

Audit, Logging and Monitoring AI Usage

For security and leads · ~10 min read

Metadata over content

The foundational decision for AI audit logging is what you choose to record. The instinct from a decade of application logging is to capture as much as possible: request body, response body, any attachment, the full chain of steps. That instinct is wrong for AI usage, and the reason is not subtle. Prompts and responses are the highest-entropy, most sensitive text your users generate in a day. They contain half-formed ideas, copy-pasted customer records, draft emails about HR situations, fragments of source code, API keys somebody forgot were in their clipboard, and occasionally entire contracts. The moment you persist that text into a log store, you have created a new data-at-rest asset that is more sensitive than almost any other log you already manage — and, because it was "just logs," it is probably less protected.

The alternative is to log the *fact* of the interaction and the *metadata around it*, and not the content itself. You record that user X called model Y at timestamp T with a prompt of length N on data classified as C, and that the outcome was allow, block, or redact. You record which policy version was in force and which classifier fired. You do not record what the user actually typed, and you do not record what the model sent back. That gives you everything you need to investigate an incident, everything you need to prove a control was working, and none of the content liability.

Do not create a new data lake of prompts.

This is not about being lazy with logging. It is about understanding that the content of an AI prompt is not log data. It is user data, of the most sensitive kind, and it belongs under the same controls as any other piece of sensitive user data — not tossed into an audit bucket because the word "audit" made it feel safe.

What to capture

The metadata set below is the minimum useful record per AI interaction. Every field is either an identifier, a classification, or a decision. None of them is content.

- **User** — authenticated identity, team, role.
- **Tool** — which client was used: web chat, CLI, IDE plugin, agent framework, internal wrapper.
- **Model** — provider, model family, and specific version identifier.
- **Timestamp** — start and, where available, end of the interaction.
- **Prompt length** — token or byte count. The size, not the text.
- **Data classification tags** — what the DLP or classifier saw: PII, secrets, source code, customer data, none.
- **Outcome** — sent, blocked, redacted, or sent-with-modification.
- **Policy version** — the ruleset identifier that was in force at the time.

That set answers the investigative questions that actually matter: who, when, with what tool, on what kind of data, under which rules, with what outcome. If a regulator, an incident responder, or an internal reviewer asks what happened, those eight fields are what they are asking about. None of them require you to keep the prompt.

What not to capture

The list of things to explicitly keep out of your AI audit logs is short and strict:

- **Raw prompt text.** Always contains more than the user meant to include.
- **Raw responses.** Often echoes or elaborates on the prompt content, so it carries the same risk.
- **Attached files.** Your audit log should not become a mirror of every document anybody ever uploaded to an AI tool.
- **Chain-of-thought snippets.** They contain the same content risk plus they are typically not meaningful out of context, so the investigative value is low and the liability is high.

If you would not store a value in plaintext inside your CRM, your ticketing system, or your source control, do not store it in your AI audit log either. "It is in the log" is not a separate privacy classification.

Enterprise admin consoles

The practical good news in 2026 is that the major providers have already made the right design choice for you. ChatGPT Enterprise exposes an admin API for workspace audit events. Claude Team and Claude Enterprise offer audit log export covering user, tool, and policy events. Google Workspace surfaces Gemini events through the same audit pipeline it has used for Docs, Gmail, and Drive for years. All three are metadata-shaped by default — they tell you *that* something happened and *who* did it, without handing you the full prompt text as a default field.

The work on your side is to actually wire those feeds into your SIEM, and to do it on day one of deployment rather than during the first incident. The common failure mode is not that the feed is missing — it is that the feed exists, nobody turned it on, and nobody built any detections against it. The enterprise audit stream is where you should be building most of your monitoring, because its shape already matches the metadata-over-content rule. Treat it the same way you treat IdP logs, endpoint telemetry, and egress proxy records: a routine, always-on source that your detection team owns.

Detection rules

Detection splits cleanly into two tiers. The first is "AI was used" — knowing where AI shows up in your environment at all. The second is "AI was misused" — finding the subset of that usage that violates policy. You have to build the first tier before the second is meaningful, because you cannot alert on anomalies in a signal you are not collecting.

For the **AI-was-used** tier, the workhorse signals are traffic to known provider domains (from endpoints, egress proxies, and DNS), process and telemetry signals from known AI CLIs and agent frameworks, and beacon traffic from IDE plugins. This does not tell you that anything bad happened. It tells you who is using AI, how often, and through which channel, which is the baseline for everything else.

For the **AI-was-misused** tier, build detections in this priority order. First, *unapproved provider*: traffic to any AI API that is not on your sanctioned list. This is the highest-signal, lowest-noise rule and should alert loudly. Second, *volume anomalies*: a user sending ten times their normal token volume, multi-megabyte attachments to an AI endpoint, or a sudden spike in agent calls. Third, *off-hours spikes*: heavy AI usage at times when the account has no normal activity pattern. Fourth, *DLP classification hits*: a sanctioned provider call whose classifier flagged PII, secrets, or customer data, recorded as a class label in the metadata rather than as the text itself. The priority order matters because the higher tiers are cheaper to investigate and produce less false positive fatigue.

Retention

Metadata audit logs for AI usage should live for 90 to 180 days. That window is long enough to cover the vast majority of incident timelines — most investigations reach back weeks, not years — and short enough that the store does not become its own high-value target. Within that window you can split further: block events and policy decisions can live at the long end of the range for trend analysis, while high-volume detail events can age out sooner.

Content retention for AI prompts and responses should be zero, and the way you achieve a zero-retention policy is by never logging the content in the first place. A retention policy that says "we will delete prompt content after 30 days" is worse than one that says "we never captured prompt content," because the first one still has a 30-day breach window and the second one has no window at all.

Privacy balance

There is a persistent intuition, especially among leaders who have seen AI incidents in the news, that the responsible answer is to log everything and review it. It is not. Over-logging prompts creates three problems at once. It creates a breach target: the prompt archive is now one of the most sensitive stores in the company, and compromising it is worth the attacker's time. It creates an insider-risk problem: anybody with query access to the store can read what staff were thinking about, not just what they did. And it creates a privacy problem: staff have reasonable expectations about what gets

captured when they are working things out in a scratchpad, and a surveillance-grade prompt archive violates those expectations even if no rule is formally broken.

The balanced position is the one this lesson argues for throughout. You log the metadata that lets you investigate, attribute, and enforce policy. You do not log the content that turns your audit trail into the thing you have to defend next. Good AI audit is boring — eight metadata fields, a feed into a SIEM, four detection rules, and a 90-to-180-day retention window. Boring is the goal.

Rules to remember

- Log metadata, not content. Never stand up a prompt lake.
- Capture: user, tool, model, timestamp, length, classification, outcome, policy version.
- Pipe enterprise admin console audits into your SIEM on day one.
- Detection rule priority: unapproved provider > volume anomalies > content-class hits.
- Retain AI audit logs 90–180 days. Longer is a liability.
- Over-logging prompts is its own privacy and insider-risk problem.