MCP SERVER | NO CODE | CLOUD HOSTED

HackerOne MCP

Triage Reports, Manage Bounties, Track Payments.

HackerOne connects your security team directly to bug bounty program operations. Use this MCP to manage vulnerabilities, track assets, and handle payments without leaving your chat window. You can list reports, change their status, add comments, award bounties, and view payment history—all through natural conversation.

A+ Quality Score 100/100

bug-bounty | vulnerability-management | security-research | penetration-testing | incident-response

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

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.

HackerOne MCP

10 tools available
Cloud-hosted on Vinkius

This MCP lets you run your vulnerability management workflows inside any AI client. You connect your organization account to get full control over bug bounty programs. Forget switching between report tabs and internal dashboards just to triage a finding. Your agent acts like a dedicated Security Program Manager, handling the day-to-day operations in real time.

You can list all submitted vulnerability reports or retrieve deep details on a specific one. Need to update something? You can change a report's state—marking it as triaged or resolved—and even award bounties directly from the chat. The system also lets you interact with asset definitions, check internal hacktivity feeds for recent discoveries, and monitor payment history. By connecting through Vinkius, this MCP gives your agent immediate access to all necessary program insights, making communication and workflow management simple.

Core Capabilities

01 — Reviewing vulnerability reports

Retrieve lists of submitted bug bounty reports or pull detailed information about a specific finding.

02 — Managing program assets

List and monitor the defined assets within your security programs to understand scope reachability.

03 — Updating report status and communication

Change a report's official state (like triaged) or add internal comments to communicate with researchers.

04 — Handling payments and bounties

Access the history of bounty payments and award rewards directly for specific vulnerability reports.

05 — Monitoring program scope

List all available bug bounty or VDP programs you have access to, along with their structured assets.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your HackerOne API Token Identifier and Value.

02 Your AI client connects the credentials, giving it read/write access to your bug bounty program data.

03 You simply ask your agent to perform an action—like 'List all high-severity reports from last week'—and get instant results.

The bottom line is you manage complex security programs and communications entirely through conversation, without ever opening the HackerOne website.

Built For

This MCP is built for people who live in a constant state of triage. If you're spending your Tuesday afternoons clicking between dashboards to track reports, manage bounties, and update statuses, this tool saves you hours.

Bug Bounty Manager

You use it to automate the process of awarding bounties, communicating status updates, and keeping a real-time overview of program health.

Security Engineer

You rely on it to instantly pull report details and severity ratings during triage, ensuring you don't miss critical information.

CISO / Director of Security

You use it to maintain a quick, actionable overview of incoming vulnerabilities and program performance without having to read dozens of detailed reports manually.

What Changes When You Connect

01 You instantly get a full list of submitted vulnerability reports and can pull deep details on any single finding using tools like `list_reports` and `get_report`. This eliminates the need to navigate multiple program dashboards just to see report metadata.

02 Bounty management becomes conversational. You can award bounties via `award_bounty`, update a report's status with `change_report_state`, or add internal notes using `add_report_comment`—all in one chat session.

03 Financial tracking is immediate. Instead of downloading CSV exports, you use `list_payments` to get the history of bounty payouts and monitor your rewards efficiently right from your agent.

04 Program scope remains clear. You can list available programs (`list_programs`) and check defined assets (`list_assets`) so that every security action is fully scoped before it starts.

05 Stay up-to-date without clicking anything. Use `list_hacktivity` to pull the latest internal or public discoveries, keeping your entire team informed on recent activity.

Real-World Applications

Handling a High-Severity Submission

A researcher submits a high-severity bug. Instead of manually checking the report ID and then opening a ticket to update its status, you ask your agent for details using `get_report`. You confirm it's critical, use `change_report_state` to mark it as 'Triaged', and immediately follow up with an internal note via `add_report_comment` telling the development team what to do next.

Running Monthly Financial Audits

It's time to audit payouts. Instead of logging into the payments tab, you ask your agent to list all recent bounties using `list_payments`. You can then cross-reference this data with `get_program` details to ensure every reward aligns with the active program scope.

Onboarding a New Team Member

A new engineer needs a quick overview of current vulnerabilities. Instead of giving them access to 10 separate reports, you ask your agent to list all open vulnerability submissions (`list_reports`). The results give them an immediate, actionable snapshot of the program's overall health.

Validating Program Scope

Before starting a new research sprint, you need to ensure coverage. You ask your agent to list all defined assets (`list_assets`) and compare that against the existing programs using `get_program` details. This quickly validates if the scope covers everything needed.

Patterns to Avoid

Manual report tracking

✗ AVOID

Jumping between HackerOne's main dashboard, the payments tab, and internal ticketing systems to figure out a report's status and who needs to be notified.

✓ INSTEAD

Use your agent to pull specific data. First, use `get_report` for details. Then, if necessary, use `add_report_comment` or `change_report_state`. Everything stays in the chat.

Ignoring payment history

✗ AVOID

Assuming a bounty was paid just because it was reported; having to manually check transaction logs later.

✓ INSTEAD

Always use `list_payments` to get an immediate and accurate historical record of all payouts. This confirms the financial state right away.

Overlooking program boundaries

✗ AVOID

Attempting to award a bounty or update a report that falls outside the officially defined scope.

✓ INSTEAD

First, use `list_programs` and then check `get_program` to confirm the official rules. This prevents accidental out-of-scope actions.

The Right Fit

Use this MCP if your workflow requires constant switching between status updates, asset checking, bug reporting, and financial tracking for a vulnerability program. You need an agent that acts as a single point of truth for all these functions. Don't use it just because you want to read reports; use `get_report` or `list_reports`. If your primary goal is only generating code based on findings, look at

generic API connectors instead. This MCP is about *operational management* and communication flow, not just data retrieval.

The pain of managing security reports across five tabs

Right now, triaging a report feels like juggling. You start on the main dashboard to list submissions, then click into a specific vulnerability to read details, and if you need to update its status, you have to switch to another tab. To communicate with the researcher or your internal team, you copy-paste notes into a separate chat tool. It's slow, error-prone, and takes you out of flow.

With this MCP, all those steps happen in one place. You tell your agent what needs doing—for example, 'Check report 12345 for details and change its state to resolved.' The agent handles the data retrieval and the status update without you ever leaving the conversation window.

HackerOne MCP: Direct Bounties and Triage Status

You don't have to manually award bounties or track payments. You just ask your agent to `award_bounty` for the specific report ID, and it processes the payment record instantly. Need to communicate a status change? Use `change_report_state`—it updates the system and logs an internal note automatically.

What's different now is that you move from being a data copy-paster to a decision-maker. Your agent manages the tedious mechanical steps, letting you focus on what matters: fixing the vulnerability.

HackerOne: 10 Tools for Security Ops

These tools give you granular control over every aspect of bug bounty management—from listing reports to awarding bounties and checking asset scope.

#	TOOL	DESCRIPTION
01	`add_report_comment`	Allows you to add a specific comment directly to any vulnerability report.
02	`award_bounty`	Processes and awards a bounty payment for a designated vulnerability report.
03	`change_report_state`	Updates the official state of a vulnerability report, such as marking it triaged or resolved.
04	`get_program`	Retrieves detailed information about a specific security program you manage.
05	`get_report`	Pulls comprehensive details for one particular vulnerability report ID.
06	`list_assets`	Generates a list of assets defined in your security programs, helping map out coverage.
07	`list_hacktivity`	Pulls the recent internal or public hacktivity feed to see what's been discovered lately.
08	`list_payments`	Retrieves a history of all bounty payments made through HackerOne.
09	`list_programs`	Shows you a list of bug bounty or VDP programs that are available to your account.
10	`list_reports`	Lists all vulnerability reports submitted within the scope of your current HackerOne program.

See It in Action

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

U List all vulnerability reports submitted this week.

I've retrieved your reports. You have 5 new submissions this week, including 'SQL Injection in /api/v1/users' (High) and 'BOLA in account settings' (Medium). Would you like to see the details for the High severity report?

U Award a $500 bounty to report ID 12345.

Bounty awarded! I've successfully processed a $500 reward for report 12345. The researcher has been notified. Should I add a comment to the report as well?

U Change the state of report 12345 to 'Triaged'.

State updated! Report 12345 is now marked as 'Triaged'. I've also added an internal note for the security team. What's the next step for this report?

Frequently Asked Questions

01 **How can I list all my open bug bounty reports using HackerOne MCP?**

You use the `list_reports` tool. This function pulls a comprehensive list of every submission tied to your active program, giving you an immediate overview of what needs attention.

02 **Does HackerOne MCP let me change a report status?**

Yes, you can use `change_report_state`. This tool updates the official status of a vulnerability report (like 'Triaged' or 'Resolved') and logs it for compliance records.

03 **How do I check past payments with HackerOne MCP?**

To review payouts, use `list_payments`. This function retrieves the entire history of bounty rewards associated with your program, helping you audit expenses quickly.

04 **Can I add a comment to a report using this MCP?**

Yes, that's what `add_report_comment` is for. You can communicate notes or internal findings directly into the record without needing to open the external platform.

05 **What information does HackerOne MCP provide about programs?**

You can use `list_programs` to see all available programs and `get_program` for deep details on a specific program's rules, scope, and assets.

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 `"hackerone": { "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

HackerOne 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 HackerOne. 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	HackerOne MCP
Server ID	`019d75ad-997e-719f-9dd9-04d7c22199cf`
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/hackerone.