MCP SERVER | NO CODE | CLOUD HOSTED

Wayback MCP

Track Web Content Changes Across Decades

Internet Archive Wayback MCP accesses the world's largest web archive, giving you access to over 800 billion archived web pages spanning decades of internet history. Check a URL's current preservation status, analyze its capture timeline, and find specific content—like images or PDFs—from any year. It lets researchers track content changes, legal teams verify evidence, and developers study how websites evolved.

A+ Quality Score 100/100

web-archiving | url-history | snapshot-analysis | digital-preservation | internet-history | cdx-api

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/internet-archive-wayback

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.

Internet Archive Wayback MCP

10 tools available
Cloud-hosted on Vinkius

Need to know what a website looked like five years ago? This MCP connects your AI agent directly to the Internet Archive Wayback Machine. Instead of guessing or relying on single-point snapshots, you can check a URL's full history—a massive archive spanning over 25 years. Your agent verifies if a page was ever archived and finds its most recent snapshot instantly. You can dig into granular details: Did they change their logo? Find all JPEG images from 2018. Was the site down on a specific date? Check the HTTP status codes for that year. The power of this data is channeled through Vinkius, making historical web analysis available to any compatible client. It's ideal for journalists tracking deleted content or developers comparing design iterations over time.

Core Capabilities

01 — Check URL availability

Determine if a specific website address has been archived and get the timestamp of its closest preserved version.

02 — Analyze capture timeline

Find out when a page was first captured or what the most recent snapshot is, giving you clear start and end points for content history.

03 — Filter by resource type

Limit searches to specific file types like PDFs, images, or stylesheets to pinpoint necessary historical assets.

04 — Track status codes

Analyze capture records specifically for HTTP error or success codes (like 404 or 200) across a period of time.

05 — Discover domain footprint

Find all archived subdomains associated with a main website, helping map out an entire organization's historical online presence.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/internet-archive-wayback** — connect your AI agent in three steps.

01 Subscribe to this MCP on Vinkius. No API key is needed; the connection is open and public.

02 Your AI agent sends a query—for example, 'Show me all captures for X URL in 2015'—to the connected archive data.

03 The MCP executes the necessary checks and returns structured historical metadata detailing the status codes, dates, and types of captured content.

The bottom line is that your AI agent treats the vast web archive like a searchable database, letting you query specific pieces of history without needing to browse the raw data yourself.

Built For

This MCP serves anyone whose job involves tracking content over time. Journalists need proof of what was online and when. Lawyers require verifiable evidence of website text or design for legal action. Developers use it to model product evolution, while academics study the internet's changing face.

Investigative Journalist

They check if a controversial statement made by an official was ever published online and find the exact date using `get_first_capture`.

Legal Compliance Officer

They prove that specific website content existed on a certain day, running checks to preserve evidence of deleted or altered material.

Frontend Developer

They compare the structure and design of their site from five years ago against today, using `get_captures_by_mime_type` for historical CSS files.

What Changes When You Connect

01 Instantly verify content history. Use `check_availability` to confirm if a page was ever archived, saving you the manual effort of checking multiple archives.

02 Analyze site evolution with precision. Instead of guessing, use `get_captures_by_year` to pull all snapshots from a specific year for comparison.

03 Track content changes over time. Use `get_first_capture` and `get_latest_capture` together to measure the gap between a page's debut and its most recent update.

04 Build domain maps easily. The `get_subdomain_captures` tool reveals the full historical footprint of an organization, finding subdomains you didn't know existed.

05 Filter data for specific evidence. Need only to check if images were posted? Use `get_captures_by_mime_type` to filter out irrelevant text and status codes.

Real-World Applications

Tracking a Journalist's Claim

A journalist needs proof that a rival company made a claim in 2017. They ask their agent to check the URL, using `get_captures_by_year` and then `get_first_capture`. The MCP reports all available snapshots from 2017, allowing them to pinpoint the exact date and status code of the original post.

Legal Discovery for a Breach

A compliance officer needs evidence that a specific policy was visible on a website in late 2021. They use `get_captures_by_status` to filter out error pages and then check the resulting records to confirm the presence of the required text block.

Developer Comparing Design Changes

A developer wants to see how a site's structure changed over time. They use `get_subdomain_captures` first, then run `get_captures_by_mime_type` for CSS files across multiple years to analyze the evolution of stylesheets.

Academic Research on Web Trends

A historian wants to study how a particular industry presented itself online over 20 years. They use `get_capture_count` and `get_captures_by_year` repeatedly across different domains to quantify the change in web presence.

Patterns to Avoid

Guessing content dates

✗ AVOID

Manually searching Google or relying on a single Wayback Machine interface view, which only gives you an estimate and doesn't provide metadata like status codes.

✓ INSTEAD

Use the specific tools. To check for any available date, use `check_availability`. If you need to analyze every year between 2015 and 2017, call `get_captures_by_year` for each one.

Missing the scope

✗ AVOID

Thinking a URL's history is limited to just that page. You might miss content from related subdomains or different file types.

✓ INSTEAD

First, use `get_subdomain_captures` to map out the whole domain. Then, check all found URLs using `get_captures_by_mime_type` for a complete picture.

Ignoring status codes

✗ AVOID

Assuming that if content was archived, it must have been functional (HTTP 200). You might miss when the site was inaccessible.

✓ INSTEAD

Always run `get_captures_by_status` alongside your date checks. This reveals exactly when a page returned an error code like 404 or 500.

The Right Fit

Use this MCP if your goal is historical content verification, tracking evolution, or establishing timelines of online presence. If you need to prove *when* something was said or seen, this tool is necessary. Don't use it if you simply want the current status; for that, a standard web request works fine. You should only rely on its specialized tools —like `get_first_capture` or `get_captures_by_mime_type` —to get granular data points like dates and file types. If your need is purely comparative (e.g., 'Is the current site better than a competitor's'), you might just use standard web scraping, but if you need to compare *historical* versions of sites, this MCP is non-negotiable.

Finding Web History Is a Click-Heavy Nightmare

Today, digging into old website content means opening the Wayback Machine and manually inputting dates. You copy a URL, check one year, then another. If you want to know if an image was posted in 2015, you have to filter by date, then filter by MIME type (image/jpeg), and hope you didn't miss anything. It’s slow, tedious, and easy to misread.

With this MCP, your agent handles the whole process. You tell it, 'Find all JPEG images for X URL between 2015 and 2017.' Your AI client gets back a clean list of data points, complete with timestamps and status codes. The guesswork is gone.

Accessing Web Archive Details with `get_cdx_captures`

Instead of clicking through dozens of different yearly views to piece together a timeline, you run one query. You use the `get_cdx_captures` tool which pulls every available metadata point—the timestamp, status code, file size, and MIME type—into one record set.

Now you have the full picture in plain data. It's not just 'archived.' It tells you *how* it was archived, letting you analyze patterns that were previously hidden behind layers of manual clicking.

Internet Archive Wayback: 10 Tools for Deep Web Analysis

These tools give your agent specific ways to query the web archive, letting you filter captures by type, time, or status code instead of just viewing a general history overview.

#	TOOL	DESCRIPTION
01	`get_captures_by_mime_type`	Finds archived pages filtered by the specific file type, like showing only PDFs or images from a URL's history.
02	`get_captures_by_status`	Filters captured records by HTTP status code (e.g., finding all 404 errors across a domain over time).
03	`get_captures_by_year`	Retrieves all archived snapshots for a specific calendar year, allowing you to analyze content frequency during that period.
04	`get_cdx_captures`	Gets a detailed list of every capture record, including the timestamp, MIME type, and file size for full archival analysis.
05	`check_availability`	Quickly determines if a URL has been archived and returns the date of the closest available snapshot.
06	`get_captures_collapsed`	Shows unique page captures for a given URL, eliminating redundant entries so you only see distinct versions.
07	`get_capture_count`	Calculates and returns the total number of times an entire URL has been archived over its history.
08	`get_first_capture`	Identifies and retrieves metadata for the earliest preserved version of a URL, showing when it was first captured.
09	`get_latest_capture`	Gets the most recent archived snapshot of a page, giving you the newest recorded version available.
10	`get_subdomain_captures`	Maps out an entire domain's historical presence by finding all captured subdomains (e.g., www and help).

See It in Action

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

U Check if https://example.com has been archived.

✅ URL is archived! Closest snapshot: 20240115120000 (January 15, 2024). View at: https://web.archive.org/web/20240115120000/https://example.com. Total captures: 1,247 over 28 years.

U Show me all captures of https://example.com from 2020.

Found 52 captures of https://example.com in 2020. First: 2020-01-03 (status 200), Last: 2020-12-28 (status 200). Average capture frequency: ~1 per week. All returned HTTP 200 (OK).

U Find all subdomains of archive.org that have been captured.

Found captures for 15 subdomains of archive.org: www, web, developer, donate, help, blog, advancedsearch, petabox, us.archive.org, and others. Most captures are from www and web subdomains. Oldest capture dates back to 1998.

Frequently Asked Questions

01 **How do I check if a URL was ever on the Internet Archive Wayback using the Internet Archive Wayback MCP?**

You run `check_availability` with the target URL. This tool immediately tells you if the page has been archived and provides the timestamp of the closest preserved version.

02 **Can I find out when a website was first online using get_first_capture?**

Yes, running `get_first_capture` gives you the initial metadata for the earliest snapshot available. It includes the timestamp and status code of that very first recorded version.

03 **How do I analyze a domain's full history using get_subdomain_captures?**

Use `get_subdomain_captures` with the root domain. This tool discovers and lists all associated subdomains that have been captured, letting you map out the entire corporate footprint.

04 **What is the best way to filter for images in a specific year?**

You combine two tools: first, use `get_captures_by_year` to narrow down the date range. Then, refine that list using `get_captures_by_mime_type` and specify 'image/jpeg' or similar.

05 **Do I need an API key for Internet Archive Wayback MCP?**

No. This connection is free and public, meaning you don't have to worry about managing credentials; just connect via your preferred AI client.

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 `"internet-archive-wayback": { "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

Internet Archive Wayback 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

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Internet Archive Wayback MCP
Server ID	`019d75b6-74cf-725b-bd9f-44abe74f65bc`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE