MCP SERVER | NO CODE | CLOUD HOSTED

Bugcrowd MCP for AI Agents

Manage vulnerability reports and security programs from any source

Bugcrowd MCP connects your AI agents directly to Bugcrowd's entire security platform. You gain immediate access to manage bug bounty programs, track every vulnerability submission, and inspect target assets—all through natural conversation. It lets you orchestrate complex cybersecurity workflows without ever touching a dashboard.

C Quality Score 79.37/100

bug-bounty | vulnerability-management | security-testing | crowdsourced-security | incident-response | cybersecurity

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

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.

Bugcrowd MCP

10 tools available
Cloud-hosted on Vinkius

Stop juggling tabs and copy-pasting data between your Bugcrowd console and your ticketing system. This MCP lets you run your entire bug bounty process directly from your AI agent. Instead of navigating through menus to see if a submission is triaged or what the scope of an active program is, you just ask. Your agent pulls the details on demand, giving you instant oversight of vulnerability reports, security programs, and specific assets.

For example, you can tell your agent to list all currently running bug bounty engagements, then ask for the full metadata on a single submission. It's like having an expert analyst sitting next to you who has immediate read-access to every piece of data. You get this power centralized through Vinkius, connecting it to any compatible AI client, letting your team stay focused on fixing bugs instead of finding reports.

Core Capabilities

01 — Track and Manage Vulnerability Submissions

List all bug reports across multiple programs or pull deep metadata for a single vulnerability submission.

02 — Orchestrate Security Programs

See which security programs are active, what their defined scopes are, and what rewards they offer.

03 — Monitor Bug Bounty Engagements

Get an overview of specific crowd executions or penetration tests that are currently running.

04 — Inspect Target Assets

View the complete inventory and detailed metadata for all assets in scope (targets) for your organization.

05 — Log New Vulnerability Reports

Quickly create a new submission record from an external source using plain language prompts.

One Click on Vinkius — From Prompt to Execution

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

01 Subscribe to this MCP and provide your Bugcrowd API Access Token.

02 Connect the credentialed MCP to your preferred AI client (like Cursor or Claude).

03 Ask your agent a question, like 'List all active bug bounty programs,' and it returns the structured data directly.

The bottom line is you use natural conversation to interact with complex security data that used to require manual dashboard navigation.

Built For

Security Engineers, Vulnerability Managers, and CISOs who spend too much time in dashboards. If your job involves tracking down a specific bug report or comparing program scopes, this is for you.

Vulnerability Manager

Triage reports by listing all submissions to identify critical flaws and retrieve detailed metadata on any single finding.

Security Engineer

Check the scope of a specific security program or list all targets to ensure coverage before starting a pentest.

CISO / Security Lead

Monitor overall program health by listing active engagements and reviewing organizational settings in one place.

What Changes When You Connect

01 Stop manually checking submission statuses. You can list all vulnerability submissions or get deep details on a single report using the `list_submissions` or `get_submission` tools, making triage instantaneous.

02 Never lose track of program boundaries again. Instantly view and retrieve detailed scope and reward information for any active security program by calling `get_program`.

03 Coordination is simplified. Use `list_targets` to quickly see every asset in scope, or use `get_target` to inspect specific target details without leaving your chat window.

04 Keep compliance current. Pull organizational settings and core account info using `get_organization_info`, giving you a single source of truth for governance.

05 Improve reporting speed. You can create new findings directly via the `create_submission` tool, logging bugs instantly from an external source.

Real-World Applications

A vulnerability manager needs to check if a newly found bug falls within program scope.

Instead of navigating multiple dashboards, the agent is asked: 'Does this specific flaw count for my main web app?' The agent runs `get_program` and checks the details against the submission metadata using `get_submission`, giving an immediate yes/no answer.

A security engineer needs to start a new penetration test on assets that haven't been inventoried.

The agent runs `list_targets` first. After confirming the needed assets, they use `get_target` repeatedly for specific details before initiating the engagement through `list_engagements`.

A CISO needs a quick overview of all active security tests across different teams.

The agent runs `list_programs` to see which programs are running, then uses `list_engagements` to pull the status and scope for every single bug bounty or pen test.

A researcher finds a critical zero-day vulnerability while reviewing internal documentation.

They simply tell their agent: 'Log this finding now.' The agent uses `create_submission` to file the report immediately, ensuring it's logged with all necessary metadata.

Patterns to Avoid

Treating security data like simple documents

✗ AVOID

Copy-pasting a list of 50 targets into an email and asking for status updates on each one manually.

✓ INSTEAD

Use the MCP to run `list_targets` once, then ask your agent to filter those results based on 'outdated' or 'unscanned' criteria. This gives you structured data instantly.

Forgetting program boundaries

✗ AVOID

Assuming a bug found in one system is covered by another program's scope because they seem related.

✓ INSTEAD

Always run `get_program` first. This tool precisely defines the boundaries and rewards for that specific security program, preventing incorrect assumptions.

Missing context on a single submission

✗ AVOID

Getting a raw ID of a bug report and not knowing who reported it or which program owns it.

✓ INSTEAD

Use `get_submission` with the ID. This provides all the necessary metadata, connecting the specific flaw back to its source program and reporting user.

The Right Fit

Use this MCP if your process requires constant cross-referencing of security data—checking a bug report against a program's scope, or listing targets before starting an engagement. It excels when you need visibility across multiple reports (`list_submissions`) and programs (`list_programs`). Don't use it if you just need to write a general security policy; those are document tasks better suited for text generators. If your main goal is simply data storage without retrieval, you might only need `create_submission` . However, because of its broad scope (covering submissions, programs, and targets), this MCP remains the best single point of truth for bug bounty workflows.

Bugcrowd MCP: Centralizing vulnerability reporting oversight

Today, tracking a serious vulnerability is a nightmare. You jump between Bugcrowd's dashboard, your internal Jira board, and email chains. To check the status of a single finding, you copy an ID, paste it into one tool, then open another tab to see the target asset list, and finally switch back out.

With this MCP, you just ask: 'What is the current triage status for bug sub_99283?' Your agent runs `get_submission` instantly. You get all the metadata—status, program association, submitter details—returned in a clean chat format. It cuts through the manual dashboard work and gives you the exact answer immediately.

Bugcrowd MCP: Coordinating security programs and assets

Before running any test, you have to manually verify that the target is in scope. This means cross-checking a list of assets against several different program rules. If you forget one step, your entire engagement might be invalidated.

The MCP handles this coordination automatically. You can run `list_targets` for an inventory check and then use `get_program` to verify that the needed asset is explicitly covered by the desired security program scope. It makes governance reliable.

Bugcrowd: 10 Tools for Vulnerability & Target Data

Use these tools to list, retrieve details on, and create data points covering your entire security program lifecycle, from initial target listing to final submission.

#	TOOL	DESCRIPTION
01	`create_submission`	Allows you to quickly log a new vulnerability submission report into the system.
02	`get_engagement`	Retrieves detailed information for one specific crowd bounty or penetration test engagement.
03	`get_organization_info`	Pulls core configuration details and settings about your overall Bugcrowd organization account.
04	`get_program`	Gets comprehensive details, including scopes and rewards, for a specific security program.
05	`get_submission`	Retrieves all the detailed metadata associated with one particular vulnerability submission report.
06	`get_target`	Gets full details about a specific asset or target that is in scope for testing.
07	`list_engagements`	Lists all active crowd engagements, such as bug bounties and pen tests, currently running.
08	`list_programs`	Retrieves a list of every security program you have set up on the platform.
09	`list_submissions`	Lists all vulnerability submissions, giving you an overview of everything reported across your programs.
10	`list_targets`	Provides a comprehensive list of every asset defined as 'in scope' for your organization or program.

See It in Action

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

U Show me all active programs and if my main web app is listed as a target.

Active Programs:
- Vinkius Bounty (ID: prog_1)
- Main Web App (ID: prog_2)
- Legacy Systems (ID: prog_3)

Target Check for 'Main Web App':
Yes, the asset `web-app.company.com` is listed in scope under Program ID `prog_2`. The details show it supports authentication and accepts submissions.

U List all vulnerability reports submitted last week that haven't been triaged yet.

Vulnerability Submissions (Un-triaged)

SUBMISSION ID	TITLE	PROGRAM	STATUS
sub_99283	XSS on Login Page	Main Web App	Unresolved
sub_10045	Broken Auth Flow	Vinkius Bounty	Pending Review

This list shows 2 findings needing immediate attention.

U I found a new bug in the legacy system. Can you log it now?

Submission Created Successfully
The following vulnerability report has been logged:
- **Title:** SQL Injection in /api endpoint
- **Program:** Legacy Systems (prog_3)
- **ID Assigned:** sub_99284

You can now track this finding's status using its new ID.

Frequently Asked Questions

01 **Can I check the scope of a security program using the agent?**

Yes! Use the `get_program` tool with the Program ID. Your agent will fetch the detailed metadata, including targets and scope descriptions, from Bugcrowd.

02 **How do I list all the vulnerability submissions for my account?**

Simply ask the agent to `list_submissions`. It will retrieve the latest vulnerability reports from your Bugcrowd account, including titles and statuses like 'triaged' or 'resolved'.

03 **Does the integration allow creating a new submission?**

Yes. Use the `create_submission` action and provide the title and description. You can also associate it with a specific program by providing the `program_id`.

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 `"bugcrowd": { "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

Bugcrowd 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 Bugcrowd. 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	Bugcrowd MCP
Server ID	`019d7565-26d0-72e7-ba96-70736dbd7de2`
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/bugcrowd.