MCP SERVER | NO CODE | CLOUD HOSTED

Pangea Security APIs MCP

Guard Inputs, Scrub Data, and Audit Everything.

Pangea Security APIs is an essential security layer for building LLM applications. It lets your AI client automatically scan inputs and outputs for sensitive data (PII), detect prompt injections, check IP origins against embargo lists, and manage user access rights before the information ever hits your model.

F Quality Score 3.6/100

ai-guardrails | pii-redaction | threat-intelligence | audit-logging | prompt-injection | data-privacy

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/pangea-security-apis

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects 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.

Pangea (Security APIs) MCP

40 tools available
Cloud-hosted on Vinkius

When you build any application around a large language model, security is the biggest headache. You can't trust what comes in or what goes out. This MCP gives you a unified way to enforce data rules right at the start of your process. It lets your agent automatically scrub plain text and structured JSON objects for private information; it also scans chat completions and prompts before they execute, stopping malicious inputs like prompt injections cold.

If you're tracking activity, you get more than just a log file—you can search through all historical events using natural language queries. You can also validate user identity by starting sign-in flows or checking if an IP address is coming from a restricted region. Because it's hosted on Vinkius, connecting this MCP to your workflow means you don't have to build custom middleware; you just connect and start securing everything.

Core Capabilities

01 — Data Privacy and Redaction

Automatically find and scrub sensitive information from plain text or complex JSON objects.

02 — AI Input Guarding

Analyze prompts and chat completions to detect malicious content, PII, and prompt injection attempts before processing.

03 — Security Auditing and Logging

Maintain a chronological record of all actions taken in your system and search those records using plain language.

04 — Identity and Access Management

Control who can access resources, manage user sessions, and programmatically create or update user accounts.

05 — Threat Intelligence Vetting

Check IP addresses for geopolitical embargoes, detect proxies, and scan files for known malware signatures.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/pangea-security-apis** — connect your AI agent in three steps.

01 First, subscribe to this MCP on the Vinkius Marketplace and input your Pangea Token and Domain credentials.

02 Next, configure your AI client—like Cursor or Claude—to route sensitive inputs and outputs through the security tools provided by this MCP.

03 Finally, when a user interacts with your application, the tools run in real time, checking for threats, redacting data, and logging the event before proceeding.

The bottom line is that it wraps your existing AI logic in layers of mandatory security checks.

Built For

Security Engineers, AI Developers, and Compliance Officers need this. If you spend time building complex agent workflows but worry about data leaks or malicious inputs, this MCP gives you the guardrails you need without writing a single line of custom middleware.

AI Developer

Implementing safety checks and PII redaction for LLM interactions on both inputs and outputs.

Security Engineer

Automating the analysis of audit logs, vetting incoming IP addresses, and managing system access controls.

Compliance Officer

Running natural language queries against historical security records to verify compliance trails quickly.

What Changes When You Connect

01 Stop data leaks before they start. Use `redact_text` or `redact_structured` to strip PII from user inputs, ensuring your model never sees private customer data.

02 Protect against bad actors with AI Guarding. Tools like `ai_guard_prompt` and `aidr_chat_completions` detect prompt injections—the biggest risk in LLM workflows.

03 Achieve compliance easily. The MCP lets you use `audit_search` to query audit logs using plain English, making compliance checks fast instead of manual report generation.

04 Control access at the source. You can manage user sessions using `authn_flow_start` or check permissions with `authz_check` before running any critical code path.

05 Vet external data sources. Before your agent uses an IP address, run `embargo_ip_check` to guarantee it's not coming from a restricted region.

Real-World Applications

Handling Customer Support Chats

A support agent needs to log a customer complaint that includes their account number and personal address. Instead of copying the whole thing into the system, your agent runs `ai_guard_text` first. It automatically strips all PII, so the security team gets a usable, non-sensitive report.

Building Financial Agents

A financial modeling bot needs to process quarterly reports that are structured JSON files containing salary data. The agent uses `redact_structured` to zero out all the sensitive salary fields before handing the data off for analysis.

Onboarding New Users

A new employee needs access to a specific shared drive. Instead of manually asking an admin, your agent first runs `authz_check` and verifies if the user's role has permission for that resource before granting access.

Processing External APIs

Your application receives data from a third-party API endpoint. Before processing it, your agent calls `embargo_ip_check` to ensure the incoming connection IP address is not from a prohibited country.

Patterns to Avoid

Assuming all inputs are safe

✗ AVOID

The developer writes code that sends user prompts directly into the LLM without any pre-screening, leaving the system open to prompt injection attacks.

✓ INSTEAD

Always run every incoming prompt through `ai_guard_prompt` and use `aidr_chat_completions`. This ensures malicious inputs are caught before they reach your model.

Manual data scrubbing

✗ AVOID

When a user complains, the team manually opens a spreadsheet, finds emails, and deletes them, which is slow and error-prone.

✓ INSTEAD

Let your agent handle it. Use `redact_text` to scrub PII from any text field automatically, guaranteeing consistency every time.

Ignoring historical activity

✗ AVOID

When an incident occurs, the team has to manually check multiple system logs and search through dates/user IDs using a basic filter.

✓ INSTEAD

Use `audit_search` to query your entire security history using natural language. Just ask: 'Show all login failures for marketing last week.' Done.

The Right Fit

You should use this MCP if your primary concern is the integrity of data flowing through or residing within your AI application. If you need to verify who can do what (authz), check where IP addresses are coming from (embargo_ip_check), or scrub sensitive fields from structured data (`redact_structured`), this is your tool. Don't use it if you simply need basic file storage—you'll want a different object store MCP instead. Only use the identity tools (`authn_` , *`authz_`*) when you are actively building user management features, not just reading logs.

The Security Headache of LLM Data Flow

Today, handling data means jumping through hoops. You build a feature that accepts user input, and then you have to worry about it: Does the prompt contain an email address? Is this IP address flagged as high risk? If it's structured JSON, did someone put an SSN in field 4B? Usually, you write custom validation checks for every single one of those things.

With this MCP, your agent handles all that overhead. You don't write the checkers; you just call them. Your workflow automatically runs `ai_guard_text` and `redact_structured`. What you get is a clean data output, guaranteed safe and ready for your model.

Control Access with Authz Tools

Before implementing any feature that changes state—like creating a user or accessing a protected folder—you currently rely on backend logic to check permissions. This means if you forget one conditional statement, the whole system breaks open.

Now, your agent uses `authz_check` and `authz_list_resources`. You simply ask: 'Does User X have permission for Action Y on Resource Z?' It's a single, reliable check that prevents unauthorized actions from ever completing.

Pangea (Security APIs) - 36 Tools

These tools allow you to scan content for threats, manage user identities, secure communication channels, and track activity logs across your entire application.

#	TOOL	DESCRIPTION
01	ai_guard_prompt	Analyzes and redacts malicious or sensitive content found in LLM prompts.
02	ai_guard_text	Scans any given text for PII, malicious patterns, and prompt injection attempts.
03	aidr_chat_completions	Guards entire LLM chat completions while logging and tracing every interaction securely.
04	audit_log_bulk	Creates multiple secure records of activity in your audit log at once.
05	audit_log	Records a single, specific security event into the tamper-proof audit trail.
06	audit_search_results	Retrieves pages of filtered search results from your historical audit log.
07	audit_search	Searches the entire audit history using natural language questions.
08	authn_flow_complete	Finalizes a user authentication flow and returns active session tokens.
09	authn_flow_start	Initiates the process for a user to sign up or log in.
10	authn_flow_update	Updates the state of an authentication flow, like submitting a password or OTP code.
11	authn_session_list	Retrieves a list of all currently active user sessions for management.
12	authn_session_logout	Invalidates and ends one or more existing user login sessions.
13	authn_user_create	Creates a new user account programmatically within your system.
14	authz_check	Determines if a specific user has permission to perform an action on a resource.
15	authz_list_resources	Lists all the resources that a given subject is authorized to access.
16	authz_tuple_create	Defines specific relationship rules for managing resource permissions (AuthZ).
17	domain_whois	Retrieves public WHOIS details associated with a given domain name.

#	TOOL	DESCRIPTION
18	embargo_ip_check	Checks if an IP address originates from any country that is currently under embargo.
19	embargo_iso_check	Verifies a two-letter ISO code against known lists of restricted countries.
20	file_scan	Scans an uploaded file to detect and flag any signs of malware or threats.
21	intel_reputation	Fetches reputation scores for specific domains, URLs, or file hashes.
22	ip_geolocate	Determines the physical location data associated with a given IP address.
23	ip_proxy	Checks if an incoming IP address is masking its true source because it's a proxy.
24	ip_reputation	Gets the overall reputation score and verdict for an entire IP address.
25	ip_vpn	Checks to see if an incoming IP address is masking its source because it's a VPN connection.
26	password_breached	Compares a password hash prefix against known databases of leaked credentials.
27	redact_structured	Removes specific fields from a JSON object based on defined paths (JSONPath).
28	redact_text	Scans and removes sensitive data from any block of plain text.
29	sanitize_file	Applies defined sanitization rules to a file, returning a cleaned version for safe use.
30	share_delete	Removes files or entire folders from the secure sharing area.
31	share_folder_create	Creates new organizational folders within the secure file sharing service.
32	share_get	Downloads a specific file or retrieves metadata about its properties.
33	share_list	Lists all objects within a designated bucket or folder, allowing for filtering.
34	unredact	Reverses the redaction process on data that was previously protected using FPE encryption.
35	user_breached	Checks if an email, phone number, or username appears in known public data breaches.
36	vault_decrypt	Performs cryptographic decryption on stored secret values.
37	vault_encrypt	Encrypts sensitive information to store it securely in the vault.

#	TOOL	DESCRIPTION
38	vault_get	Retrieves item details or specific secret values from your secure Vault storage.
39	vault_key_generate	Generates new symmetric or asymmetric cryptographic keys for use in the vault.
40	vault_secret_store	Stores a secret value or access token securely within the system's Vault.

See It in Action

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

U Redact any PII from this text: 'My email is john.doe@example.com and I live in New York'.

I've processed the text using `redact_text`. The redacted version is: 'My email is <EMAIL> and I live in <LOCATION>'.

U Search the audit logs for any 'login' events from the last 24 hours.

I've queried the audit logs using `audit_search`. I found 3 login events. Would you like to see the details for each entry?

U Check if the IP address 1.1.1.1 is from an embargoed country.

Running `embargo_ip_check` for 1.1.1.1... The check returned that this IP is not associated with any embargoed countries.

Frequently Asked Questions

01 **How do I use Pangea Security APIs MCP to redact JSON data?**

You call `redact_structured` and pass the JSON object along with a list of fields you want removed. This method is better than generic redaction because it uses specific JSON paths.

02 **Is Pangea Security APIs MCP good for checking suspicious IPs?**

Yes. Use `ip_geolocate` to find where the IP comes from, and then run `embargo_ip_check` or `ip_reputation` to see if it's flagged as a proxy or coming from an embargoed nation.

03 **What is the difference between audit_log and audit_log_bulk in Pangea Security APIs MCP?**

`audit_log` creates one single record for a specific event. Use `audit_log_bulk` when you need to create many related records at once, like logging ten user sign-ins.

04 **Can I use Pangea Security APIs MCP to check if my password is safe?**

Yes, use the `password_breached` tool. It compares a hashed version of your password against public databases of leaked credentials to tell you if it's compromised.

05 **Does Pangea Security APIs MCP help with general data storage?**

No, this MCP handles security and access control, not storage. You use `vault_encrypt` or `vault_secret_store` to secure the *data*, but you need a separate service for actual file storage.

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 `"pangea-security-apis": { "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

Pangea (Security APIs) 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 Pangea (Security APIs). 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	Pangea (Security APIs) MCP
Server ID	`019e38d2-0adb-73a7-9430-682b11f9cd23`
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/pangea-security-apis.