harmonic | Guide

Claude Security Guide

Protecting Sensitive Data Across Every Claude Product

Claude

Claude has expanded from a single chat interface into five distinct products, each with different access levels, data flows, and risk profiles. The security controls available — and the risks you need to manage — differ meaningfully across them. This guide gives you 3–5 high-impact settings or practices for each product, organized around the most common enterprise concerns: preventing sensitive data from reaching the model, avoiding malicious Skills/MCPs, and general privacy hygiene.

Executive Summary:
5 Actions to Take Tomorrow

The 25 controls in this guide consolidate into five cross-cutting priorities. Each applies across multiple Claude products and can be acted on immediately — no policy cycle or vendor engagement required.

#	Action	What to do — and where
1	**Configure data privacy and retention** All products	**Consumer plans (Chat, Cowork):** Settings → Privacy → Model Training → toggle off. Without this, conversations may be retained up to 5 years and used for model training. **Enterprise (Chat):** Admin Console → Organization Settings → set a custom data retention period matched to your compliance obligations. Shorter is better for regulated industries. **Claude Code / API (Enterprise):** Request and activate a Zero Data Retention (ZDR) addendum — this ensures no prompts, outputs, or metadata are ever stored on Anthropic's servers.
2	**Lock down your MCP and connector surface** Chat • Cowork • Code • Agent Teams	**Chat and Cowork:** Organization Settings → Connectors → remove any connector not explicitly approved by IT. For regulated teams, also disable the public extensions directory (Organization Settings → Extensions). **Claude Code and Agent Teams:** Deploy a managed-mcp.json to the system-wide config directory (/Library/Application Support/ClaudeCode/ on macOS). This gives IT exclusive control — users cannot add unapproved servers. Pair with a denylist for belt-and-suspenders coverage. **All products:** Treat every MCP server like a software dependency. Check the publisher, review source code where possible, and pin to a known version before deploying org-wide.
3	**Enforce least-privilege access everywhere** All products	**Chat (Enterprise):** Enable SSO (SAML 2.0 / OIDC) and domain capture so every org user is enrolled in the managed workspace — closing the shadow-account loophole that bypasses all enterprise controls. **Cowork:** Create a dedicated AI working folder and grant access only to that directory. Never approve root, home, or any path containing .env files or credential stores. **Claude Code:** Add deny rules in settings.json for .env, ~/.ssh/, secrets/, and credentials/. Never run as root — Claude Code inherits your shell's permissions, so a compromised session would have admin access.API: Create a separate API key per application or integration. Set per-key spend limits. Never share a single root key across multiple systems.

4	**Add human checkpoints for irreversible actions** Cowork • Code • Agent Teams • API	**Cowork:** Adopt a team policy that all Cowork-produced content — emails, reports, documents — is reviewed before external distribution. Cowork runs asynchronously and makes decisions without check-ins; treat its outputs as drafts. **Agent Teams:** Enable plan-approval mode before spawning any agent team. Each teammate must submit a plan for the lead to review before any file is written, committed, or deployed to production. **Claude Code:** Keep file-write and command-execute permissions on "Ask" for anything touching production systems, git, or network endpoints. Only pre-approve read-only operations on known-safe paths. **API:** Use a restrictive system prompt to define Claude's role and hard limits. Pre-screen user inputs with a lightweight classifier (e.g., Haiku) to catch prompt injection and PII before they reach the main model.
5	**Protect credentials and build an audit trail** Code • Agent Teams • API	**All products:** Never paste API keys, passwords, or connection strings into any Claude interface. They are transmitted to Anthropic's servers and may appear in logs. Inject secrets at the infrastructure layer only. **Claude Code and Agent Teams:** Enable OpenTelemetry audit logging and route to your SIEM (e.g., CloudWatch, Splunk). Set alerts on credential file access patterns (.env, ~/.ssh). After any large autonomous Agent Teams run, rotate all credentials the session could access. **API:** Monitor per-key token consumption via the Console Usage API. Anomalous spikes — especially off-hours or from unexpected IPs — are your primary early-warning signal for key compromise.

⚠ **Critical foundation: know which plan tier you're on**

Consumer plans (Free, Pro, Max) now default to training on your data. Enterprise, Team, and API plans operate under Commercial Terms and never use data for training. For sensitive business use, do not use consumer plans — the contractual protections are fundamentally different.

1 - Claude Chat (claude.ai)

Risk profile: Conversational data, uploaded files, and Connector-sourced content all pass through Anthropic's servers. The primary risks are training data exposure, shadow use of personal accounts, and unvetted third-party Connectors.

Claude Chat — Security Settings

claude.ai web, mobile, and desktop app

#	Setting / Control	Risk Category
1	**Use Enterprise or Team plan — not Pro/Max for business data** Consumer plans (Free, Pro, Max) now default to training on data. Enterprise & Team plans operate under Commercial Terms with no model training, ever.	Data Residency
2	**Disable model training in Privacy Settings (if on consumer plan)** Go to Settings → Privacy → Model Training → Toggle off. Without this, new/resumed chats may be retained 5 years and used for training.	Data Governance
3	**Enable SSO and enforce domain capture (Enterprise)** Configure SAML 2.0/OIDC SSO so all org users authenticate through your identity provider. Domain capture auto-enrolls users in the org workspace, preventing shadow personal accounts.	Identity
4	**Set a custom data retention period and minimum** Enterprise admins can configure org-wide retention periods (min 30 days). Set this to match your compliance obligations — shorter is better for regulated industries.	Compliance
5	**Restrict which Connectors/MCP servers are enabled org-wide** On Team/Enterprise, only admins can add Connectors. Manage the allowlist under Organization Settings → Connectors. Disable all connectors not explicitly approved.	MCP Security

2 - Claude Cowork

Risk profile: Cowork has local file access and browser access to logged-in sessions — it can read emails, internal dashboards, and any file in approved folders. The async, long-running nature means it may take broad actions before a human reviews them.

Claude Cowork — Security Settings

macOS desktop app, research preview

#	Setting / Control	Risk Category
1	**Grant folder access only to specific, non-sensitive directories** Cowork requests folder-level permissions. Never grant access to root, home, or directories containing credentials (.env, .ssh, secrets/). Create a dedicated working folder for AI tasks.	Least Privilege
2	**Close or hide sensitive windows before starting computer use tasks** When Cowork uses your browser, it can see your logged-in sessions. Close tabs with banking, HR systems, internal dashboards, or any privileged access before initiating tasks.	Data Exposure
3	**Restrict Chrome connector to specific tabs or profiles** Use a dedicated Chrome profile for Cowork with only the accounts/tabs needed for the task. Avoid giving it access to your primary profile with all accounts logged in.	Browser Hygiene
4	**Review Skills from the allowlist only; disable public extension directory if regulated** Enterprise admins can upload custom-vetted Skills and restrict which public extensions users can install. Under Organization Settings → Extensions, enable the allowlist and disable the public directory.	Skills / MCP
5	**Treat Cowork outputs as drafts — require human review before publishing or sending** Cowork operates asynchronously and may make decisions based on ambiguous instructions. Establish a policy that Cowork-produced content (emails, reports, code) is reviewed before external distribution.	Human Oversight

3 - Claude Code

Risk profile: Full terminal access means Claude Code runs with your shell permissions — it can read credentials, .env files, SSH keys, and execute arbitrary commands. MCP servers can connect to production systems. Prompt injection via code comments or READMEs is a real, documented attack vector.

Claude Code — Security Settings

Terminal / CLI, VS Code, Web, iOS

#	Setting / Control	Risk Category
1	**Deploy managed-mcp.json to lock down MCP servers org-wide** Place a managed-mcp.json in the system-wide config directory (/Library/Application Support/ClaudeCode/ on macOS). This gives IT exclusive control — users cannot add unapproved MCP servers. Use denylist as well for belt-and-suspenders.	MCP Security
2	**Add deny rules for sensitive file patterns and dangerous commands** In settings.json, add deny rules for .env, ~/.ssh/, secrets/, credentials/, and network commands like curl/wget. These prevent Claude Code from reading secrets or exfiltrating data even if a prompt injection attempts it.	Data Exposure
3	**Never run Claude Code as root or with admin privileges** Claude Code inherits your shell's permissions. If it runs as root, a compromised session or prompt injection has admin access. Run as a standard user with only the file access needed for the current project.	Least Privilege
4	**Use Zero-Data-Retention mode for sensitive codebases (Enterprise add-on)** ZDR mode ensures no prompts, outputs, or metadata are stored on Anthropic's servers. Essential for proprietary IP, financial code, or regulated environments. Requires an Enterprise plan security addendum.	Compliance
5	**Enable OpenTelemetry audit logging to your SIEM** Claude Code exports traces via OpenTelemetry. Route this to your SIEM (e.g., CloudWatch, Splunk) to maintain audit trails of all file reads, commands executed, and MCP tool calls. Set alerts for credential file access patterns.	Auditability

4 - Agent Teams (inside Claude Code)

Risk profile: Multiple agents running in parallel multiply the attack surface. Each agent operates in its own context window with its own permissions. Mistakes (or injections) can propagate across the team before a human catches them. Token costs and credential exposure scale with team size.

Agent Teams — Security Settings

Experimental feature inside Claude Code

#	Setting / Control	Risk Category
1	**Require plan-approval mode before teammates make any changes** In Agent Teams, you can require each teammate to submit a plan before executing. The lead reviews and approves before any file changes. Use this for production code and anything touching sensitive directories.	Human Oversight
2	**Pre-approve only low-risk operations; keep write/exec permissions on 'Ask'** Before spawning teammates, configure permissions so routine reads are auto-approved but file writes, command execution, and git pushes require manual confirmation. Reduces interruptions without sacrificing control.	Least Privilege
3	**Scope each teammate's context to only the files/modules it needs** When spawning agents, specify exactly which files or directories each teammate should work in. A teammate building API endpoints doesn't need access to auth modules or secret management code.	Isolation
4	**Monitor all teammate sessions in split-pane mode during sensitive operations** Use tmux or iTerm2 split-pane view to watch all agent sessions simultaneously. Steer agents away from unexpected behavior early — before they waste tokens or make harmful changes.	Monitoring
5	**Rotate API keys and revoke session credentials after large autonomous runs** Agent Teams can generate a large number of API calls and access tokens across sessions. After major runs, rotate any credentials the session had access to, especially if MCP servers with production access were used.	Credential Hygiene

5 - Claude Console & Developer API

Risk profile: The API is the most powerful and most flexible — and the least opinionated about security. Developers control the system prompt, model selection, data flow, and access patterns. The primary risks are data sent to the model (PII, IP, regulated data), prompt injection from end users, and API key management.

API / Console — Security Settings

console.anthropic.com + API (AWS Bedrock, Google Vertex AI)

#	Setting / Control	Risk Category
1	**Enable Zero Data Retention (ZDR) for regulated or sensitive workloads** ZDR processes every request ephemerally — no prompts, outputs, or metadata persist on Anthropic's servers. Available via Enterprise API addendum. Essential for PHI, PII, financial data, or IP-sensitive applications.	Compliance
2	**Deploy via AWS Bedrock or Google Vertex AI for full network isolation** Both platforms support VPC-isolated Claude endpoints (AWS PrivateLink / GCP Private Service Connect). Traffic never traverses the public internet, meeting network sovereignty requirements for banking, healthcare, and government.	Network Isolation
3	**Use a system prompt to restrict what users can ask Claude to do** Craft a system prompt that explicitly limits Claude's role, forbids it from discussing certain topics, and prevents disclosure of system prompt contents. This is your first line of defense against misuse and data extraction.	Access Control
4	**Pre-screen inputs with a lightweight model (e.g., Haiku) for injection and PII** Route user inputs through a Haiku-powered classifier before the main model. Check for prompt injection patterns, jailbreak attempts, and PII using structured outputs. Block or sanitize before proceeding.	DLP / Injection
5	**Scope API keys by use case; rotate regularly and monitor via Usage API** Create separate API keys for each application or integration. Set spend limits per key. Use the Console Usage API to monitor token consumption by key — anomalous spikes may indicate key compromise or abuse.	Key Hygiene

At a Glance: Security Feature Availability

Key controls and which plan tier enables them across the five products.

	Chat	Cowork	Claude Code	Agent Teams	API / Console
No training on data	Opt-out req.	Opt-out req.	Enterprise only	Enterprise only	**✓ Always**
MCP/connector allowlist	Admin ctrl.	Admin ctrl.	managed-mcp.json	Inherits Code	Dev-defined
Zero Data Retention	—	—	Enterprise add-on	Enterprise add-on	**✓ Available**
SSO / SCIM	Enterprise	Enterprise	Enterprise	Enterprise	Enterprise
Audit logging	Enterprise	Limited	OpenTelemetry	OpenTelemetry	Usage API
Custom retention	Enterprise	Enterprise	N/A (local)	N/A (local)	30-day default

Universal Recommendations (All Products)

🔑 Never paste credentials into ClaudeAPI keys, passwords, connection strings, and tokens sent as chat context are transmitted to Anthropic's servers and may appear in logs. Use placeholder references and inject secrets at the infrastructure layer.

🛡️ Treat MCP servers like software dependenciesVet every MCP server before installation. Check the publisher, review the source code if possible, and pin to known versions. Malicious MCP servers can exfiltrate data through tool call responses.

📋 Establish an acceptable-use policy for AI toolsDefine what data categories (PII, IP, regulated) may not be sent to AI models. Employees using personal Claude accounts bypass enterprise controls entirely — policy must address shadow use.

🔍 Assume prompt injection is possible — design for itAny content Claude reads from the web, files, or emails could contain malicious instructions. For agentic tools (Cowork, Code, Agent Teams), apply deny rules and human-in-the-loop checkpoints for irreversible actions.

Current as of February 2026 · Prepared for webinar reference · Security controls subject to change with product updates · Verify current settings at support.claude.com and platform.claude.com/docs

Why Harmonic?

Harmonic Protect empowers security teams to safeguard sensitive data without the need for extensive data labeling or complex rule-setting.

Your data in your hands
Our models do not need to store or train on client data. Instead, we use our unique sets of public, anonymized data.

Recognized for innovation
Harmonic was named Gartner Cool Vendor and RSA Innovation Sandbox finalist.

Scale the security team
Minimal false positives and end-user coaching minimize security team effort with our “zero-touch” approach.

> We wanted to adopt GenAI tools but were worried about the risks to our sensitive data. Existing controls could block the whole category, but that’s not what we wanted. Harmonic gives me visibility and control. I’m excited to roll our their ‘human-like’ data protection. Current-gen DLP is too much effort for my team and doesn’t find things I care about.
>
> **Sascha Maier,**
> CISO, SV Group

Getting Started with Harmonic

Getting started with Harmonic is quick and easy. Simply install the Harmonic browser extension to start gaining insights in GenAI usage and secure your sensitive data.

Within 30 minutes, the extension may be rolled out to your entire organization with Group Policy Object (GPO), Microsoft Intune, JAMF, or Kandji.

Get Started >

harmonic harmonic.security sales@harmonic.security

Harmonic Security gives security teams the tools to protect sensitive data without the headaches of labeling and complex rules. Our pre-trained data protection models enable secure innovation through user interaction and gentle nudges. Recognized as an RSA Innovation Sandbox finalist in 2024, Harmonic Security redefines data protection for the GenAI era.