Spellwarden research

MCP server security, November 2026

What an unauthenticated probe of every host in the official MCP Registry found

Data collected	10 November 2026
Registry entries	59,944 active servers
Hosts probed	30,402, one endpoint each
Scanner	Spellwarden 1.10.0, rules 2026.11.3, Apache-2.0

62%

of the MCP hosts in the official registry listed their tools to a client that had not signed in.

spellwarden.ai · Licensed CC BY 4.0: quote and reuse with a link to the source

Summary

The short answer

On 10 November 2026 we took every server listed in the official MCP Registry (59,944 active entries) and sent one unauthenticated probe to each of the 30,402 hosts behind their remote endpoints. Of the 26,548 hosts that answered as MCP servers or asked for a sign-in, 16,470 (62%) listed their tools to a client that had not signed in, and 9,691 of those lists include at least one tool that does not declare itself read-only. Only 769 hosts speak the 2026-07-28 revision of the protocol, and 6 of them also pass every readiness check for it. Instruction-like text in tool descriptions is uncommon: about one open tool list in a hundred steers the model towards its tool, and fewer ask it to keep something from the user.

62%
of MCP hosts listed their tools without any sign-in (16,470 of 26,548)

9,691
open tool lists include a tool that does not declare itself read-only

at least 15,064
tools that declare themselves destructive were listed to an anonymous client

36%
of OAuth servers support Client ID Metadata Documents; most of the rest rely on dynamic registration

5.2%
of hosts that report a version speak 2026-07-28 (769 of 14,664)

6
of the 769 hosts on 2026-07-28 also sign clients in with CIMD and use none of its removed or deprecated features

What this means for a team connecting agents

- A listing in the registry tells you who published an entry. It does not tell you whether the server needs a sign-in, what its tools can do or what its tool descriptions say. Check each server before an agent connects.
- An open tool list is often deliberate (public documentation and public data). It becomes a risk when the tools write, delete or spend money and nothing asks who the caller is.
- Tool descriptions are text the model reads and follows. Read them, or have a scanner read them, before you allow a server.

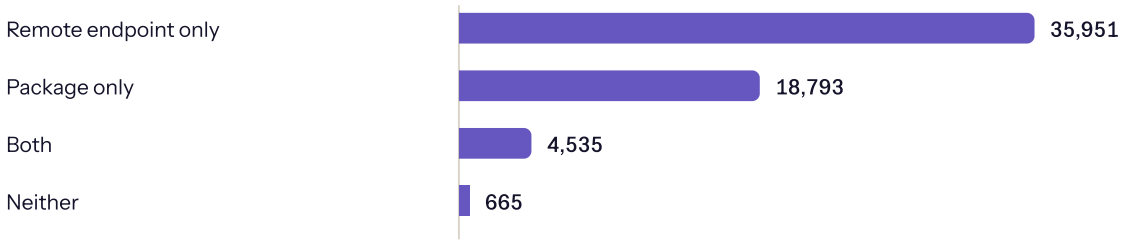
We never called a tool, sent a credential or tried to get past a sign-in. This report names no server and no publisher. Method and limits are on the last pages; the scanner is open source and the scripts are available on request, so the numbers can be checked.

The registry in November 2026

The official MCP Registry held 59,944 active servers at 2026-11-10 06:07:30 UTC, counting the latest version of each. 40,486 of them (68%) offer a remote endpoint that a client reaches over HTTPS, 23,328 ship a package that runs on the user's machine (npm, PyPI, container images and others), 4,535 offer both and 665 offer neither. The remote entries point to 41,643 distinct URLs, 41,238 of them on 30,402 hosts; 405 URLs contain a placeholder that each user fills in and were left out of the probe.

Publishing is concentrated at the top and spread thin below it. 37,147 namespaces published at least one server and 33,803 published exactly one, while 21 namespaces with 100 or more servers each account for 11,398 entries (19%). The largest single namespace holds 2,366. 57% of entries sit under an `io.github.*` namespace, which the registry grants to whoever authenticates as that GitHub user or as an owner of that organisation. The registry is also young and moving fast: for 31,680 entries (53%) the latest version was published in October or November 2026.

Registry entries by how a client reaches them



59,944 active servers, latest version of each, 10 November 2026.

What a registry listing proves. The registry is in preview and holds metadata only; its documentation says it delegates security scanning to package registries and aggregators. Its namespace check proves that whoever published an entry controls the GitHub account or the domain in the entry's name. The project's maintainers state that the registry does not verify that a remote URL belongs to the publisher, and since 6 November 2026 several entries may list the same URL. A listing is a statement of who published it, not a security review.

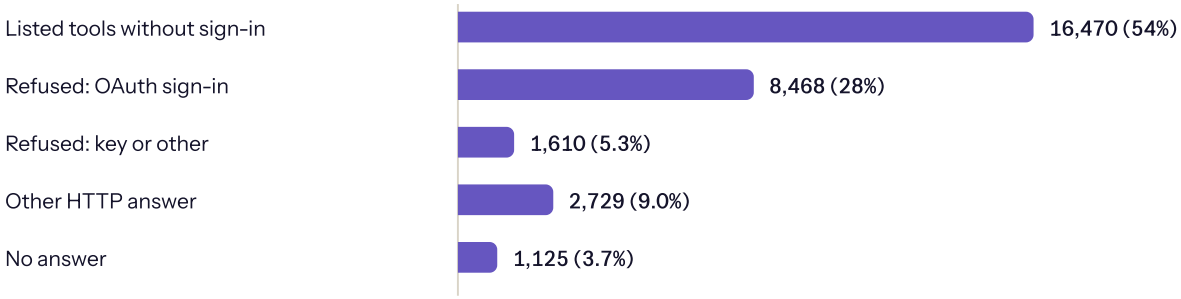
Because one publisher can register thousands of servers on one host, we probed one endpoint per host, not one per entry. 29,226 hosts carry a single registered endpoint; 69 carry ten or more. Counting hosts keeps a single bulk publisher from setting the result.

2 · Who can connect

How the hosts answered

Each probe made at most nine requests, as an MCP client does when it first connects: the 2026-07-28 `server/discover` call, then `initialize`, then `tools/list`, and the OAuth metadata documents (RFC 9728 protected resource metadata and the authorization server's metadata) where it was refused. 29,277 of 30,402 hosts (96%) gave an HTTP answer.

How each host answered an unauthenticated MCP client



30,402 hosts, one endpoint each. Other HTTP answers: 400: 430, 402: 208, 404: 769, 405: 100, 429: 241, 5xx: 589, other statuses: 316, and 76 hosts that answered 200 without an MCP answer.

Two groups matter most. 16,470 hosts answered `tools/list` without asking who was calling; 1,669 of them listed tools without completing the `initialize` handshake, and 912 of those sit on one platform's domain. 10,078 refused, 9,929 with 401 and 149 with 403 (which is sometimes a block rather than a request for credentials). Together they are the 26,548 hosts this report calls MCP hosts; the open group is 62% of them.

208 hosts answered `402 Payment Required`, the status that pay-per-call schemes use, and 379 registry remotes declare a `payment-signature` header. Paid agent calls are already part of the registry.

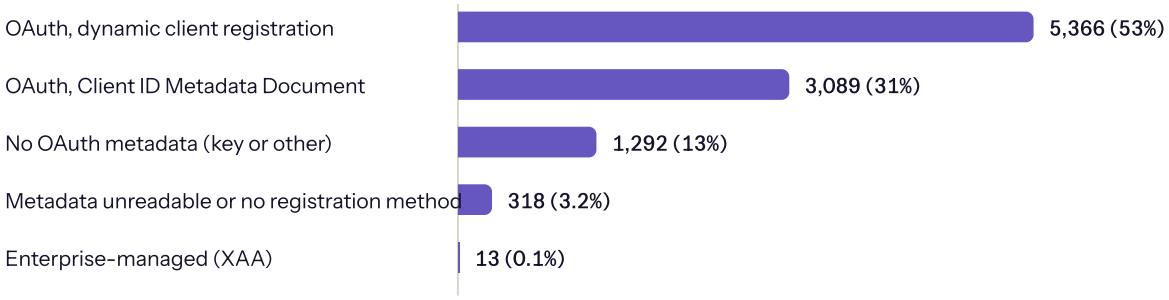
---

3 · Sign-in

How servers ask for a sign-in

Of the 10,078 hosts that refused an anonymous client, 8,577 (85%) publish authorization server metadata that a client can read, and 8,468 (84%) advertise a way for a client to register: dynamic registration, CIMD or enterprise-managed authorization. The MCP authorization specification now prefers Client ID Metadata Documents (CIMD), where the client identifies itself with a URL it controls, over dynamic client registration (DCR), where any client can register itself. The 2025-11-25 specification says clients and authorization servers SHOULD support CIMD and MAY support DCR for backwards compatibility, and the 2026-07-28 revision deprecates DCR in favour of CIMD. Authorization itself stays optional; a server that does ask for a sign-in MUST publish RFC 9728 protected resource metadata.

Sign-in method of hosts that refused an anonymous client



10,078 hosts that answered 401 or 403. Classification as in the open-source scanner.

5,366 OAuth hosts (63%) rely on dynamic registration and do not advertise CIMD support. 1,292 hosts refuse without publishing protected resource metadata, so a current client cannot discover an authorization server; in practice that is an API key, a token issued elsewhere, a custom scheme, or OAuth under the older 2025-03-26 discovery rules. The registry metadata points the same way: 7,258 remote entries (17%) tell the client to send a header, 5,626 of them an `Authorization` header and 939 an `X-API-Key`. A key pasted into configuration is usually long-lived, shared by every agent that reads the file and revoked only by rotating it for all of them.

4 · Open tool lists

Tool lists open to anyone

16,435 hosts listed at least one tool without a sign-in (62% of MCP hosts). The median open list has 6 tools. An open list is not a flaw in itself: documentation search and public data are meant to be open. What a list contains decides the risk, and MCP lets each tool say whether it only reads (`readOnlyHint`) or can destroy data (`destructiveHint`).

6,744
open lists where every tool declares itself read-only

9,691
open lists with at least one tool that does not (59% of open lists)

15,064
tools in open lists that declare themselves destructive (at least: some lists were not read whole)

Across the 196,542 tools in lists we could read whole, 116,663 (59%) declare themselves read-only and 41,165 (21%) carry no hint either way. The hints are the server's own claim, so a read-only hint is a statement, not a guarantee.

We listed tools; we never called one. A server can list its tools openly and still check the caller on each call, so these figures count what an anonymous client can see, not what it can do. The scanner's rule for this (SW-MCP-001) grades an open list as low when it holds no tool, medium when every tool declares itself read-only, and high otherwise or when the list could not be read whole; 38 lists are high for that last reason alone and are not in the count above.

---

5 · Tool descriptions

Instruction-like text in tool descriptions

A model reads every tool description it is given and tends to follow what they say. Tool poisoning hides instructions there: call this tool before any other, do not tell the user, pass along the conversation. The scanner's rule SW-MCP-011 looks for that kind of text in each listed tool's description.

The rule fired on 1,444 tool descriptions on 967 hosts, 5.9% of the hosts with an open tool list. On 35 hosts the only match was steering text (“always use this tool”); on 932 at least one description matched a concealment, override or injected-instruction pattern, or steering text next to a word for conversation or credential material.

967
hosts where the pattern rule fired

about 145
hosts steer the model towards their tools, estimated from the reviewed sample

6 of 60
sampled hosts ask the model to hide something, follow a script or collect data (1 clearly)

The rule is a pattern match, and on this population it over-matches. On a pass that read every flagged description in full, the word *hidden* matched on 620 of the 964 flagged hosts, and on 543 it was the only match; tool descriptions mostly use the word in its ordinary sense (hidden files, hidden form fields, hidden game content). To see what the flags hold, we drew a random sample of 60 of the 965 flagged hosts (seed 20261110). A model reviewer with read-only access labelled the full recorded text (up to

1,000 characters) of every flagged description in it against written definitions, giving each borderline case the more serious label, and we read the decisive sentences of three of the 6 serious cases ourselves.

9 of the 60 only steer the model towards the tool: call it first, always use it, or use it instead of answering directly. 6 go further. One is clear: it tells the model to call the tool without asking and not to mention it to the user afterwards. The other 5 are borderline: not mentioning a paid product unless the user asks, never showing a link that works as a credential, relaying the server's text word for word, a scripted persona, and pre-filling personal details from the conversation. The remaining 45 are benign: the word hidden as above, or a privacy note such as “the key is never shown again”.

Scaled to the 965 flagged hosts, that is about 145 hosts whose only instruction-like text steers the model (95% interval 78 to 252) and about 96 whose descriptions ask it to hide something, follow a script or collect data (45 to 194). Counting only clear cases like the one above gives about 16. Against the 16,435 open tool lists, steering is about 0.9% and the rest about 0.6%. A sample of 60 is small, so read these as orders of magnitude.

So instruction-like text is uncommon in the registry's open tool descriptions: roughly one open list in a hundred steers the model, and fewer ask it to keep something from the user. Both are worth reading before an agent connects: a description that tells the model to call a tool first, for any message, decides what the agent does before the user's own request does.

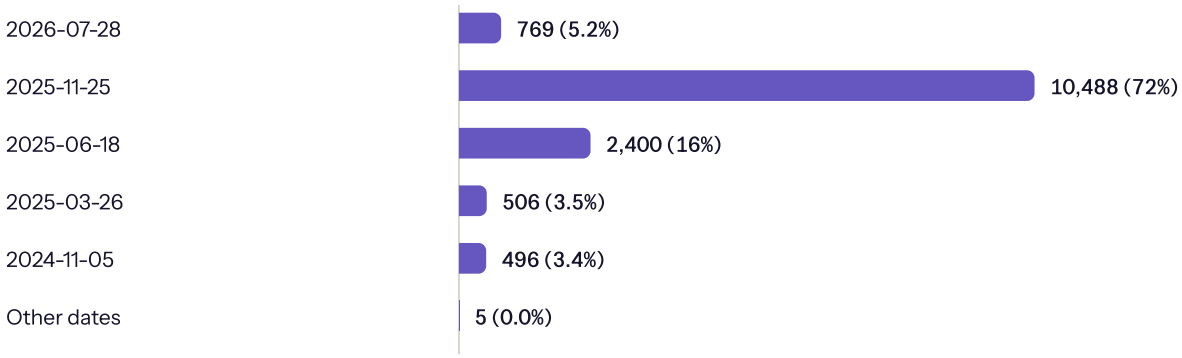
We could read descriptions only on servers that list tools without a sign-in. Servers behind a sign-in were not read, so this is a floor for the registry, not a rate for it.

6 · Protocol versions

Protocol versions and readiness for 2026-07-28

14,664 hosts reported a protocol version. 10,488 (72%) negotiate 2025-11-25 and 769 (5.2%) report support for the 2026-07-28 revision. 1,088 answered `server/discover`, the call that revision introduces.

Newest protocol version each host reported



14,664 hosts that reported a version: the newest date in server/discover, else the version negotiated in initialize.

The scanner calls a server ready for 2026-07-28 when it supports that revision, signs clients in with CIMD (or enterprise-managed authorization) and uses none of the features the revision removes or deprecates. A host with no sign-in cannot pass that check by construction. Of the 769 hosts on the revision, 6 meet all three. Among the hosts that completed the handshake, 1,569 (11%) still hand out protocol sessions (`Mcp-Session-Id`), which the revision removes, and 715 still offer Roots, Sampling or Logging, which it deprecates. A further 56 hosts, none of which completed the handshake, answer only over the deprecated HTTP+SSE transport.

---

7 · Packages

Packages that run on the user's machine

25,455 package entries describe servers that a client downloads and runs locally, 24,873 of them over standard input and output. By registry: cargo 124, mcpb 2,231, npm 16,143, nuget 174, oci 1,729, pypi 5,054. Version pinning in the metadata is mostly good: all but 15 npm, PyPI, NuGet and Cargo entries name an exact version, and every MCP bundle carries a SHA-256 of the file. Container images are the gap: 17 of 1,729 images are pinned by digest, and the rest by a tag that the publisher can move.

9,916 package entries ask the user for at least one secret in an environment variable. Each of those secrets typically ends up in a client configuration file on a developer's machine, which is where our scanner's configuration rules look first.

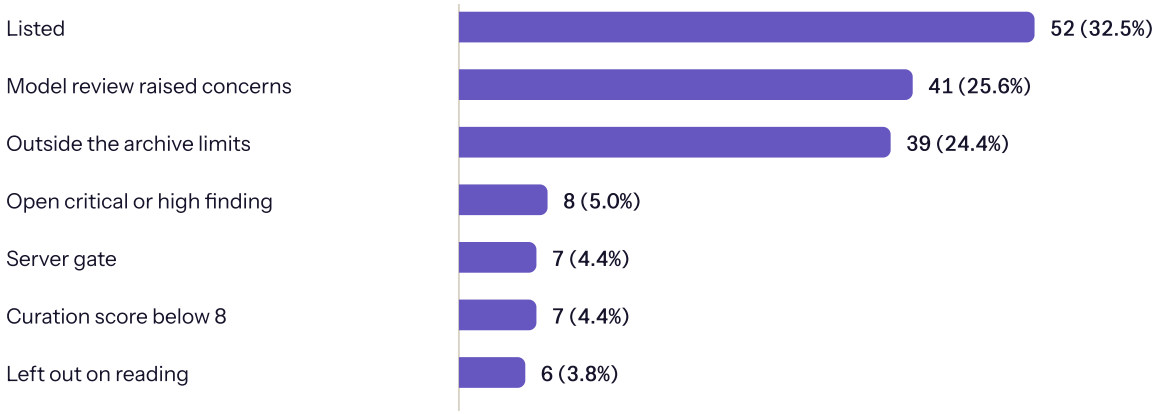
The probe does not reach local packages: a package has no endpoint until a user runs it. Scanning package contents is the job of the scanner's skill and configuration rules and is outside this report.

8 · A closer look

What hand vetting adds

A probe sees what a server says about itself. Spellwarden's curated catalog goes further: each candidate is scanned with every rule, its sign-in mode is recorded, a model reviews its files and an operator reads the result before anything is listed. In the first round (2026-11-06 to 2026-11-07), 160 popular public skills, plugins and MCP servers were imported and 52 (32.5%) were listed at the time.

Outcome for each candidate in the first catalog round



160 candidates, November 2026. Outside the archive limits: too many files, symlinks, unsafe paths or too large to scan whole.

Candidates that were not listed mostly failed on process rather than malice: 39 could not be scanned whole (too many files, symlinks, unsafe paths or size), 41 drew concerns from the model review, 8 had an open critical or high finding, 7 failed the server gate (6 answered without a sign-in and are not public documentation, 1 had instruction-like tool text), 7 scored below the curation bar and 6 were left out on reading. That is the gap between being listed somewhere and being fit for an agent at work.

9 · Context

How this compares

MCP server security has been measured before, from three directions: internet-wide scans that find exposed servers through search engines such as Shodan and FOFA, studies of server source code, and, during 2026, probes of servers listed in the official registry. The table gives the studies closest to this one, with their figures as each source states them. Samples, units and dates differ, so the numbers are context, not a trend line.

Study	Sample	Headline finding, as published
Knostic, July 2025	1,862 internet-exposed servers found through Shodan; 119 checked by hand	All 119 checked servers listed their tools without authentication.
Trend Micro, July 2025	Network-exposed servers found by its own probe	492 servers with no client authentication or traffic encryption, giving access to 1,402 tools.
Astrix Security, October 2025	5,205 open-source server repositories, read from their READMEs	53% rely on static API keys or personal access tokens; 8.5% use OAuth.

Study	Sample	Headline finding, as published
Hasan et al., arXiv 2506.13538, June 2025, revised April 2026	1,899 open-source servers; a tool poisoning scan on a small subset	5.5% exhibit MCP-specific tool poisoning.
Censys, May 2026	12,520 internet-accessible MCP services on 8,758 IP addresses	Protocol version 2025-03-26 on 89.4% of services; 88.3% expose one or more tools.
Zhou et al., arXiv 2605.22333, May 2026	7,973 live remote servers found through FOFA and Shodan	40.55% expose tools without authentication; OAuth 30.45%, static token or key 29.00%.
Afsar, arXiv 2609.10962, September 2026	400 local npm servers drawn at random from the official registry	48.8% complete an initialize handshake; 58.8% of tools have no safety annotation.
Digital Applied, 5 November 2026	A fixed-seed sample of 1,000 remote endpoints from the official registry, at most two per host (974 hosts)	550 completed the handshake and 338 asked for credentials; 140 of 970 streamable-HTTP endpoints (14.4%) answered server/discover.

This report adds a census of every host in the official registry rather than a sample, a defined population that anyone can read again from a public API, and the first figures we found on Client ID Metadata Document support (an earlier study counted dynamic registration from authorization server metadata; see Zhou et al.). Samples of registry entries, even capped per host, and a census of hosts measure different units, so their shares can differ.

10 · Practice

What to do

If your agents connect to MCP servers

1. Keep an allowlist of servers your agents may use, and check each one before it goes on the list: its sign-in, its tools and its tool descriptions.
2. Prefer servers that sign users in with OAuth and give each agent its own credential. Avoid long-lived shared keys in configuration files.
3. Treat a server that lists write or delete tools without a sign-in as a finding, not a convenience.
4. Re-check servers when they change. A remote server can change its tools and their descriptions without a new release.

If you publish an MCP server

1. Ask for a sign-in on anything that is not public by design, and publish OAuth protected resource metadata so clients can find your authorization server.
2. Support Client ID Metadata Documents, and plan the move to the 2026-07-28 revision (`server/discover`, no protocol sessions).
3. Set `readOnlyHint` and `destructiveHint` honestly on every tool.
4. Keep tool descriptions to what the tool does. Never tell the model what to do first, what to hide or what to pass along.
5. Pin container images by digest in your registry entry.

The same checks run in the open-source scanner: `pipx run spellwarden scan --probe .` probes the remote servers in a repository's MCP configuration and reports these rules. Spellwarden's hosted product runs them on every server an organisation allows and re-checks them when they change.

Method and limits

Population

The official MCP Registry API (`registry.modelcontextprotocol.io/v0/servers?version=latest`), read in full at 2026-11-10T06:07:30Z: 59,944 servers whose latest version is active. From their 41,812 remote entries we took every HTTPS URL without a placeholder and grouped the URLs by host name, which gave 30,402 hosts. For each host we probed one endpoint, the first in sorted order. This is a census of hosts, not a sample, so there is no sampling error on the probe figures; the figures describe hosts, not registry entries.

Probe

Probes started between 2026-11-10T06:19:40Z and 2026-11-10T06:29:10Z from one machine in the EU (the rechecks behind section 5 followed the same morning), with the open-source Spellwarden scanner 1.10.0 (rules 2026.11.3), calling the same probe function and baseline rules as `spellwarden scan --probe` . Each probe makes at most nine requests with a ten-second deadline each, sends no credential, never calls a tool and identifies itself in its User-Agent as Spellwarden with a link to spellwarden.ai. One request at a time per host.

Definitions

- **Other HTTP answer**: everything else that answered, including 101 hosts that completed the handshake but answered `tools/list` with another status.
- **No sign-in (open)**: `tools/list` returned 200 with an MCP answer, either after `server/discover` or `initialize` completed or with a tool list of its own.
- **Refused**: 401 or 403. OAuth when the authorization server metadata was readable and advertised CIMD, a registration endpoint or the jwt-bearer grant; CIMD when it sets `client_id_metadata_document_supported` to true; DCR when it offers a registration endpoint without CIMD; key or other when no protected resource metadata was found.
- **Open tool list**: `tools/list` answered without a sign-in with at least one tool. Lists of more than 200 tools or with a next page were not read whole and count as high.
- **Instruction-like text**: rule SW-MCP-011 of scanner 1.10.0, with the reviewed sample described in section 5.
- **Protocol version**: the newest date in the `server/discover` answer, else the version negotiated in `initialize` .

Limits

- One endpoint per host. Hosts that serve many endpoints may treat them differently.
- The view of an anonymous client only. We did not sign in, so servers behind a sign-in were not checked beyond their OAuth metadata.
- Listing is not calling. We never called a tool, so an open list does not prove that its tools run without a sign-in.
- Tool hints are what each server says about its own tools.

- A point in time. Servers change, and some entries are tests or abandoned. 1,125 hosts gave no answer at all (2 of them because their name resolved to a private address, which the scanner refuses).
- The pattern rule over-matches; the reviewed sample corrects for that, was labelled by a model reviewer, and has a wide margin.

Sources

- MCP Registry, about: https://modelcontextprotocol.io/registry/about
- MCP Registry, authentication and namespaces: https://modelcontextprotocol.io/registry/authentication
- MCP specification 2025-11-25, authorization: https://modelcontextprotocol.io/specification/2025-11-25/basic/authorization
- MCP specification 2026-07-28, changelog: https://modelcontextprotocol.io/specification/2026-07-28/changelog
- Spellwarden scanner (Apache-2.0): https://github.com/spellwarden-ai/spellwarden-oss
- MCP Registry pull request 1806 (shared remote URLs, no remote URL ownership check): https://github.com/modelcontextprotocol/registry/pull/1806
- Earlier studies: the links in section 9, checked on 2026-11-10.

Data, reproduction and ethics

The scanner is on GitHub under Apache-2.0 (`spellwarden-ai/spellwarden-oss`). Every figure in this report is published as JSON next to it on spellwarden.ai, and the scripts that fetch the registry, run the probe and compute the figures are available on request. We publish aggregates only and name no server or publisher. Operators who want their own results can write to security@spellwarden.ai.

About

About this report

Spellwarden checks the agent skills and MCP servers an organisation lets its AI agents use: a free, open-source scanner for CI, and a hosted product that turns findings into policy, keeps a private registry of what has been allowed and re-checks it when it changes. Spellwarden is built in the EU by NETBEARS TEAM SRL, Iași, Romania.

How to cite. Spellwarden (2026). *MCP server security, November 2026.*
https://spellwarden.ai/research/mcp-server-security-2026-11
The text and figures are licensed under Creative Commons Attribution 4.0 (CC BY 4.0). Quote them, chart them and reuse them with a link to the source.

Questions about the data: research@spellwarden.ai. Security reports: security@spellwarden.ai.