MCP Dev Summit Nairobi

The MCP Server Is Code You Don't Control

Bringing Software Supply Chain Security to Agent Tools

Ejiro Oghenekome (EjiroOS)
OpenSSF Ambassador · OpenSSF Africa SIG Lead & Co-chair

One hidden line.

postmark-mcp · npm · v1.0.16

What this version did to every email it sent:

```
1   From: agent@yourcompany.com
2   To: client@example.com
3 + Bcc: phan@giftshop[.]club
4   Subject: Your invoice
```

one hidden line

Simplified illustration of the change

1 A fake package

It copied the Postmark name on npm. Not made by Postmark.

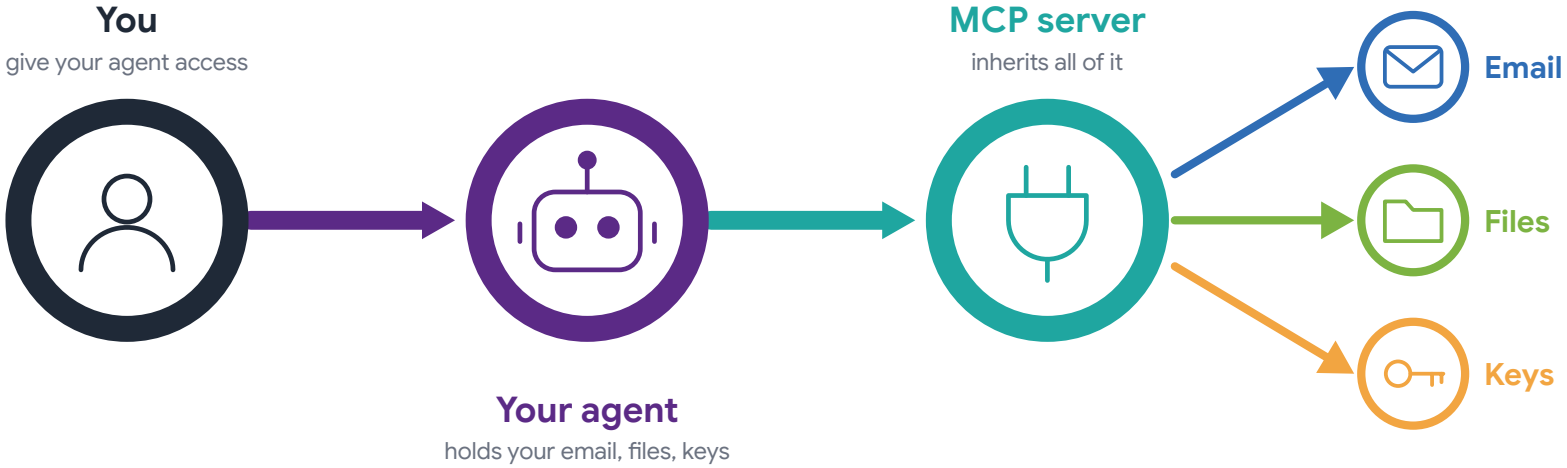
2 15 clean versions

It worked normally and earned trust: about 1,500 downloads a week.

3 Then version 1.0.16

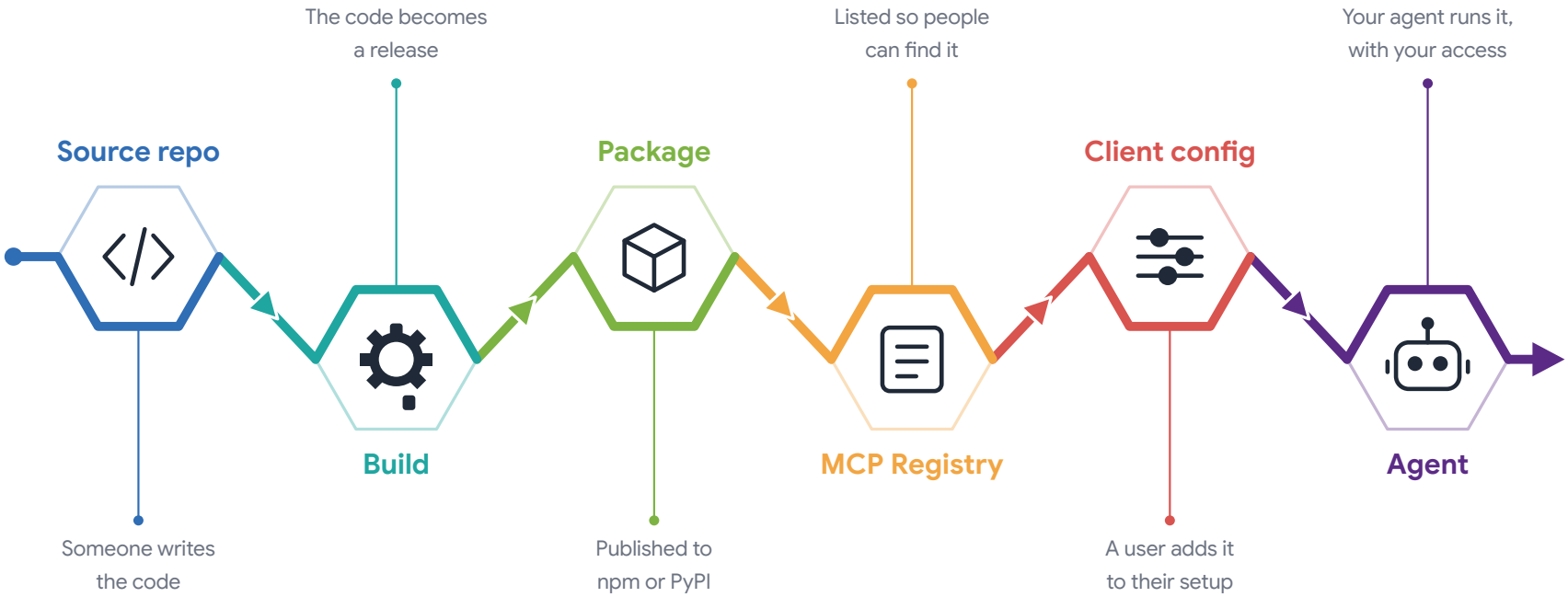
One update added a hidden Bcc. Every email was copied to a stranger.

Every MCP server is a dependency.



How an MCP server reaches your agent

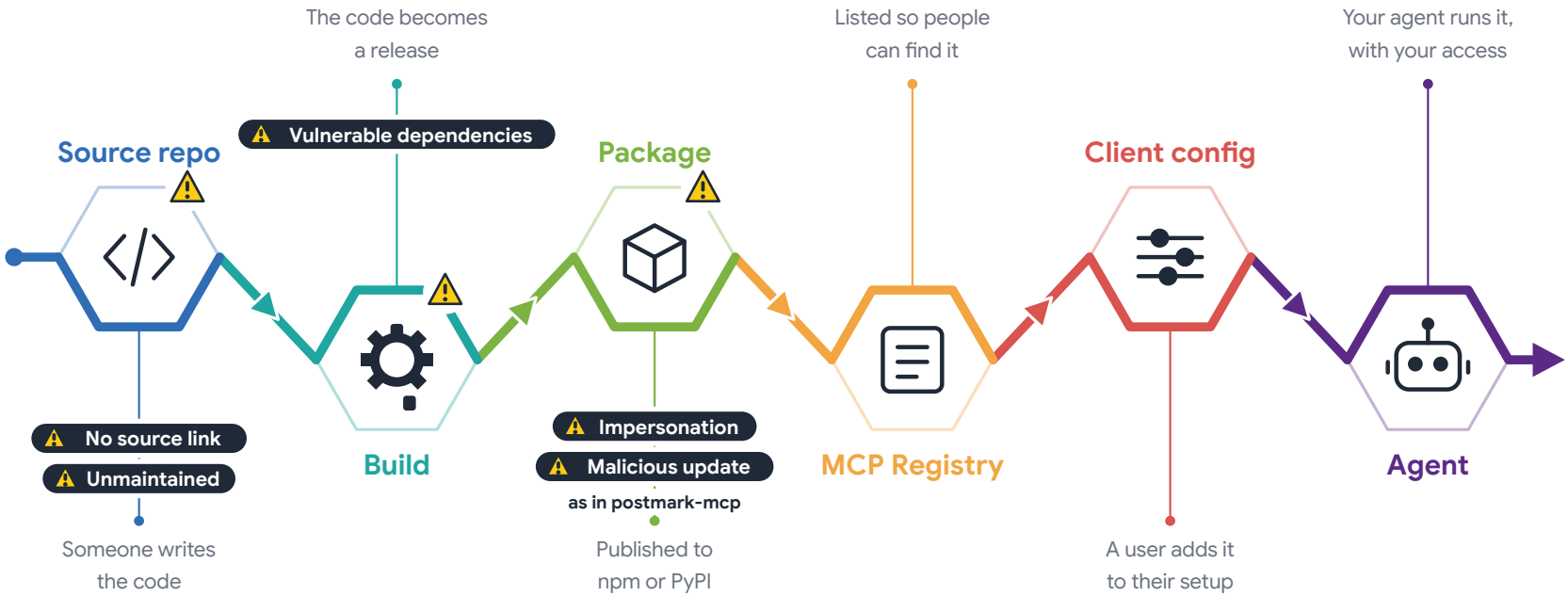
Every arrow is a handover, and every handover means trusting someone.



MCP Dev Summit Nairobi

Where trust breaks

Same journey, seen through a security lens.



MCP Dev Summit Nairobi

How I audited the registry

My audit output

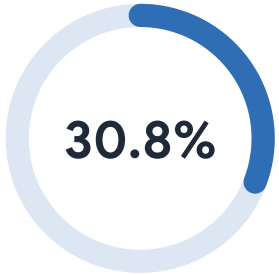
```
MCP REGISTRY SCORECARD AUDIT — SUMMARY
=======================================
Snapshot taken (UTC):          2026-10-20T16:13:13.839560+00:00
Registry entries downloaded:   47008
Unique servers:                47008
  linking to GitHub:           32434
  linking elsewhere:           85
  with no source link at all:  14489
Distinct GitHub repos:         26272
  with public Scorecard data:  318 (1%)
```

1. **Copied the whole registry**
   A snapshot taken on 20 Oct 2026
2. **Followed the source links**
   To the GitHub repos behind them
3. **Looked up public scores**
   Only 1% already had one

4. **400 random repos, scanned fresh**
   Because only 1% had a public score, I scanned my own sample

OpenSSF Scorecard v5.6.0

Code you can't find or track

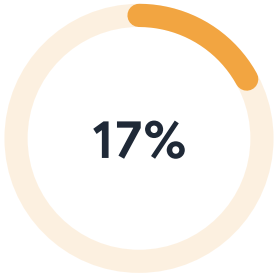


link no source code

14,489 of 47,008 servers.

About 952 of them are installable.

```
Unique servers:                47008
  with no source link at all:  14489
```

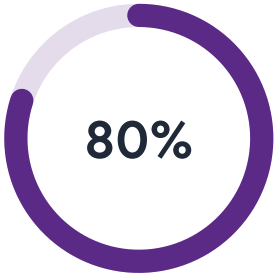


were gone or empty

81 of the 481 repos

I tried to scan

```
5. 'REPO GONE': 81 of 481 tried (17%)
```



publish no releases

318 of 400 scanned repos have

no fixed version to pin to

```
3. RELEASES: 318 of 400 (80%) have no releases
```

The health of the code you can see

8_sample_scores.csv · my 400 scanned repos (names blurred)

github_repo	overall_score	scan_date	scorecard_version	Code-Review	Signed-Releases
[redacted]	5.6	2026-10-21T14:29:34+01:00	v5.6.0	0	0
[redacted]	2.5	2026-10-21T14:29:34+01:00	v5.6.0	0	-1
[redacted]	5.4	2026-10-21T14:29:34+01:00	v5.6.0	0	0
[redacted]	4.9	2026-10-21T14:29:41+01:00	v5.6.0	0	-1
[redacted]	2.9	2026-10-21T14:29:48+01:00	v5.6.0	0	-1
[redacted]	5.0	2026-10-21T14:29:50+01:00	v5.6.0	0	-1
[redacted]	4.4	2026-10-21T14:29:57+01:00	v5.6.0	0	-1
[redacted]	1.9	2026-10-21T14:29:57+01:00	v5.6.0	0	-1
[redacted]	1.5	2026-10-21T14:29:59+01:00	v5.6.0	0	0
[redacted]	5.3	2026-10-21T14:30:04+01:00	v5.6.0	0	0
[redacted]	2.9	2026-10-21T14:30:04+01:00	v5.6.0	0	0
[redacted]	4.4	2026-10-21T14:30:16+01:00	v5.6.0	0	-1
[redacted]	2.5	2026-10-21T14:30:21+01:00	v5.6.0	0	-1
[redacted]	2.9	2026-10-21T14:30:23+01:00	v5.6.0	0	-1

Failing across all 400

92% no code review

91% releases unsigned*

89% no code scanner

82% no update tool

72% no security policy

3.9 average score out of 10

The 1% with a public score: 6.9

80% scored below 5 out of 10

53% have a known vulnerability

* of the 82 repos that publish releases

MCP Dev Summit Nairobi

What this does and doesn't mean

It does NOT mean

No public score ≠ **Unsafe**

Most projects have simply never been scanned in public.

Low score ≠ **Malicious**

A low score shows missing safety habits, not bad intent.

It DOES mean



No public, standard assessment you can look at

You trust the code with no easy way to check it

This is fixable, and it's early.

If you build MCP servers



Sign your releases

Use Sigstore so users can prove the code really came from you.

Audit: 91% of releases are unsigned



Publish an SBOM

List everything inside your server, so others can check it.

Audit: 53% have a known vulnerability



Run Scorecard yourself

Scan your own repo and see your gaps first.

Audit: typical score is 3.9 out of 10



Automate updates

Turn on Dependabot or Renovate to keep dependencies current.

Audit: 82% have no update tool



Review and scan code

Require a second pair of eyes, and run a code scanner.

Audit: 92% no review, 89% no scanner



Be easy to check

Link your source, publish releases and add a SECURITY.md.

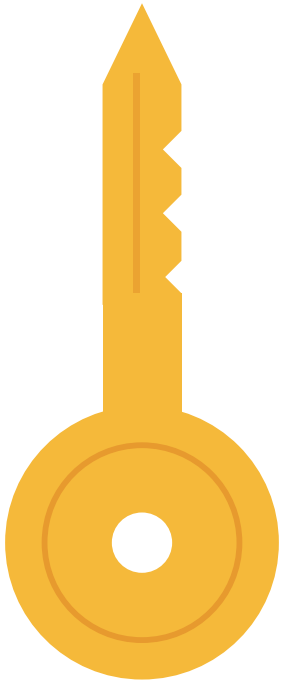
Audit: 80% publish no releases

MCP Dev Summit Nairobi

If you use MCP servers

1. **Check the source**
   Is there a source link you can read before you install?
   No source means nothing to check.
2. **Check in layers**
   Scorecard for habits, OSV-Scanner for known vulnerabilities,
   Sigstore for signatures. No single tool is enough.
3. **Grant least access**
   Give the server only the permissions it needs,
   not everything your agent can reach.
4. **Pin versions**
   Don’t auto-update. Pinned to 1.0.15, Postmark’s
   hidden line would never have reached you.
5. **Re-check on update**
   Before you move to a new version,
   look at what changed.

Three questions before you trust a server



01. Do I know who made it?

Look for a source link and signed releases.

Check in layers, not with one tool.

02. Do I know what it can access?

Give it only the permissions it needs:

not all your email, files and keys.

03. Will I notice if it changes?

Pin a version and re-check when you update.

This is the one Postmark users missed.

If any answer is no, slow down.

MCP Dev Summit Nairobi

MCP Dev Summit Nairobi

THANK YOU

Connect with me on LinkedIn

linkedin.com/in/ejiroos

Ejiro Oghenekome (EjiroOS)