Contents

IntentGate SIEM Runbook — Splunk

Audience. SOC analysts and Splunk admins operating IntentGate in production or evaluating the lab. This runbook is the canonical reference for the audit-event schema, the queries you'll actually run, and the HEC configuration that puts events into your index.

Gateway audit schema: v6 (see `schema_version` field on every event). **Document version:** 1.0 (May 2026).

---

1. What you're getting

IntentGate emits one structured audit event for every `tools/call` request that traverses the gateway. The event records the gateway's verdict (allow / block / escalate), which check produced it (capability, intent, policy, budget, or upstream forward), the agent that made the call, the tool it tried to invoke, and enough correlation metadata to reconstruct an incident end-to-end.

The shape is OCSF-lite: a flat JSON document with `lowercase_underscore` field names so it merges cleanly into Splunk CIM, Elastic Common Schema, or full OCSF without a custom mapper. The gateway ships events over Splunk's HTTP Event Collector (HEC) in newline-delimited batches.

Field names are stable across minor versions. New fields are added at the end with `omitempty`, so existing dashboards and saved searches continue to work after a gateway upgrade.

---

2. Field schema

Every event arrives in Splunk wrapped in a standard HEC envelope (`time`, `host`, `source`, `sourcetype`, `index`) with the gateway's payload nested under the `event` key. Splunk's JSON sourcetype flattens this so you reference fields directly — `decision`, `tool`, `agent_id`, etc. — not `event.decision`.

Field	Type	Always present	Meaning
`ts`	RFC3339Nano string	yes	Event time in UTC. Splunk's `_time` is also populated from the HEC `time` field.
`event`	string	yes	Always `intentgate.tool_call` in v6. Routing key for SIEM correlations.
`schema_version`	string	yes	Currently `"6"`. Compare against this doc's header to spot drift.

Field	Type	Always present	Meaning
`decision`	string	yes	`allow`, `block`, or `escalate`. Note: NOT deny — that's a common mistake.
`check`	string	no	Which gate produced the decision: `capability`, `intent`, `policy`, `budget`, `upstream`. Empty on a clean allow that passed every stage.
`reason`	string	no	Free-text rationale (e.g., `"budget exhausted: 0 tokens remain"`).
`tenant`	string	yes (default `"default"`)	Trust-domain namespace from the verified capability token. Filter on this in multi-tenant deployments.
`agent_id`	string	usually	The AI agent making the call. From the capability token, not the request.
`session_id`	string	no	Per-session correlation key when the agent maintains state.
`tool`	string	yes	Tool the agent tried to invoke (e.g., `read_invoice`).
`arg_keys`	array of strings	no	Names of arguments the agent passed. Privacy-preserving: values are not included by default.
`arg_values`	object	no	Redacted argument values. Populated only when `INTENT-GATE_AUDIT_PERSIST_ARG_VALUES` is set on the gateway. Numbers/booleans survive; strings are nulled.
`capability_token_id`	string	usually	JTI of the capability token. Correlate to the issuance event.
`root_capability_token_id`	string	no	JTI of the chain root for delegated tokens. Equal to `capability_token_id` for root tokens.
`caveat_count`	int	no	Number of attenuations on the token chain.

Field	Type	Always present	Meaning
`pending_id`	string	no	Links an `escalate` event to its eventual `allow`/`block` resolution.
`decided_by`	string	no	Operator identity who resolved a step-up / approval flow.
`intent_summary`	string	no	One-line summary the intent extractor pulled from the call. Never the raw prompt. The string `[stub: no rule matched]` appears when the extractor had no rule for this tool — a signal you should add coverage.
`latency_ms`	int	yes	Wall-clock time the gateway spent on this request.
`remote_ip`	string	no	Source address of the agent as seen by the gateway.
`upstream_status`	int	no	HTTP status returned by the upstream tool server. `0` means the gateway was in stub mode or failed before any response.
`requires_step_up`	bool	no	Policy flagged this call as high-risk; route to your alerting pipeline.
`elevation_id`	string	no	JIT-elevation row id for the operator who issued the call. Empty for direct agent traffic. Join to `console_elevations` for “what was approved when X happened.”

HEC envelope fields (Splunk-injected, not gateway-emitted):

Field	Default	Notes
`source`	`intentgate`	Configurable via `INTENTGATE_SIEM_SPLUNK_SOURCE`.
`sourcetype`	`_json`	Configurable via `INTENTGATE_SIEM_SPLUNK_SOURCETYPE`.

Field	Default	Notes
`index`	(token default)	Set explicitly via `INTENTGATE_SIEM_SPLUNK_INDEX` if you want events in a non-default index.
`host`	Splunk-derived	Gateway does not set host; Splunk attributes from the HEC connection.

---

3. Sample event

What one event looks like in your index, pretty-printed:

```
{
  "ts": "2026-05-18T17:59:56.182441Z",
  "event": "intentgate.tool_call",
  "schema_version": "6",
  "decision": "allow",
  "check": "upstream",
  "tenant": "default",
  "agent_id": "siem-victory",
  "tool": "read_invoice",
  "arg_keys": ["id"],
  "capability_token_id": "01HXYZ...",
  "intent_summary": "siem victory [stub: no rule matched]",
  "latency_ms": 14,
  "upstream_status": 200
}
```

A blocked call looks much the same but with `decision: "block"`, a populated `check` (e.g., `"budget"`), and a `reason` explaining what failed.

---

4. Canonical SPL queries

These five queries cover ~90% of the questions a SOC analyst asks of an IntentGate event stream. Paste them into Splunk search, adjust the index name, save as alerts if they matter to you.

> Replace `index=main` with whichever index your HEC token writes to.

4.1 All blocked calls in the last 24 hours, with reason

```
index=main source=intentgate decision=block
| stats count by check, reason, tool, agent_id
| sort -count
```

What it answers: “Which agents got rejected, by which check, why, on what tool?” The first query you run when a customer asks why their agent isn’t working.

4.2 Budget breaches

```
index=main source=intentgate check=budget decision=block
| stats count by agent_id, tenant, tool
| sort -count
```

What it answers: “Who is hitting their token budget, on which tool?” Useful for capacity planning and for catching runaway agents.

4.3 Top agents by activity (last 7 days)

```
index=main source=intentgate earliest=-7d
| stats count as calls,
        count(eval(decision="allow")) as allowed,
        count(eval(decision="block")) as blocked,
        count(eval(decision="escalate")) as escalated
        by agent_id
| eval block_rate = round(blocked * 100.0 / calls, 1)
| sort -calls
```

What it answers: "Which agents are loudest, and what fraction of their traffic is being blocked?" High block-rate agents are either misbehaving or under-provisioned.

4.4 Top tools by activity (last 7 days)

```
index=main source=intentgate earliest=-7d
| stats count as calls,
        avg(latency_ms) as avg_latency,
        count(eval(decision="block")) as blocked
        by tool
| eval avg_latency = round(avg_latency, 0)
| sort -calls
```

What it answers: "Which tools are most popular, slowest, most likely to get blocked?" Triage input for the platform team.

4.5 Intent-extractor coverage gaps

```
index=main source=intentgate intent_summary="*[stub: no rule matched]*"
| stats count by tool, agent_id
| sort -count
```

What it answers: "Which tool calls are getting past the gateway without intent extraction?" Each hit is a coverage gap to file with the IntentGate ops team — every additional rule tightens the four-check pipeline.

4.6 (Bonus) Step-up flagged but allowed

```
index=main source=intentgate requires_step_up=true decided_by=allow
| table _time, agent_id, tool, decided_by, elevation_id, reason
| sort -_time
```

What it answers: "Where did our policy say 'high risk' but we let it through anyway?" The events you most want a human to glance at.

---

5. HEC setup recap

Your Splunk admin configures one HEC token; the gateway points at it via three env vars.

On Splunk: 1. *Settings* → *Data inputs* → *HTTP Event Collector* → New Token. 2. Name: `intentgate`. Source type: `_json`. Index: `main` (or a dedicated index — `siem_intentgate` is a common convention). 3. Save the token value — you'll paste it into the gateway's env. 4. Confirm the HEC port is reachable from the gateway. Default 8088. TLS is recommended in production; the lab runs HTTP for simplicity.

On the gateway (env vars):

Variable	Required	Example
`INTENTGATE_SIEM_SPLUNK_URL`	yes	`https://splunk.example.com:8088/serv`
`INTENTGATE_SIEM_SPLUNK_TOKEN`	yes	the HEC token from step 3

Variable	Required	Example
`INTENTGATE_SIEM_SPLUNK_INDEX`	Optional	`siem_intentgate` (defaults to the token's default index)

The gateway validates both URL and token at startup. Either both are set or both are empty; a half-configured emitter is rejected. Tokens are never logged or exposed via the admin status endpoint.

Smoke test the connection:

```
curl -k https://splunk.example.com:8088/services/collector/health
# Expect: {"text":"HEC is healthy","code":17}
```

If that returns healthy and your gateway is wired correctly, events will appear in the configured index within seconds.

---

6. Other SIEMs

Same JSON event shape, different transport.

- **Datadog Logs.** Set `INTENTGATE_SIEM_DATADOG_API_KEY` and (optionally) `INTENTGATE_SIEM_DATADOG_SITE`. The gateway batches over Datadog's logs intake.
- **Microsoft Sentinel.** Set `INTENTGATE_SIEM_SENTINEL_WORKSPACE_ID` and `_KEY`. Events land via the Log Analytics Data Collector API.
- **Anything else (Elastic, Chronicle, Sumo).** Tail the gateway's stdout — events emit as newline-delimited JSON in the same shape — and forward with your existing log shipper (Vector, Fluent Bit, Promtail).

See `lab/.env.example` for the full env-var list and `gateway/internal/siem/` for the source of truth on supported emitters.

---

7. Updates

This runbook tracks the gateway's audit schema. When the schema version bumps, this document is updated and re-published. The current version is shown at the top of the document and on every event under `schema_version`. If those don't match, check for a newer runbook before re-tuning your dashboards.

Source: `lab/docs/siem-runbook.md` in the IntentGate repository. Questions: `support@intentgate.app`.