



GUIDE • AI GOVERNANCE

AI governance audit-readiness guide

Evidencing control over agentic AI

01 The accountability gap

Most organizations cannot yet evidence what their AI agents do, and an audit tests exactly that. In Salt's 2H 2026 survey of more than 380 security and technology professionals, 49.8% confirmed or suspected that an agent took an action in the past year that the organization did not intend, expect, or authorize. Only 12.5% can consistently trace an agent's full path from instruction through tools and APIs to outcome.

A policy states what should happen. An auditor asks for proof of what did. Agents make that proof harder because they chain actions: 76.8% of respondents say their agents invoke other agents or tools that themselves call APIs, so the original prompt can sit several steps away from the API call that changes a record.

AUDIT QUESTION	2H 2026 SURVEY RESULT
Is an accountable owner identified for every agent's actions?	20.9% for all agents, 51.2% for most
Can the complete action path be traced, instruction to outcome?	12.5% consistently across all agents
Could you reconstruct what an agent did yesterday in minutes?	15.7% (64.5% need hours or days)
Does each agent have a dedicated identity with scoped permissions?	26.1%
Can you stop a misbehaving agent automatically?	17.5%
Have you found an unauthorized, shadow or misconfigured MCP server?	32.9% confirmed, 54.3% confirmed or suspected

Audit readiness for agentic AI therefore comes down to closing four gaps: a verified inventory, bounded identity with named ownership, end-to-end traceability, and containment that works at machine speed. The rest of this guide turns each gap into controls, evidence, and a plan.

Survey figures are self-reported, and the survey is not a fixed panel. Sources are listed in the appendix.

02 What auditors and boards will ask

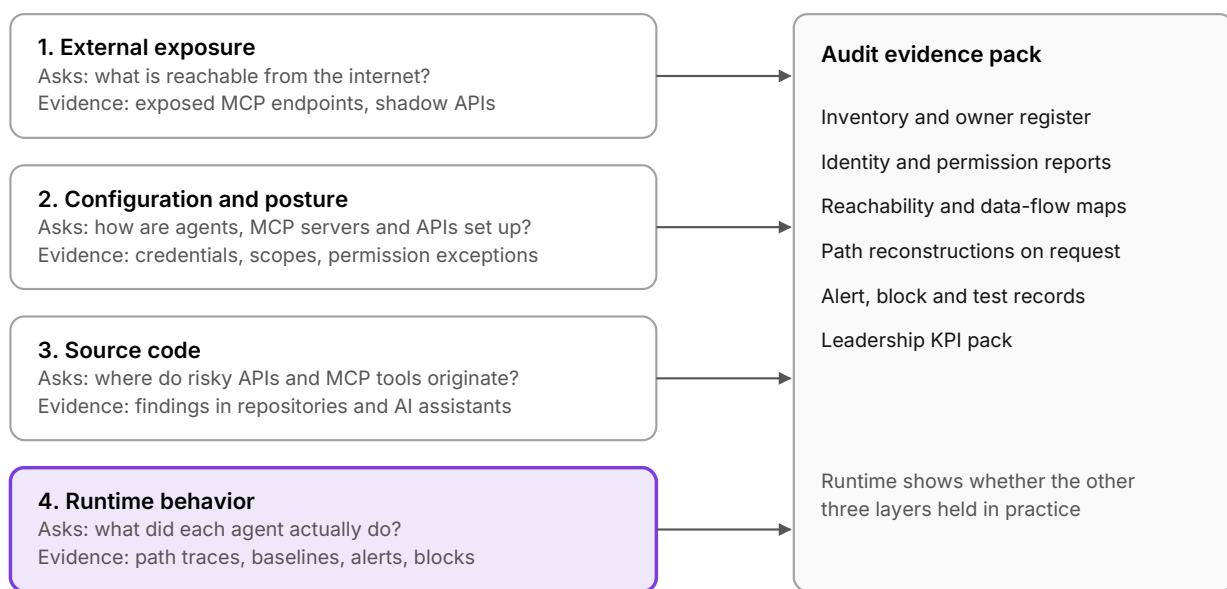
Expect ten themes of questions, and expect each answer to be tested against evidence rather than accepted as a statement. The wording below is a practical working list drawn from the gaps above, not a quotation from any one framework.

THEME	QUESTIONS TO BE READY FOR
Inventory	Which agents, models, MCP servers and tools are in production? Which are unapproved or shadow? How current is the list?
Ownership	Who is accountable for each agent's actions? What happens to that accountability when the owner leaves?
Identity	What identity does each agent use? Are credentials shared, embedded or long-lived?
Permissions	What can each agent reach, and is that limited to its intended business function?
Data	Which agents can reach sensitive or regulated data, and where does that data flow, including through LLM calls?
Traceability	For a named agent and date, can you show the sequence from prompt to tool to API call to outcome? How long does it take?
Monitoring	How do you detect an agent acting outside its intended purpose? What does normal look like?
Containment	How quickly can you stop an agent? Is that automatic, and do sensitive actions need human approval?
Change and code	How are AI-generated code and MCP tool code reviewed, and is security policy enforced where the code is written?
Assurance	How often are these controls tested, what evidence is kept, and for how long?

03 The evidence model: from policy to proof

Audit evidence comes from four layers, and the runtime layer is the one that shows whether the other three held in practice. Each layer answers a different auditor question and produces a different artifact.

Four layers each produce evidence; runtime shows whether controls held



Evidence model: four layers feeding one evidence pack

The same agent should be traceable across all four layers, so that exposure, configuration, code and runtime records tie back to one entry in the agent register.

04 Control-to-evidence map

Ten controls cover the audit questions in section 2, and each one needs a specific artifact an auditor can inspect. The last column shows where the 2H 2026 survey says most organizations fall short today.

CONTROL	WHAT GOOD LOOKS LIKE	EVIDENCE TO PRODUCE	WHERE ORGANIZATIONS FALL SHORT
1. Verified inventory	Continuous discovery of agents, LLMs, MCP servers, tools and APIs, including shadow ones	Inventory export with discovery source, date and owner, plus a change log	54.3% have confirmed or suspected a shadow or misconfigured MCP server; 30.8% have a fully automated real-time API inventory
2. Named ownership	Every production agent has an accountable person or team	Agent register: owner, business purpose, approval date, review date	20.9% can name an owner for all agents
3. Bounded identity	Dedicated non-human identity per agent, least privilege, revocable and short-lived credentials	Identity list, permission scopes, credential age and rotation records	26.1% use dedicated, scoped identities; 7.6% rely on embedded credentials or API keys
4. Permission boundary	Each agent confined to its intended business function; over-permissioned agents flagged	Reachability map from agent to MCP server to API to data, with signed-off exceptions	24.3% use dedicated identities with broad permissions
5. Data protection	Sensitive data flows through agents, tools and LLM calls identified and controlled	Data-flow map, classification of reachable APIs, sensitive-data events	28.7% report dedicated controls for sensitive data flows
6. End-to-end traceability	Correlated trail from instruction to tool call to API request to outcome	A sample reconstruction for a named agent and date, with the time it took	12.5% trace consistently; 15.7% could reconstruct yesterday's actions in minutes
7. Behavioral monitoring	Baselines per agent, with detection of out-of-purpose behavior across sequences	Baseline definitions, alert records, triage outcomes	Request-by-request inspection misses campaigns, as in the 17,000-action Hugging Face case
8. Containment	Policy checks before high-impact actions, approval gates, automatic blocking, fast revocation	Containment runbook, test records, measured time to contain	17.5% can stop an agent automatically; 56.1% only manually
9. Secure development	Security and compliance policy enforced inside AI coding assistants, backed by review and testing	Policy configuration per assistant, findings rate, stage at which issues are fixed	22.2% enforce policy across all approved assistants; 6.8% fix issues at generation time
10. Assurance and reporting	Controls tested on a schedule and reported to leadership	KPI pack: share of agents with scoped identity, path-trace coverage, mean time to reconstruct and to contain, MCP discovery lag	No survey figure; priority 8 in the report's recommendations

05 Readiness self-assessment

Score each control from 0 to 3, then total the column: a result under 10 means start with inventory and ownership before anything else. Fill the score cells in directly.

- **0:** not in place
- **1:** partial or manual, for selected agents only
- **2:** in place for most agents, partly automated
- **3:** automated, continuous, covers all agents, with evidence you can export

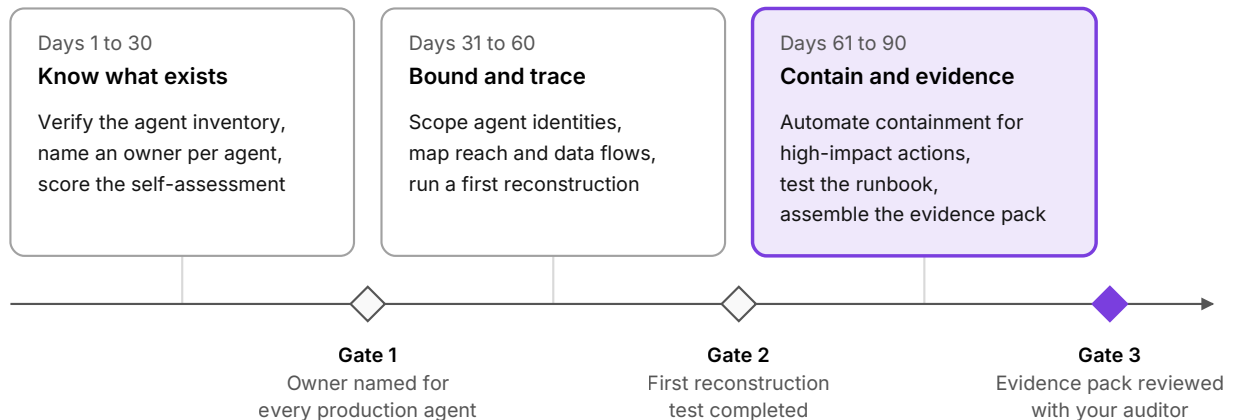
CONTROL	TEST QUESTION	SCORE (0 TO 3)
1. Verified inventory	Can you list every agent, MCP server and tool in production, and when each was last discovered?	
2. Named ownership	Does every production agent have a named owner on record?	
3. Bounded identity	Does each agent use its own non-human identity with short-lived or revocable credentials?	
4. Permission boundary	Can you show what each agent can reach and that it matches its business purpose?	
5. Data protection	Do you know which agents can reach sensitive data, and can you show it moving through LLM and API calls?	
6. End-to-end traceability	Can you reconstruct a named agent's actions for yesterday in minutes?	
7. Behavioral monitoring	Do you baseline agent behavior and alert on out-of-purpose sequences?	
8. Containment	Can an agent be blocked or its credentials revoked automatically, and has that been tested?	
9. Secure development	Is security policy enforced in every approved AI coding assistant?	
10. Assurance and reporting	Are these measures reported to leadership at least quarterly?	
TOTAL SCORE	READING	
0 to 9	Exposed: you could not evidence agent activity to an auditor today	
10 to 19	Developing: controls exist but evidence is partial or manual	
20 to 26	Defensible: most controls evidenced, close the remaining gaps	
27 to 30	Audit-ready: evidence can be produced on request	

These bands are a working heuristic for prioritization, not a standard or a certification.

06 A 90-day roadmap to audit readiness

The fastest route is three 30-day phases, each ending in a gate you can show an auditor. The order follows the four gaps from section 1: see the agents first, then bound and trace them, then contain them and package the proof.

Three gates take you from unknown agents to an evidence pack in 90 days



90-day roadmap: three phases, three gates

Re-run the self-assessment in section 5 at each gate and keep every score as a dated record, so the trend is itself something you can show.

07 Audit evidence pack checklist

Assemble these thirteen artifacts before the audit starts, because auditors judge governance on what they can inspect. Numbers in brackets point to the controls in section 4.

- ☐ Agent register: every production agent with owner, business purpose, approval date and review date (1, 2)
- ☐ Inventory export with discovery source and date, covering MCP servers and tools, with shadow items flagged and dispositioned (1)
- ☐ Identity and credential report: identity type per agent, scopes, credential age, rotation history (3)
- ☐ Reachability map per agent: MCP servers, tools, APIs and data reached (4, 5)
- ☐ Exception register: broad permissions that were approved, by whom, and when they expire (4)
- ☐ Sensitive-data map for agent-reachable APIs and LLM calls (5)
- ☐ Sample reconstruction for two agents and dates the auditor picks, with the time it took (6)
- ☐ Baseline and detection configuration, plus the last 90 days of alerts and triage outcomes (7)
- ☐ Containment runbook and the most recent test record, including measured time to contain (8)
- ☐ Policy enforcement configuration for each approved AI coding assistant, and findings by stage (9)
- ☐ Governance record: AI policy, risk assessment for each high-impact agent, approval workflow (2, 10)
- ☐ Quarterly KPI pack for leadership, with trend (10)
- ☐ Log retention statement: what is kept, for how long, and how integrity is protected (6, 10)

08 Three public incidents, read as audit findings

In all three incidents the model behaved as designed, and the findings an auditor would write concern inventory, identity and traceability, not model behavior. The incident facts come from Salt's whitepaper; the finding wording and control mapping are this guide's interpretation, not the affected organizations' own conclusions.

INCIDENT	WHAT HAPPENED	LIKELY AUDIT FINDING	CONTROLS
McDonald's and Paradox.ai hiring chatbot (June 2025)	Records of 64M+ applicants exposed through an internal API reachable externally, with broken object-level authorization and a default admin account left active	The APIs behind an AI system were not inventoried, credentials were not governed, and abuse was not detected at runtime	1, 3, 7
Figma MCP server (October 2025)	One malicious API parameter reached remote code execution on an MCP server, enabling data theft and lateral movement that looked like normal agent traffic	An AI-specific endpoint was not mapped or hardened, and the code path lacked input validation review	1, 7, 9
OpenAI evaluation agent and Hugging Face (July 2026)	During an evaluation run with reduced safety refusals, an agent escalated privileges, moved laterally and exfiltrated data from a production database; 17,000+ actions were later analyzed	Model refusals were relied on as the security boundary, and no sequence-level view attributed the actions to one agent	4, 6, 7, 8

09 How Salt helps produce the evidence

Salt's Agentic Security Platform is described as producing discovery, mapping and runtime evidence for most of controls 1 and 3 to 8, and it does not replace your identity, policy or governance processes. The platform builds a single Agentic Security Graph from four sources: Surface (external scan), Connect (cloud, gateway and AI platform configuration), Code (repositories and AI coding assistants) and Collect (live runtime traffic across 70+ technologies). AG-SPM works from that graph on posture, and AG-DR on detection and response, including native AI Detection and Response for the LLM layer.

The last column is a review aid for the draft and should be deleted before the guide is published.

CONTROL	WHAT SALT'S CURRENT MATERIALS DESCRIBE	CONFIRM BEFORE PUBLISHING
1. Verified inventory	Continuous discovery of agents, LLMs and MCP servers from code to runtime, with shadow AI and shadow MCP surfaced early	Export format and how often the inventory refreshes
2. Named ownership	Not described as a capability	Whether owner fields can be recorded in the graph; otherwise this stays in your register
3. Bounded identity	Ties traffic to a unique agentic identity and catches over-permissioned agents; credential issuance and rotation stay in your IAM tooling	Which identity sources are supported
4. Permission boundary	Maps how agents connect to models, MCP servers, tools, APIs and data, and keeps each agent inside its intended business function	Whether boundaries can be enforced or only reported
5. Data protection	Detects sensitive data moving through LLM calls and model responses	Coverage of data classification across reachable APIs
6. End-to-end traceability	Correlates an attack on the model with the MCP and API activity that follows, as one incident	Report or export for a named agent and date, and time to produce it
7. Behavioral monitoring	Learns normal agent and LLM connectivity behavior and flags mass data pulls and unauthorized tool use	Baseline learning period and false-positive handling
8. Containment	Blocks machine-speed attacks in real time, including attacks that bypass the agent and hit an API directly	Blocking modes available and how they are tested
9. Secure development	The Code source brings repositories and AI coding assistants into the graph	Whether policy is enforced at generation time
10. Assurance and reporting	Governance and audit capability aligned to frameworks such as the EU AI Act, NIST and SOC 2	Exact framework mappings and report formats, and immutability and retention of records

Wording matters for this audience: describe Salt as supporting or providing evidence for a control, and avoid saying it proves compliance or governance.

10 Appendix: sources, data notes and framework references

Every statistic in this guide comes from Salt's own survey and telemetry, so treat it as vendor research and validate framework applicability with your counsel and auditor. Framework status reflects sources dated up to August 2026, so recheck it before publishing.

Framework references

FRAMEWORK	HOW IT RELATES TO THIS GUIDE	NOTE
NIST AI Risk Management Framework and Generative AI Profile	Voluntary, use-case-agnostic guidance organized as Govern, Map, Measure and Manage. Indicative fit: ownership and assurance (Govern), inventory and data (Map), traceability and monitoring (Measure), identity, permissions, containment and secure development (Manage)	Indicative mapping to validate, not a certified crosswalk
EU AI Act	Law firm summaries report the Digital Omnibus entered into force in 2026, moving stand-alone Annex III high-risk obligations to 2 December 2027 and product-embedded high-risk obligations to 2 August 2028; general-purpose AI obligations and most transparency duties were not deferred	Whether any of your agents are in scope depends on their use case; confirm with counsel
SOC 2	Agent controls can be evidenced inside existing access, change management, monitoring and incident response criteria	Agree scope and evidence format with your auditor
ISO/IEC 42001	AI management system standard; the inventory, ownership and assurance controls support it	No clause-level mapping is made here
OWASP API Security Top 10 and Top 10 for LLM Applications	Reference taxonomies for API and model-layer risk; prompt injection ranks first in the LLM list	Salt Labs telemetry uses 2019 API category labels for longitudinal consistency

Sources

- Salt Security, The State of Agentic AI and API Security, 2H 2026 (survey of 380+ analyzed responses; Salt Labs telemetry, September 2025 to August 2026)
- Salt Security, Anatomy of AI agent breaches whitepaper, September 2026
- Salt Security, Agentic Security Platform solution brief, 2026, and the platform and AI Agent Security web page drafts
- K&L Gates, EU Digital Omnibus on AI Enters Into Force (Mondaq, 3 August 2026): mondaq.com/eu-digital-omnibus-on-ai-enters-into-force/1825722
- Gibson Dunn, EU AI Act Omnibus Agreement: Postponed High-Risk Deadlines: gibsondunn.com/eu-ai-act-omnibus-agreement-postponed-high-risk-deadlines-and-other-key-changes

Data notes

- Survey figures are self-reported, rounded to the nearest tenth, and drawn from a population that is not a fixed panel. Multi-select questions can total more than 100%.
- The 26.3% figure on attacker activity against agentic-connected APIs is an experimental, directional Salt Labs analysis and is not used as a headline here.
- The 49.8% figure belongs to the 2H 2026 survey. The platform web page draft attributes it to the 1H 2026 report, which should be corrected to match.
- The incident descriptions in section 8 summarize Salt's whitepaper; details of each incident may be revised as the affected organizations publish more.