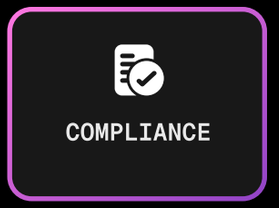
SNYK CHEATSHEET

The 3 Governance Gaps That Put Agentic Applications at Risk

Most organizations already have agents running inside applications. The question is whether you know what those agents are doing — and whether you can stop them when they do the wrong thing. Once agents are live, governance breaks down across three dimensions.



1 VISIBILITY - Discovering where AI is built and runs

You can't govern agents you can't see

Security teams need a live picture of where agents are running, which models they use, which tools they invoke, and which data they access. Most don’t have it.

Shadow AI compounds the gap. Teams build agents and connect them to internal data without security review, making it impossible to enforce policy against what hasn’t been found.

- Unknown agents operate without assigned ownership
- Unapproved models carry unknown risk
- Unvalidated tools, MCP servers, and integrations accumulate across environments

EXAMPLE: A team deploys a support agent. Months later, a second team adds a data enrichment integration downstream. The agent now has access to production customer records that it was never approved to touch.

BUSINESS IMPACT:

Without visibility into deployed agents, organizations can’t determine who owns them, what they can access, or whether they meet security requirements.

snyk

2 RISK INTELLIGENCE - Assessing agent actions

Rules must run where agents act

Many organizations have AI governance policies. But most of those policies live in documents that agents never encounter. That's paper governance, and it creates a dangerous false sense of control.

Agentic applications need enforcement at the execution layer: rules that evaluate what an agent is trying to do, what data it's touching, and which tools it's invoking before the action completes.

- Prompt injection can manipulate agent behavior through user inputs or retrieved content
- Data leakage goes undetected when no policy evaluates the action before a response is generated
- Tool, data, and API chains can create risky flows even when each component appears safe on its own

EXAMPLE: A public-facing assistant is manipulated into summarizing internal support tickets containing PII and including them in its response. No enforcement layer evaluates the action before the data leaves.

BUSINESS IMPACT:

One unblocked action can expose sensitive data, trigger unsafe tool use, or create a risky chain of events before security has a chance to intervene.

3 COMPLIANCE - Becoming auditable

Prove what happened, why, and with what authority

When an agent causes a business problem, basic logs are not enough. Security teams need a defensible record of what the agent did, what data it accessed, which policy applied, and who owned the workflow.

Frameworks like NIST AI RMF, the EU AI Act, and ISO 42001 increasingly require organizations to show that controls were active and not just documented.

- No agent-level audit trail to reconstruct multi-step decisions
- No continuous evidence, only point-in-time snapshots
- No defensible record for regulators, auditors, or legal teams

EXAMPLE: A workflow agent confirms an unauthorized transaction after following a compromised instruction chain. Logs show a request and a response, but security can't reconstruct which policy applied, what data was accessed, or whether the action should have been blocked.

BUSINESS IMPACT:

Without continuous evidence, teams can't prove which controls were active, whether the action was authorized, or why the agent was allowed to proceed.

Visibility without enforcement isn't governance. To govern agentic applications, organizations need a live inventory of what's running, policy enforcement where agents act, and continuous evidence of what happened.

Read the Executive Guide to Operationalizing and Enforcing AI Governance to learn how to move from paper governance to continuous visibility, control, and proof.

snyk