MCP SERVER | NO CODE | CLOUD HOSTED

Kandji MCP

Audit every device and security policy in your fleet.

Kandji connects your AI agent directly to Apple's Mobile Device Management system. Audit, manage, and enforce compliance across entire macOS and iOS fleets from a single prompt. This MCP lets you retrieve specific device details, track user assignments, check security blueprints, and audit historical management commands without logging into the Kandji dashboard.

A+ Quality Score 100/100

mdm | apple-device-management | fleet-security | it-automation | macos-management | ios-security

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/kandji

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Kandji MCP

10 tools available
Cloud-hosted on Vinkius

Managing large groups of Apple devices involves dozens of dashboards, reports, and manual checks. With this connector, your AI agent handles that overhead. You can query everything from basic device inventory to deep security compliance status in plain language. Need to know which user owns a specific Mac? Or check if the latest OS patch was applied across 50 units? Your agent pulls those details for you.

It acts as an automated extension of your existing IT workflow, allowing you to gather comprehensive reports on device health and security history instantly. When connected via Vinkius, this MCP becomes a key part of your overall enterprise intelligence layer, letting any compatible AI client execute complex auditing tasks across multiple systems—all without needing to know the underlying Kandji API structure.

Core Capabilities

01 — Audit Device Inventory and Status

Retrieve comprehensive lists of all managed Apple devices, including their OS version and unique IDs.

02 — Examine Security Configuration

List available security parameters (policies) and blueprints to understand how your organization categorizes device compliance.

03 — Track Historical Events

View logs of management activity, recent commands sent to devices (like wipes or restarts), and account changes over time.

04 — Identify Ownership and Software

List all users associated with the fleet, as well as every custom and auto-deployed application running on those machines.

05 — Verify Organizational Scope

Confirm details about your Kandji account identity before executing large-scale audit commands.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/kandji** — connect your AI agent in three steps.

01 Tell your agent the specific data point you need, such as 'list all devices in California' or 'show last week's security changes.'

02 The MCP translates that request into the necessary Kandji API calls and fetches the structured report data.

03 Your agent receives a clean, summarized output, allowing you to immediately read, analyze, and act on the compliance findings.

The bottom line is: your AI client gets actionable device security reports without you ever touching an MDM console.

Built For

The IT Operations Engineer who needs to run a full compliance audit before a major rollout. The Security Analyst who has to prove data retention policies manually. Or the Systems Administrator tired of clicking through multiple dashboards just to answer one question about device status.

IT Operations Engineer

Running routine fleet checks, such as listing all managed devices and checking if required software is deployed across every unit.

Security Analyst

Auditing system changes and tracking administrative activity to ensure policy adherence and detect unauthorized access attempts.

Systems Administrator

Verifying user assignments across the entire device pool or checking which security blueprints are active for new hardware deployments.

What Changes When You Connect

01 You eliminate the need to manually cross-reference spreadsheets. By using list_devices, you get a real-time roster of every Apple asset currently enrolled in the network.

02 Never waste time guessing compliance status again. Use list_parameters and list_blueprints to see exactly what policies are available and how your devices are categorized for enforcement.

03 When something goes wrong, you don't have to guess why. By running list_activity or viewing recent management commands via list_commands, you get a full audit trail of who did what and when.

04 Finding out who owns a device used to take digging through multiple tabs. Now, calling list_users instantly maps every asset back to its primary user account.

05 It saves hours of work by letting your agent aggregate information from different sources—like combining list_devices data with list_custom_apps data—in one query.

Real-World Applications

Pre-Audit for New Policy Rollout

An engineer needs to deploy a new security policy. They ask their agent to first run list_devices and then use get_device on several sample units. The agent compiles the current OS version and compliance status of each, ensuring no device falls outside the acceptable range before deployment.

Investigating Device Loss

A user reports a lost Mac computer. The analyst asks their agent to check list_activity and list_commands for that specific device ID. The agent quickly identifies if the last recorded action was a 'Wipe' command or if there are any unusual system changes in the logs.

Onboarding New Departments

A manager needs to ensure all new employees have correct software and user assignments. They ask their agent to list_users for a specific department, then run list_custom_apps to confirm that the required departmental applications are installed on every assigned device.

Compliance Reporting

The security team needs proof of adherence to regulations. They ask their agent to pull data using list_parameters and get_organization details, generating a report proving that all managed assets meet the minimum required security controls defined by the organization.

Patterns to Avoid

Asking for 'All device data'

✗ AVOID

The agent gets overwhelmed with raw JSON dumping every single metadata point, making it impossible to find the specific OS version or user ID they needed.

✓ INSTEAD

Instead of general queries, use list_devices first to get a clean roster. Then, if you need deep detail on one unit, call get_device using the specific device name or ID.

Ignoring historical context

✗ AVOID

The team sees a compliance failure today but doesn't know when it started. They just check the current dashboard view.

✓ INSTEAD

Always run list_activity to get recent management history. This tool shows exactly which administrative change or system event caused the deviation, giving you critical context.

Assuming a policy exists

✗ AVOID

A user assumes 'Disk Encryption' is an available setting and tries to enforce it without checking if the blueprint supports it.

✓ INSTEAD

First run list_parameters. This tool lists all valid security controls, ensuring you only attempt to configure policies that actually exist in your Kandji environment.

The Right Fit

Use this MCP if your primary need is automating IT auditing and compliance checks against a standardized Apple MDM system (macOS/iOS). This connector excels at listing assets (list_devices), verifying ownership (list_users), and proving policy adherence by checking available controls (list_parameters).

Don't use it if you are trying to manage non-Apple hardware (like Windows desktops) or if your goal is communicating with a separate system, like an HR database. For cross-platform data integration, look for a general enterprise connector type. If your task involves writing code that *uses* the device data but doesn't *read* it from Kandji, you might need to integrate this MCP output into a workflow automation tool instead.

The pain of manual compliance audits

Today, proving that your entire corporate fleet meets security standards is an exercise in clicking. You have to jump between the device roster tab, the user management section, and several different policy dashboards just to piece together a single report. Then you copy-paste those details into a spreadsheet for review.

With this MCP, the process changes entirely. Your agent handles that tedious navigation. You ask it one question—for example, 'Show me all devices missing mandatory encryption'—and it pulls the data from list_devices and compares it against the status found in list_parameters. The result is an immediate, actionable report.

Get instant device inventory with Kandji MCP

Manually tracking which devices are running out-of-date software or haven't been assigned a user requires multiple checks across the dashboard. You have to check list_devices, then open details for each one, and finally cross-reference with list_users.

Now, you just ask your agent: 'List all MacBooks that are running macOS Monterey or older.' The MCP runs the necessary queries and delivers a filtered list right away. It's immediate audit power.

Kandji: Device & Compliance Audits (10 Tools)

These tools let your AI agent perform deep audits on every aspect of your Apple device fleet—from current software versions to historical security commands.

#	TOOL	DESCRIPTION
01	`get_device`	Retrieves deep, specific details and metadata for a single, named Apple asset.
02	`get_organization`	Verifies your account's identity by retrieving core details about the Kandji organization itself.
03	`list_activity`	Gathers a chronological list of recent system changes and management actions taken within the platform.
04	`list_auto_apps`	Lists all standard software libraries that Kandji manages for deployment across your devices.
05	`list_blueprints`	Shows the available templates used to categorize, configure, and enforce standards on different device groups.
06	`list_commands`	Lists recent remote management commands sent out, such as Lock, Wipe, or Restart actions.
07	`list_custom_apps`	Provides a list of proprietary or non-store applications that have been deployed to your fleet.
08	`list_devices`	Returns a complete roster of all managed Apple devices, showing their IDs, names, and OS versions.
09	`list_parameters`	Lists every available security control or policy parameter that can be used to secure the environment.
10	`list_users`	Retrieves a comprehensive list of all users associated with your managed devices, confirming ownership records.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U List all managed Mac computers in Kandji.

I'll fetch the list of all Apple devices currently enrolled in your Kandji account.

U Show me the details for device ID 'abc-123'.

I'll retrieve the full inventory and security metadata for that specific Apple device.

U Check recent administrative activity in Kandji.

I'll look up the log of recent management actions and system events in your Kandji account.

Frequently Asked Questions

01 **How do I find out which user owns a specific device using Kandji MCP?**

You can use the list_users tool to view all associated users and then reference those results against your device inventory. This confirms ownership records quickly.

02 **Does Kandji MCP allow me to see past security changes?**

Yes, run list_activity to get a historical log of recent management actions and system events. It gives you the audit trail for compliance review.

03 **Can I check what apps are installed on my devices with Kandji MCP?**

You can use both list_auto_apps and list_custom_apps to see all standard and proprietary software deployed across your entire fleet.

04 **How do I audit the overall scope of my account in Kandji MCP?**

Use get_organization to verify core identity details about your Kandji setup. This is a good first step before running large-scale audits.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"kandji": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

Kandji is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Kandji. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Kandji MCP
Server ID	`019d75bf-2f03-7202-a8e6-2d6b34f5f0d6`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/kandji.