals. Revocation via the Trust Registry halts further acceptance.

#### 3.1.2. T2. Insider-style Agent misbehaviour

A legitimately deployed Agent, under the effect of prompt injection, context poisoning, supply-chain-compromised dependency, or adversarial orchestration, performs actions outside its intended scope.

AEBA mitigation: Behavioural-layer signals (see Section 8) detect deviations from the Agent’s own baseline and from its peer-group baseline. Pattern-matching Detection Rules map known attack chains to alert conditions.

#### 3.1.3. T3. Supply-chain compromise causing drift

A legitimate Agent’s dependencies, Skills, or underlying model are replaced with malicious versions; Agent behaviour subsequently drifts.

AEBA mitigation: Abrupt baseline shift not preceded by a signed deployment event triggers Deviation Signal. Event schema requires deployment-change events to be emitted and signed, enabling distinction between legitimate change and unauthorised drift.

#### 3.1.4. T4. Sybil agents

An adversary creates multiple agent identities to evade per-identity rate limits, spread malicious actions across identities, or forge peer-group support.

AEBA mitigation: Peer-Group membership requires cryptographic attestation signed by an authority recognised by Consumers. Rapid creation of many new Agent identities from a single keyholder triggers Detection Rules.

#### 3.1.5. T5. Event forgery

An adversary injects fake AEBA Events into a Collector in order to mislead baseline computation or to pollute audit records.

AEBA mitigation: All AEBA Events MUST be signed. Collectors MUST verify signatures before forwarding. Unsigned or invalidly-signed Events are rejected and themselves logged as a threat indicator.

#### 3.1.6. T6. Event suppression

An adversary suppresses AEBA Events to hide malicious behaviour, either at the Host Runtime, in transit, or at the Collector.

AEBA mitigation: Sequential AEBA Events MUST include a monotonic per-Agent sequence number and a hash reference to the previous signed Event (the Agent Event Chain). Gaps, non-monotonic sequences, or broken chain references trigger Detection Rules.

#### 3.1.7. T7. Baseline poisoning

An adversary gradually shifts an Agent’s baseline through low-deviation-per-event behaviour in order to render later malicious behaviour indistinguishable from the new normal.

AEBA mitigation: Baselines MUST retain a long-term reference distribution distinct from the short-term rolling baseline. Significant divergence between short-term and long-term baselines is itself a Detection Rule signal. Peer-group comparison provides an independent reference resistant to per-Agent poisoning.

#### 3.1.8. T8. Peer-group spoofing

An Agent falsely claims membership in a trusted peer group to benefit from the more permissive baseline of that group.

AEBA mitigation: Peer-Group membership attestations MUST be signed by a recognised authority, verified by Analysers, and revocable.

#### 3.1.9. T9. Detection evasion via context-window manipulation

A compromised Agent observes or infers detection thresholds and crafts behaviour to remain below them.

AEBA mitigation: Detection Rules SHOULD incorporate randomised sampling and non-deterministic threshold components to frustrate modelling. Analysers MUST NOT publish detection thresholds to Agents.

#### 3.1.10. T10. Cross-host replay

An adversary captures signed Events from Host Runtime A and replays them into the Collector serving Host Runtime B.

AEBA mitigation: AEBA Events MUST include Host Runtime identifier and a nonce; Collectors MUST reject duplicates and mismatched host bindings.

#### 3.1.11. T11. Time skew attacks

An adversary manipulates timestamps to alter sequence interpretation or to replay stale events.

AEBA mitigation: Events MUST carry UTC timestamps; Analysers MUST reject events outside a configurable freshness window (recommended: +/-300 seconds) at ingestion, subject to explicit backfill procedure.

#### 3.1.12. T12. Metadata smuggling

An adversary uses Event metadata fields (labels, tags) to smuggle encoded data (exfiltration channel).

AEBA mitigation: Metadata fields MUST be subject to schema validation and per-field length limits. Analysers SHOULD compute entropy statistics over metadata fields and alert on outlier entropy.

#### 3.1.13. T13. Delegation-chain abuse

A parent Agent spawns a child Agent that performs actions the parent's own policy would have denied.

AEBA mitigation: Delegation events MUST be emitted and signed (parent-signs-child). Child Agent capability scope MUST NOT exceed parent Agent's scope; Analysers MUST enforce scope-inheritance limits and alert on violations.

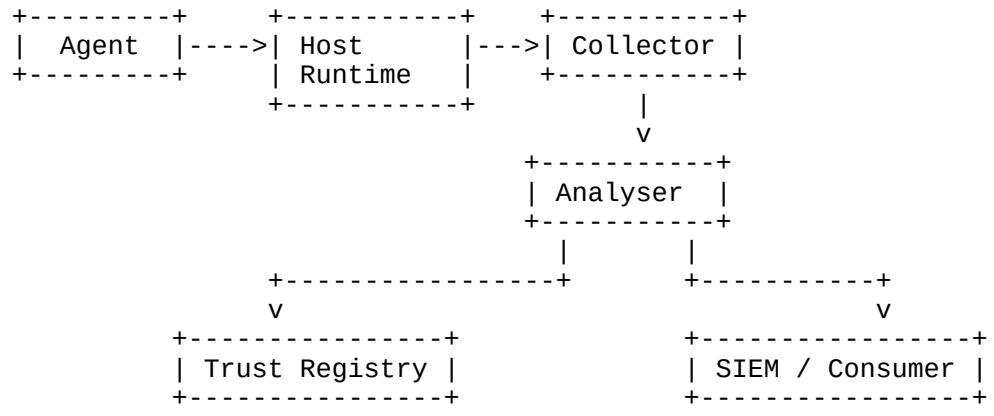
### 3.2. Out-of-scope threats

AEBA does not address:

* Loss of confidentiality in Agent model internals (a model-protection problem);
* Denial of service at the Host Runtime or Collector layer (handled by conventional network and infrastructure controls);
* Physical-layer compromise of signing key storage (addressed by hardware security module practice).

## 4. AEBA Architecture

The AEBA architecture comprises five roles.




Role descriptions:

1. The *Agent* performs actions. Its runtime-observable actions produce AEBA Events.

2. The *Host Runtime* observes Agent actions and emits signed AEBA Events. It MAY perform this signing itself or it MAY forward Agent-self-signed Events.

3. The *Collector* receives Events from one or more Host Runtimes, verifies signatures, deduplicates, and forwards.

4. The *Analyser* computes Baselines, applies Detection Rules, and emits alerts.

5. The *Trust Registry* publishes revocation and peer-group attestation signals; the Analyser consults the Registry when evaluating identity binding.

Any AEBA implementation MAY combine these roles. The protocol defines the contract between roles; it does not prescribe deployment topology.

## 5. Event Schema

An AEBA Event is a JSON object with the following mandatory fields and an extensible set of optional type-specific fields.

### 5.1. Mandatory fields

* aeba: (string) protocol version. Current value: "1.0".

* id: (string) UUID v4 uniquely identifying this Event.

* agentId: (string) identifier of the Agent, matching the keyid field in the signature envelope.

* hostRuntimeId: (string) identifier of the Host Runtime from which the Event originated.

* ts: (integer) Event timestamp in seconds since Unix epoch (UTC).

* seq: (integer) monotonically increasing per-Agent sequence number.

* prevHash: (string) hex-encoded SHA-256 digest of the canonical form of the previous Event emitted by this Agent, or 64 zeros for the first Event.

* type: (string) Event type, drawn from the registered AEBA Event Type registry (see Section 15).

* category: (string) one of "identity", "tool", "payment", "skill", "delegation", "deployment", "compliance", "custom".

* payload: (object) type-specific fields.

## 5.2. Envelope fields

The signed envelope wrapping the Event contains:

* alg: (string) signing algorithm identifier. Initial values: "ES256" (mandatory-to-implement).

* keyid: (string) identifier of the signing key.

* sig: (string) base64-encoded signature over the canonical form (see Section 6).

* pubkeyHint: (string, optional) URL at which the verification key may be retrieved.

## 5.3. Example

```
{
  "aeba": "1.0",
  "id": "7c6e5b94-f8a1-4e9d-a3c2-b0d7f2e1c5a8",
  "agentId": "agent:acme-payments-01",
  "hostRuntimeId": "host:prod-us-east-cluster-7",
  "ts": 1744746000,
  "seq": 14823,
  "prevHash": "3f4a8b2c1d7e9f0a6b5c4d3e2f1a0b9c...",
  "type": "tool.call",
  "category": "tool",
  "payload": {
    "toolName": "agentpass_pay",
    "toolArgs": {
      "rail": "x402",
      "to": "merchant-hash:8a7f...",
      "amount": 2500,
      "currency": "USD"
    },
    "durationMs": 142,
    "outcome": "allowed"
  }
}
```

## 5.4. Event type categories

Each AEBA Event MUST declare a category. Categories are:

* identity -- Agent identity events: keypair issuance, key rotation, credential verification, delegation issuance.

* tool -- Tool invocation events: tool call, tool response, tool selection decision.

* payment -- Payment-related events: payment initiated, payment settled, sanctions check performed.

* skill -- Skill events: skill loaded, skill verified, skill rejected.

* delegation -- Agent-to-agent events: child agent spawn, delegated authority issued, delegated authority exercised.

* deployment -- Runtime environment change events: code deployed, dependency updated, model version change.

* compliance -- External compliance signals: KYC completed, AML check, regulatory report.

* custom -- Implementation-specific events. Custom events MUST use reverse-DNS-scoped type names to prevent collision.

## 6. Signing and Identity Binding

AEBA Events MUST be cryptographically signed. The signing scheme reuses the canonical-signing-string construction of [I-D.sharif-mcps-secure-mcp] and [I-D.sharif-agent-payment-trust] to enable shared verification infrastructure.

### 6.1. Canonical form

The canonical form of an AEBA Event for signing is:

```
<ts> <seq> <agentId> <hostRuntimeId>
   <sha256-hex(canonical-json(event))>
```

Where canonical-json(event) is the JCS-canonicalised JSON form of the Event object, and fields are joined by single space characters.

### 6.2. Signature algorithms

The mandatory-to-implement algorithm is ECDSA P-256 with SHA-256 (JWA ES256) as defined in [RFC7518]. Implementations MAY support additional algorithms consistent with existing agent-identity stacks (for example, Ed25519).

### 6.3. Identity binding

The keyid field in the signature envelope MUST equal the agentId field in the Event body, or MUST be a designated signing key for that Agent as registered with the applicable Trust Registry.

Host Runtimes that sign Events on behalf of Agents (server-side signing) MUST carry a signedBy field in the envelope distinguishing the signing identity from the Agent identity, and MUST bind the Host Runtime signing key to the Agent's declared Host Runtime via the Trust Registry.

## 7. Baseline and Peer-Group Model

### 7.1. Baselines

A Baseline is a statistical model of normal behaviour for an Agent or Peer Group over a given time window. AEBA defines two baseline classes:

* *Short-term baseline*: rolling window (recommended: 24 hours). Sensitive to recent behaviour; used for near-real-time anomaly scoring.

* *Long-term baseline*: reference window (recommended: 90 days). Resistant to short-term drift; used for detecting baseline poisoning (see Section 3.1.7).

Implementations MUST maintain both classes and MUST compute deviation against both.

### 7.2. Baseline fields

A minimal Baseline record comprises:

* eventRatePerMinute: distribution (mean, stddev, p50, p95, p99)

* eventCategoryMix: distribution across categories

* toolDiversity: distinct tools invoked per unit time

* costPerTask: distribution of token / compute cost per logical task

* paymentRecipientConcentration: Herfindahl index or equivalent

* peerGroupMembership: attested group(s)

* operationalSchedule: if applicable (distributions by hour-of-day, day-of-week)

Implementations MAY extend this set; extensions MUST use reverse-DNS scoping.

### 7.3. Peer-group model

A Peer Group is a named set of Agent identities plus an authority attestation signed by a recognised Peer-Group authority. The attestation MUST specify:

* peerGroupId: stable identifier

* purpose: human-readable group purpose

* members: list of agentId values in the group

* issuedBy: identifier of attesting authority

* validFrom / validUntil: validity window

* attestation: signature by the authority

Peer-Group authorities are trust anchors; their identity and authorisation MUST be established out of band (typically by the Trust Registry operator).

### 7.4. Baseline exchange

Baselines MAY be exchanged between Host Runtimes and Analysers so that an Agent deployed in multiple environments carries its behavioural history with it. Baseline-exchange messages MUST be signed by the emitting party, MUST include a validity window, and MUST reference the constituent Agent or Peer Group identifiers.

## 8. Deviation Signalling

For each incoming Event, the Analyser computes a deviation score bounded in [0.0, 1.0] where 0.0 denotes behaviour identical to baseline and 1.0 denotes maximal deviation observed during the baseline window.

### 8.1. Per-event score

The per-event deviation score is a weighted composite over:

* Event rate component (current rate vs baseline distribution)

* Category-mix component (divergence of current category from expected)

* Tool-choice component (likelihood of tool selection under baseline)

* Peer-group component (agreement of Event with peer baseline)

The weighting is implementation-defined; implementations SHOULD document the weighting and MUST publish it to Consumers for audit.

### 8.2. Sequence score

Sequences of Events are scored collectively against Detection Rules (see below). Sequence scores are not bounded to [0.0, 1.0]; they are Rule-defined alert conditions.

### 8.3. Signalling format

Deviation and alert signals are themselves AEBA Events with category "custom" and reverse-DNS type names such as com.example.aeba.alert. They are signed by the Analyser and forwarded to Consumers.

## 9. Detection Rules and Kill-Chain Patterns

Detection Rules are declarative specifications of known malicious patterns. A Detection Rule comprises:

* ruleId: stable identifier (reverse-DNS scoped)

* description: human-readable description

* scope: applicable Agent categories or peer groups

* pattern: pattern specification (event sequence, rate, divergence)

* severity: "low" | "medium" | "high" | "critical"

* action: recommended Consumer action on match

### 9.1. Mandatory rule categories

Implementations MUST support at least the following rule categories:

* *Identity-layer*: signature failures, concurrent-keyid detection, rotation-frequency anomalies.

* *Behavioural-layer*: event-rate spike, tool-diversity spike, category-mix shift.

* *Economic-layer*: payment-velocity anomaly, rail-mix shift, recipient-concentration spike, failed-payment rate.

* *Relational-layer*: delegation-depth anomaly, sub-agent spawn rate, cross-framework interaction.

* *Compliance-layer*: sanctions-match rate, KYC-failure rate, trust score degradation.

### 9.2. Rule interchange

Detection Rules are exchanged between parties as signed JSON documents. The interchange format is defined in Section 15.3.

## 10. Revocation Protocol

When an Agent is confirmed as misbehaving, the applicable Trust Registry MUST issue a revocation signal. AEBA revocation signals MUST:

* be signed by the Trust Registry authority;

* name the revoked Agent keyid;

* specify revocation scope: "all_events", "future_events_only", or "specific_type:<type>";

* specify effective time (revokedFrom);

* specify reason ("key_compromise", "malicious_behaviour", "policy_violation", "retired").

Consumers MUST check revocation state for each Agent either at Event-ingestion time (strict) or at alert-evaluation time (lazy). Implementations SHOULD maintain a local cache with a maximum freshness of 24 hours for high-criticality flows and 7 days otherwise.

Revocation cascades follow delegation chains: revocation of a parent Agent keyid MUST cause implicit revocation of all descendants.

## 11. Federation and Cross-Host Exchange

AEBA supports federated deployments in which Agents move between Host Runtimes operated by different organisations (for example, a Skill published centrally and invoked locally). Federation SHALL use:

* cross-host baseline exchange as defined above;
* Trust Registry references resolvable via HTTPS;
* mutual attestation between federating Host Runtimes.

Federated Consumers SHOULD NOT rely on unsigned baseline imports.

## 12. Interoperability Bindings

### 12.1. Syslog (RFC 5424)

AEBA Events MAY be emitted as Syslog [RFC5424] messages. The APP-NAME MUST be "aeba". Structured data MUST include the AEBA Event JSON as a single SD-ELEMENT:

```
<14>1 2026-04-15T08:00:00Z host.example aeba - AEBA
  [aeba@99999 event="<base64-canonical-json>"
   sig="<base64-signature>"] AEBA event
```

### 12.2. CEF (Common Event Format)

AEBA Events MAY be emitted as CEF records with:

* Device Vendor: "AEBA"
* Device Product: implementation name
* Device Version: protocol version ("1.0")
* Signature ID: AEBA Event type
* Name: human-readable Event description
* Severity: mapped from deviation score (0-10)

Structured AEBA fields MUST be carried in CEF extension fields using reverse-DNS naming.

### 12.3. LEEF (Log Event Extended Format)

AEBA Events MAY be emitted as LEEF records following the same field conventions as CEF.

## 13. Security Considerations

AEBA is itself a security-critical protocol. The following considerations apply.

### 13.1. Signing is mandatory

All AEBA Events MUST be signed. Unsigned events compromise the entire analytical model and MUST be rejected at ingestion.

### 13.2. Freshness and replay

Events MUST include a timestamp and sequence number. Analysers MUST reject events outside a freshness window (recommended: +/-300 seconds) and MUST reject replayed prevHash sequences.

### 13.3. Key management

Agent signing keys MUST be stored in accordance with host platform best practice (hardware security module, enclave, OS keystore). Loss of a signing key MUST trigger revocation.

### 13.4. Baseline integrity

Baselines are themselves security-critical artifacts. Exchange between parties MUST be signed. Long-term baselines MUST be preserved through key rotation and custodial transitions.

### 13.5. Detection evasion

Implementations MUST NOT publish detection thresholds or rule weightings to unauthenticated parties. Detection Rule interchange MUST be between trusted parties.

### 13.6. Supply-chain threats

AEBA cannot detect Agent behaviour that is permanently within baseline but intrinsically malicious. AEBA is a complement to, not a replacement for, pre-deployment supply-chain verification (Skill signing, model attestation, dependency verification).

### 13.7. Abuse of AEBA data

AEBA Events describing Agent actions may reveal sensitive information about the Agent’s operator, the Agent’s users, or downstream systems. Access to AEBA data MUST itself be access-controlled and auditable.

## 14. Privacy Considerations

AEBA Events can contain personally identifiable information (PII) and confidential business data, including customer identities, payment recipients, document contents, and authentication artifacts. Implementations MUST:

* minimise PII in Event payloads, preferring hashed or tokenised identifiers where correlation is sufficient;

* apply encryption at rest to AEBA stores;

* implement retention policies consistent with applicable regulation (for example, GDPR Article 5(1)(e), CCPA, UK Data Protection Act);

* support the right of erasure (where legally required) by cryptographic redaction of AEBA stores, recognising that signed chain integrity may be partially lost on erasure.

Baselines themselves are subject to privacy analysis. A Baseline may reveal aggregate user behaviour patterns and SHOULD be treated as confidential.

## 15. IANA Considerations

This document requests IANA to establish two new registries.

### 15.1. AEBA Event Type Registry

A registry named "AEBA Event Types" with the following columns:

* type: the event type string (e.g., "tool.call", "payment.initiated")

* category: one of the defined categories

* description: human-readable description

* reference: defining document

Registration policy: Specification Required for the reserved categories; First Come First Served for reverse-DNS-scoped custom types.

Initial entries corresponding to the categories defined in Section 5.

### 15.2. AEBA Signing Algorithm Registry

A registry named "AEBA Signing Algorithms" listing acceptable alg values. Initial entries:

* ES256 -- ECDSA P-256 with SHA-256 (from [RFC7518], mandatory)

* EdDSA -- Edwards-curve Digital Signature Algorithm (optional)

### 15.3. Rule Interchange Format

IANA is requested to register a media type application/aeba-rule+json for the Detection Rule interchange format defined in this document.

## 16. References

## 17. References

### 17.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <https://www.rfc-editor.org/info/rfc5424>.

[RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <https://www.rfc-editor.org/info/rfc7518>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9421] Backman, A., Ed., Richer, J., Ed., and M. Sporny, "HTTP Message Signatures", RFC 9421, DOI 10.17487/RFC9421, February 2024, <https://www.rfc-editor.org/info/rfc9421>.

### 17.2. Informative References

[UEBA-Gartner]
Gartner, "Market Guide for User and Entity Behavior Analytics", 2023.

[CEF] Micro Focus ArcSight, "Common Event Format", n.d..

[LEEF] IBM QRadar, "Log Event Extended Format (LEEF)", n.d..

[MITRE-ATTACK]
MITRE Corporation, "MITRE ATT&CK Framework", n.d..

[OWASP-AST10]
OWASP Foundation, "OWASP Agentic Skills Top 10", n.d..

[OWASP-MCP10]
OWASP Foundation, "OWASP MCP Top 10", n.d..

[I-D.sharif-mcps-secure-mcp]
Sharif, R., "MCPS: Cryptographic Security Layer for the Model Context Protocol", Work in Progress, Internet-Draft, draft-sharif-mcps-secure-mcp-00, 14 March 2026, <https://datatracker.ietf.org/doc/html/draft-sharif-mcps-secure-mcp-00>.

[I-D.sharif-agent-payment-trust]
Sharif, R., "Trust Scoring and Identity Verification for Autonomous AI Agent Payment Transactions", Work in Progress, Internet-Draft, draft-sharif-agent-payment-trust-00, 25 March 2026, <https://datatracker.ietf.org/doc/html/draft-sharif-agent-payment-trust-00>.

[I-D.sharif-agent-audit-trail]
Sharif, R., "Agent Audit Trail: A Standard Logging Format for Autonomous AI Systems", Work in Progress, Internet-Draft, draft-sharif-agent-audit-trail-00, 29 March 2026, <https://datatracker.ietf.org/doc/html/draft-sharif-agent-audit-trail-00>.

[I-D.sharif-agent-identity-framework]
Sharif, R., "Agent Identity Framework: Trust and Identity for Autonomous AI Agents", Work in Progress, Internet-Draft, draft-sharif-agent-identity-framework-00, 6 April 2026, <https://datatracker.ietf.org/doc/html/draft-sharif-agent-identity-framework-00>.

[I-D.sharif-attp-agent-trust-transport]
Sharif, R., "ATTP: Agent Trust Transport Protocol for Secure Agent-to-Server Communication", Work in Progress, Internet-Draft, draft-sharif-attp-agent-trust-transport-00, 30 March 2026, <https://datatracker.ietf.org/doc/html/draft-sharif-attp-agent-trust-transport-00>.

## Appendix A. Worked Example: Detecting Prompt-Injection-Driven Drift

This appendix illustrates AEBA in action against an attack described in Section 3.1.2 (Insider-style Agent misbehaviour).

Scenario: An e-commerce Agent’s baseline shows it calls payment.pay with rail=stripe 95% of the time, currency=USD 90% of the time, with amount distributed log-normally around $40. Average event rate: 8 per minute over business hours.

Adversary injects a prompt that causes the Agent to attempt a rail=x402 payment to a newly-observed recipient, amount=$5000, repeatedly over 30 seconds.

AEBA detection:

1. Per-event deviation score: the rail choice is a 5% rail with probability weight 0.05, producing high component score.

2. Recipient concentration: recipient is new to the agent; Herfindahl shift component adds further weight.

3. Event rate: 15 events in 30 seconds (30/min) vs baseline 8/min triggers rate rule.

4. Category-mix: normal mix has payment = 40%; this burst is payment = 100%; mix-shift rule triggers.

5. Composite sequence score exceeds alerting threshold; Analyser emits alert of severity "high" with recommended action "suspend_agent".

## Appendix B. Worked Example: Detecting Payment Rail Shift Attack

[Illustration omitted for brevity; pattern follows that of the previous appendix but focused on Economic-layer detection.]

## Appendix C. Baseline Aggregation Canonical Form

[Specification of the canonical form for Baseline records, ensuring deterministic hashing and inter-party agreement. Omitted for the -00 draft.]

## Author’s Address

Raza Sharif
CyberSecAI Ltd
Email: raza.sharif@outlook.com