OBSIDIAN

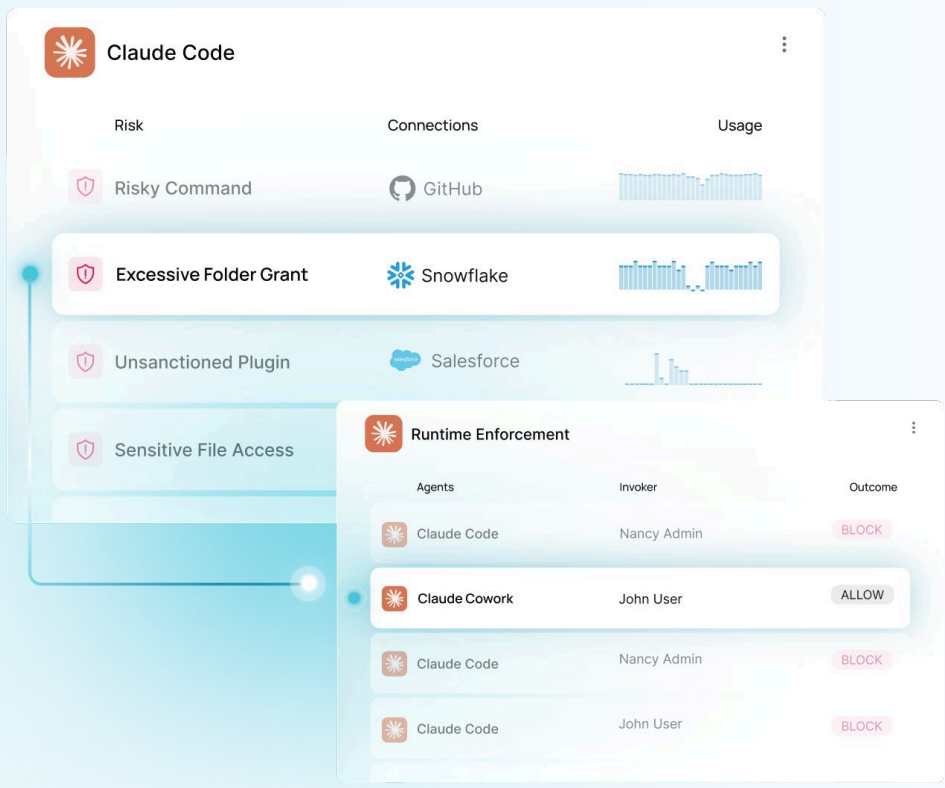
SOLUTION BRIEF

Govern what Claude Code and Cowork agents can access and do

Set policy for the apps, data, tools, and MCP servers Claude Code and Cowork can use, and the actions they can take.

Claude Code is allowed to read SSH keys and overwrite code in GitHub, while Cowork queries tables in Snowflake and sends the data over Slack. And none of it is governed today.

Obsidian Security alerts you when agents hold risky access or take dangerous actions, and enforces policy so agents and their activity remain secure.



Outcomes

Know what every agent can reach:

See the owner, models, skills, permissions, apps, files, tools, and MCP servers behind each Claude agent

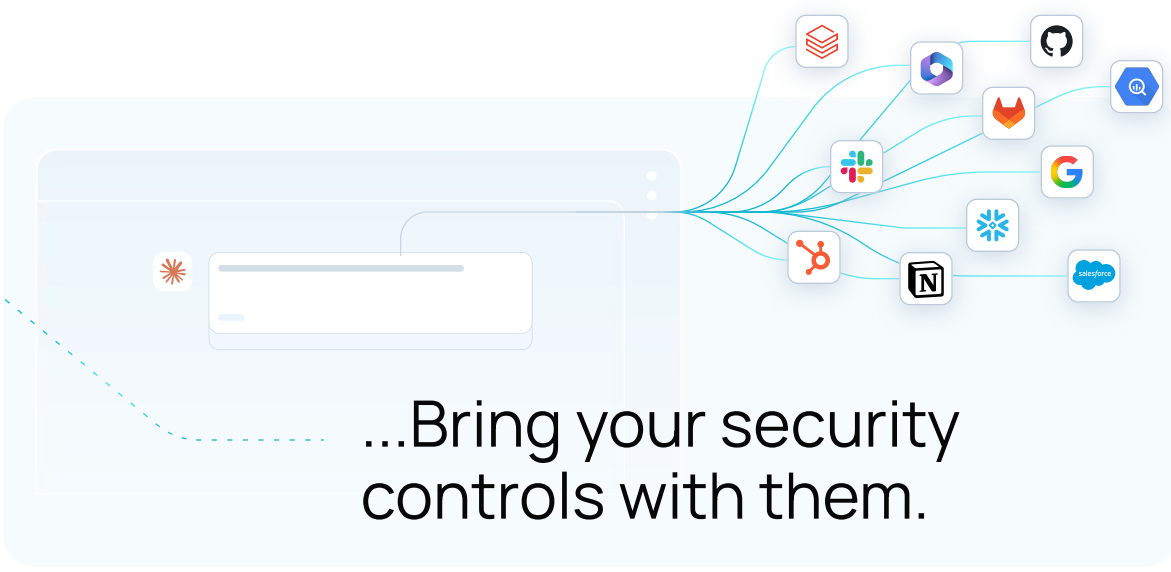
Remove access the agent doesn't need:

Find excessive permissions and prevent sensitive files and credentials from entering agent context

Stop actions that violate policy:

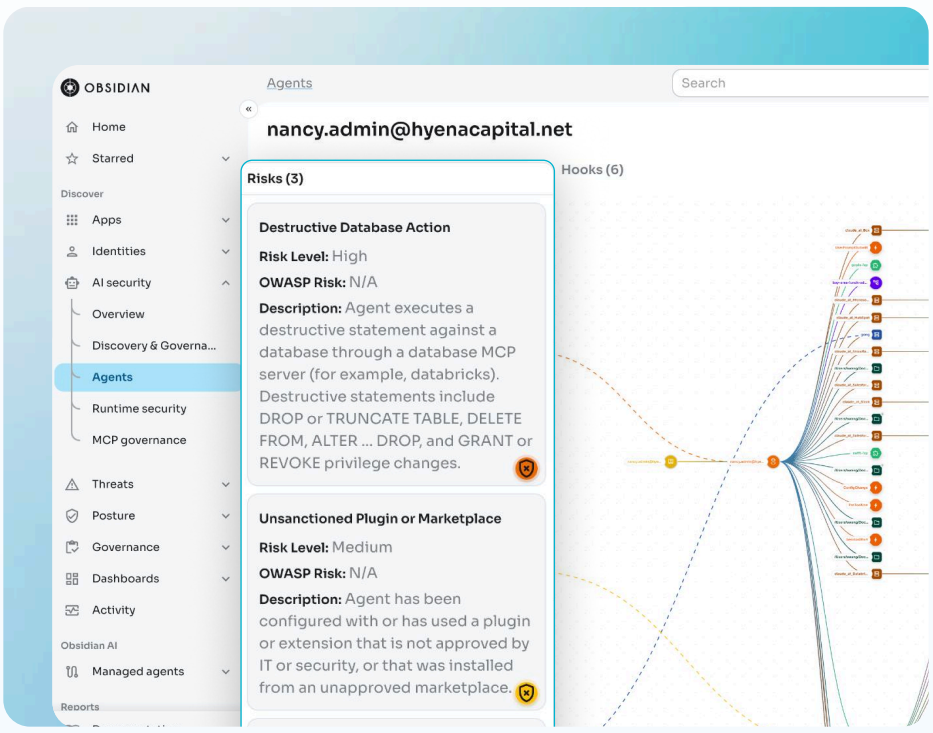
Block unapproved tool calls and prohibit destructive actions before they run

Claude agents are already connecting to business applications...



...Bring your security controls with them.

Key capabilities



Remove risky privilege and access

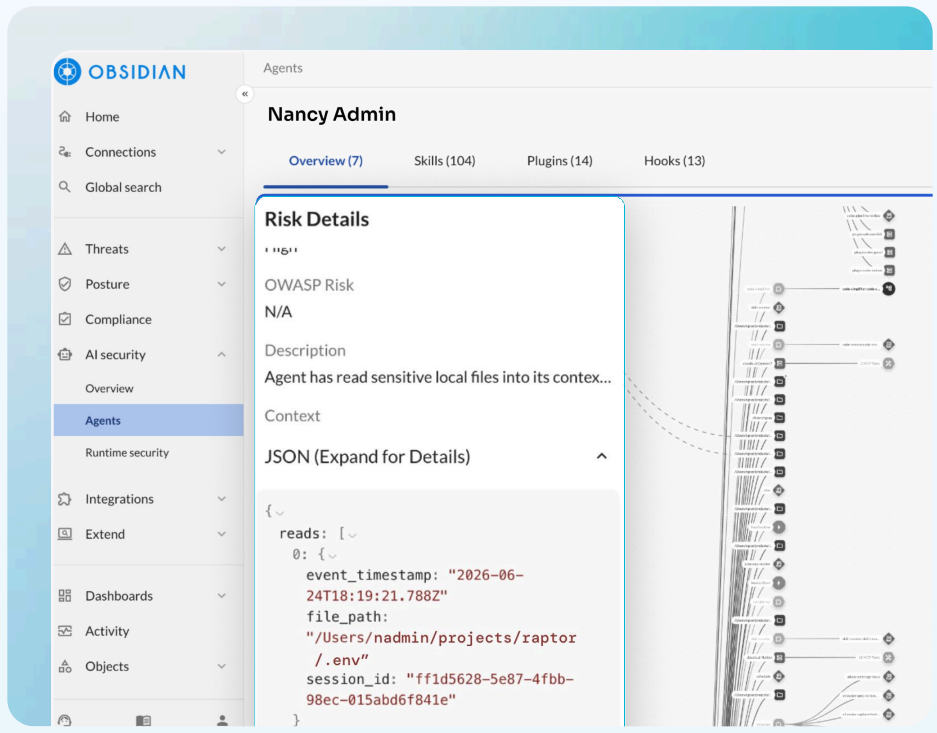
Find and fix agents with more access than their work requires.

- **Cut over-privilege:** See which agents hold dangerous or unnecessary permissions, and what they've already done with them
- **Shrink the blast radius:** Find broad access across files, apps, and systems
- **Prioritize remediation:** Rank agents by the risk of their permissions and evidence of destructive activity

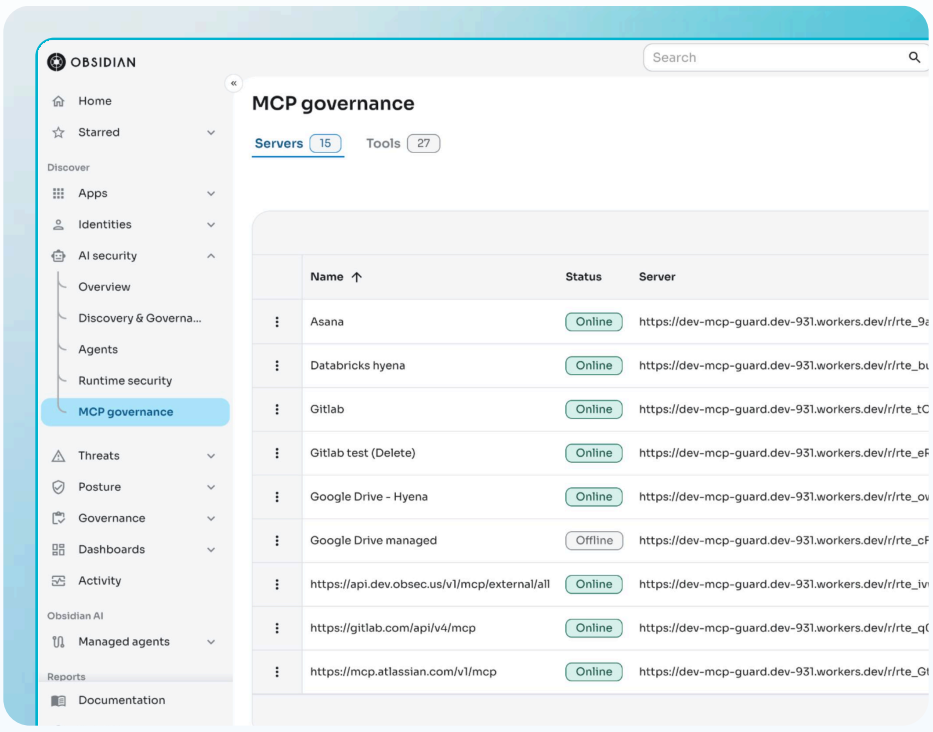
Manage sensitive data access

Know which sensitive data each Claude agent can reach.

- **Map agent access:** See which third-party apps and local files each agent can read, change, move, or share
- **Protect sensitive files:** Detect agents that can read cloud credentials, private keys, and other sensitive local files
- **Find secrets in skills:** Identify skills containing embedded credentials or other sensitive information



Key capabilities continued



Control MCP and tool usage

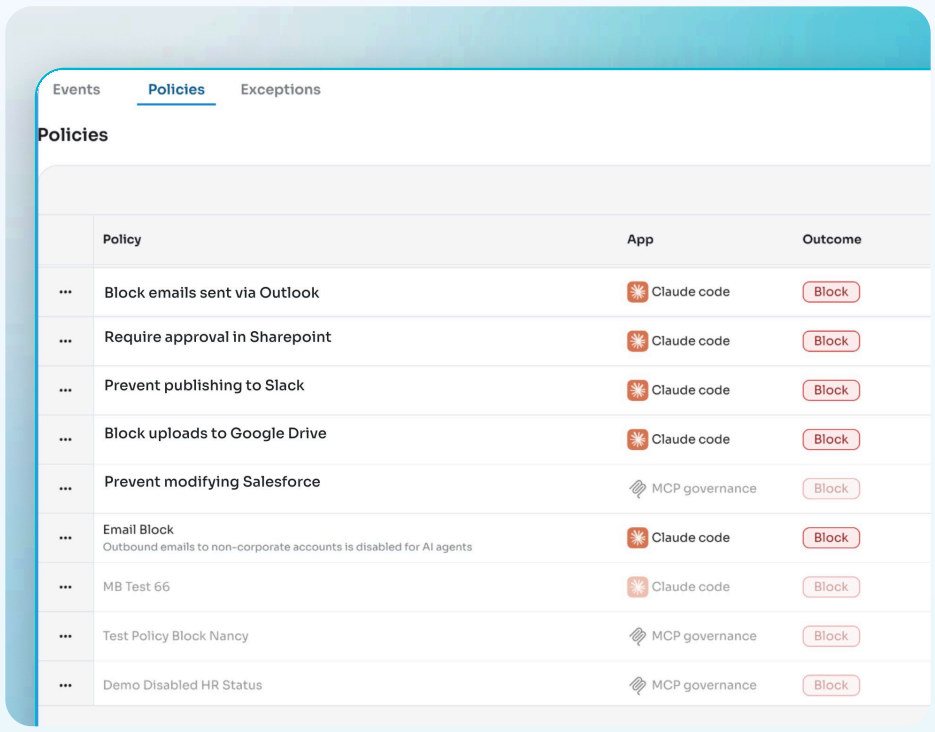
Keep Claude Code and Cowork inside approved tools and MCP servers.

- **Inventory what agents use:** See the models, skills, hooks, plugins, tools, and MCP servers connected to each agent
- **Find unsanctioned connections:** Detect personal, local, custom, or unapproved third-party systems
- **Block unapproved systems:** Prevent tool calls to MCP servers and applications that aren't approved

Prevent destructive actions

Evaluate each agent action against policy before it runs.

- **Choose the enforcement:** Log, warn, block, or require human approval based on the action
- **Apply precise exceptions:** Define which users, agents, and actions a policy applies to
- **Record every decision:** See the attempted action, the policy it triggered, and what Obsidian did in response

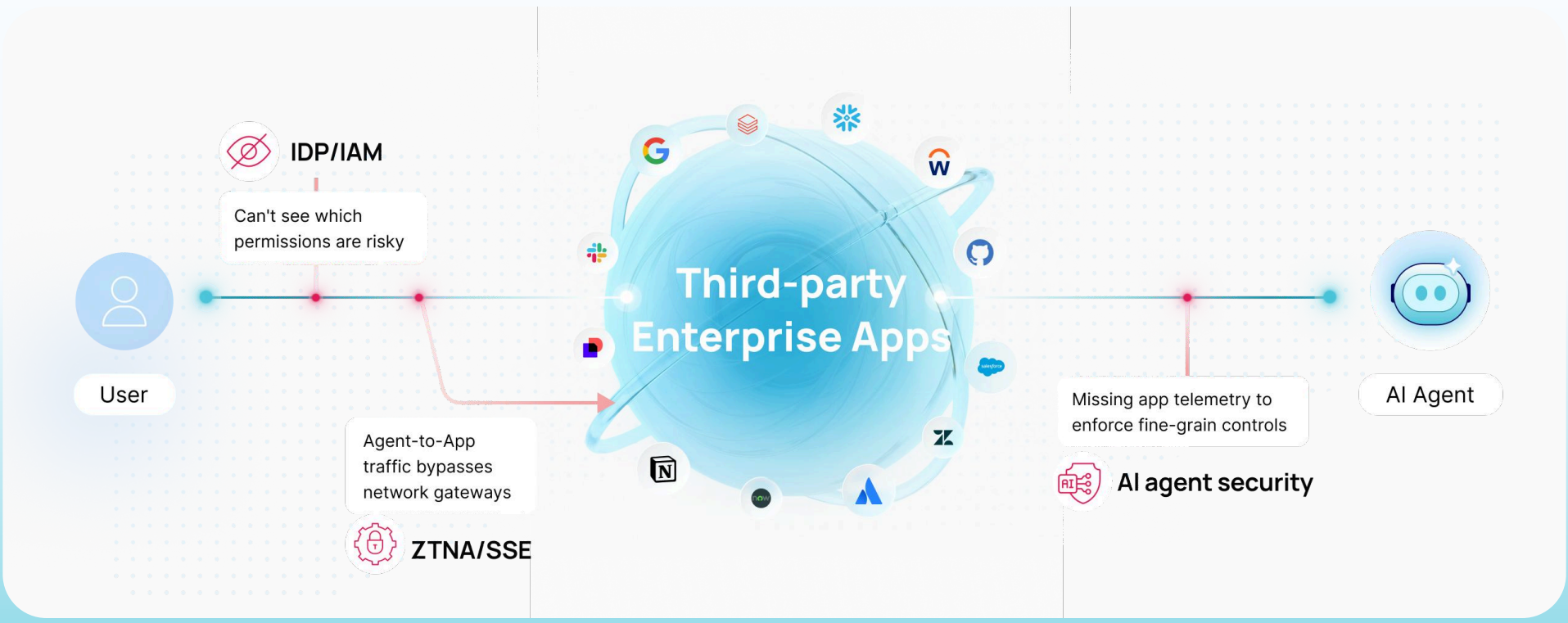


Govern what Claude Code and Cowork agents can access and do

Why Obsidian

Claude Code and Cowork agents act through MCP connections and direct integrations, bypassing every layer of your security architecture.

Obsidian sits where the activity actually happens, so risky actions are detected and stopped before they run.



Network gateways don't see agent actions

Claude can act through OAuth, APIs, and MCP servers without sending the tool call through a network gateway. Obsidian correlates the agent action with what changed in the connected app.

Identity providers don't evaluate agent privilege

SSO and MFA confirm who signed in. They don't tell you that the account behind an agent can drop a production table. Obsidian maps the permissions the agent inherits and shows which ones create risk.

Native controls don't assess the risk behind access

Anthropic controls which tools, commands, and MCP servers Claude can use. Obsidian shows which permissions expose sensitive data, which agents have taken destructive actions, and which actions violate policy.

Start securing your Claude agents today

See every Claude Code and Cowork agent, reduce risky access, and stop prohibited actions before they run.

Get a demo ↗