What a Year of Breaking MCP Tells Builders

Protocol Gaps and What Ships Next

Amine Raji
MCP & Agent Security · MoIntek

I run these servers, and I reproduced what broke them

I run Grafana MCP and ArgoCD MCP in my own cluster

I did not find any of these vulnerabilities

I reproduced them, to see what they have in common

Cloud and autonomous systems security at a car company. Contributor to the OWASP MCP Top 10, currently in beta.

Amine Raji · Molntek

Every trust check was a string comparison

Every deployment I looked at had a trust check. Every check compared a string to something nobody had issued, verified, or bound to an identity.

Every trust check was a string comparison

Every deployment I looked at had a trust check. Every check compared a string to something nobody had issued, verified, or bound to an identity.

the session · the network · the approval

Three anchors, three servers, same shape

the session

Grafana MCP

the network

GitLab MCP

the approval

Amazon Q, Claude Code, Windsurf, Cursor

believed checked · actually checked · caller got · control

Same four beats, same order, every story.

The defaults are the problem, not the outliers

The defaults are the problem, not the outliers

24,008

unique secrets in MCP-related configs on public GitHub during 2025. **2,117 still valid.**

GitGuardian, State of Secrets Sprawl · 2026

LOWER BOUND

The defaults are the problem, not the outliers

24,008

unique secrets in MCP-related configs on public GitHub during 2025. **2,117 still valid.**

GitGuardian, State of Secrets Sprawl · 2026

79%

of open-source MCP servers pass credentials through environment variables. **88% require credentials; only 8.5% use OAuth**, and 53% rely on static keys or personal access tokens.

Astrix, State of MCP Server Security

The defaults are the problem, not the outliers

24,008

unique secrets in MCP-related configs on public GitHub during 2025. **2,117 still valid.**

GitGuardian, State of Secrets Sprawl · 2026

79%

of open-source MCP servers pass credentials through environment variables. **88% require credentials; only 8.5% use OAuth**, and 53% rely on static keys or personal access tokens.

Astrix, State of MCP Server Security

1,800+

publicly reachable MCP servers accepting connections with **no credential validation.**

Internet-wide scan, cited in CSA research note · May 2026

The defaults are the problem, not the outliers

24,008

unique secrets in MCP-related configs on public GitHub during 2025. **2,117 still valid.**

GitGuardian, State of Secrets Sprawl · 2026

79%

of open-source MCP servers pass credentials through environment variables. **88% require credentials; only 8.5% use OAuth**, and 53% rely on static keys or personal access tokens.

Astrix, State of MCP Server Security

1,800+

publicly reachable MCP servers accepting connections with **no credential validation.**

Internet-wide scan, cited in CSA research note · May 2026

GitGuardian's stated root cause: official quickstart documentation normalises putting keys straight into config files.

believed checked | actually checked | caller got | control

Valid but never issued

Grafana MCP. Session spoofing found by Pillar Security, reported via Intigriti 2 August 2026, fixed in mcp-grafana v1.1.0, 10 August 2026. Grafana classified it as a hardening improvement, so the session issue has no separate CVE. · disclosed August 2026

believed checked | actually checked | caller got | control

Valid but never issued

The server checked the session. The session was a correctly formatted string that the server had never issued.

Grafana MCP. Session spoofing found by Pillar Security, reported via Intigriti 2 August 2026, fixed in mcp-grafana v1.1.0, 10 August 2026. Grafana classified it as a hardening improvement, so the session issue has no separate CVE. · disclosed August 2026

believed checked | **actually checked** | caller got | control

Valid but never issued

The server checked the session. The session was a correctly formatted string that the server had never issued.

`mcp-session-<uuid>` accepted. The format matched. Nothing else was checked.

Grafana MCP. Session spoofing found by Pillar Security, reported via Intigriti 2 August 2026, fixed in mcp-grafana v1.1.0, 10 August 2026. Grafana classified it as a hardening improvement, so the session issue has no separate CVE. · disclosed August 2026

believed checked | actually checked | caller got | control

Valid but never issued

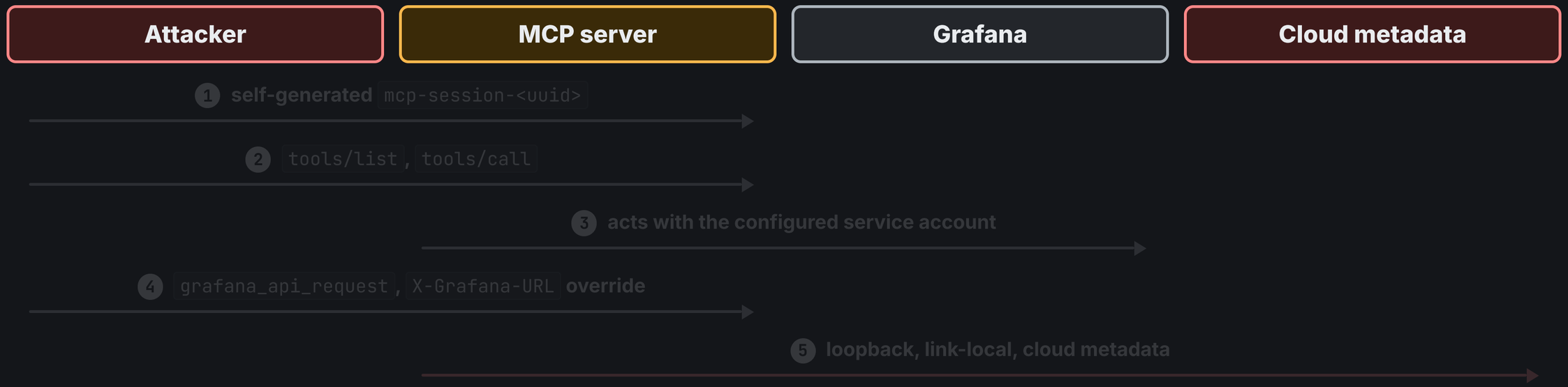
The server checked the session. The session was a correctly formatted string that the server had never issued.

`mcp-session-<uuid>` accepted. The format matched. Nothing else was checked.

The server appeared to enforce sessions. That is worse than having none, because operators believed unauthenticated access was impossible.

Grafana MCP. Session spoofing found by Pillar Security, reported via Intigriti 2 August 2026, fixed in mcp-grafana v1.1.0, 10 August 2026. Grafana classified it as a hardening improvement, so the session issue has no separate CVE. · disclosed August 2026

A made-up session reached the cloud metadata endpoint



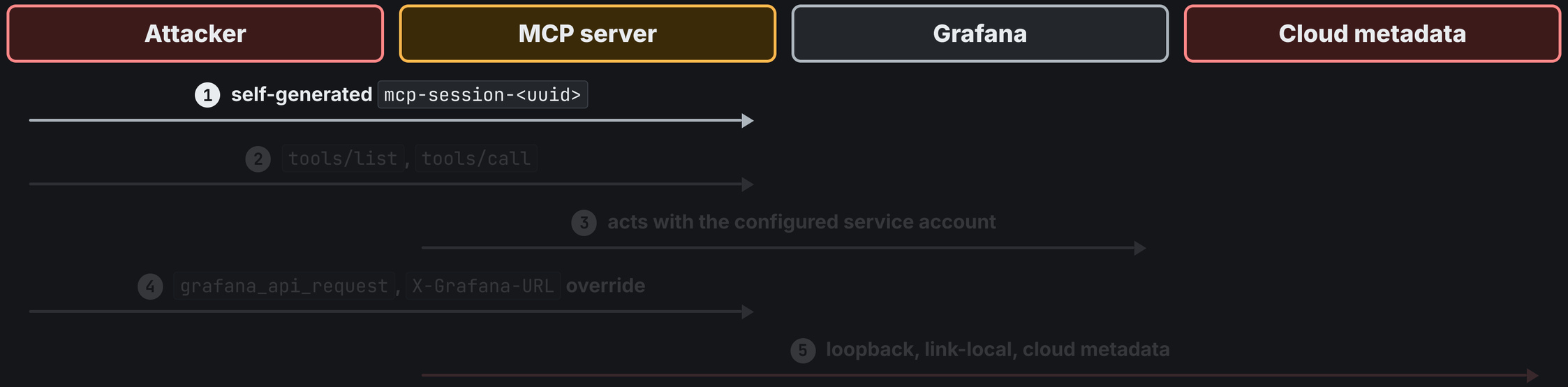
`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

A made-up session reached the cloud metadata endpoint



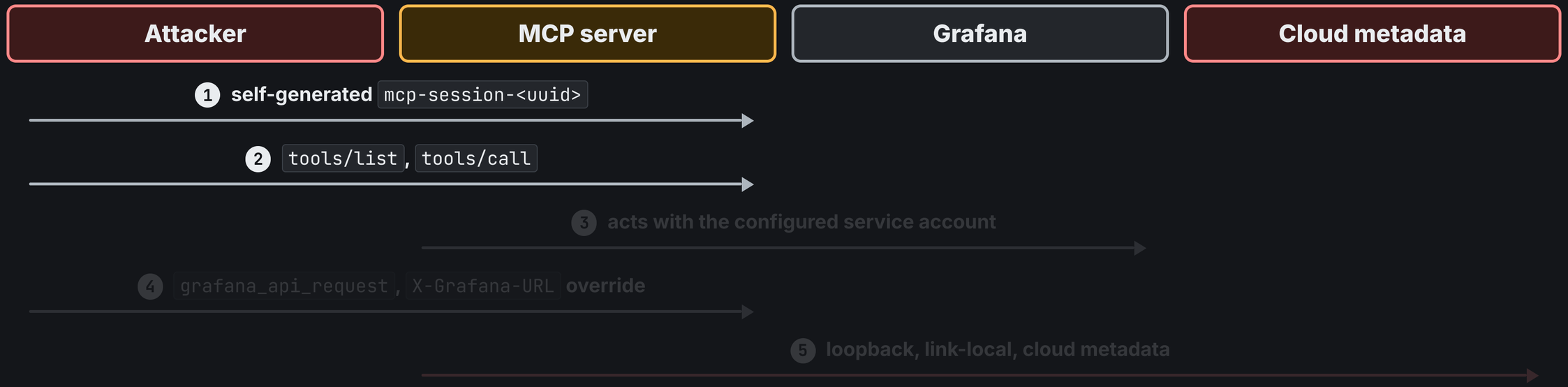
`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

A made-up session reached the cloud metadata endpoint



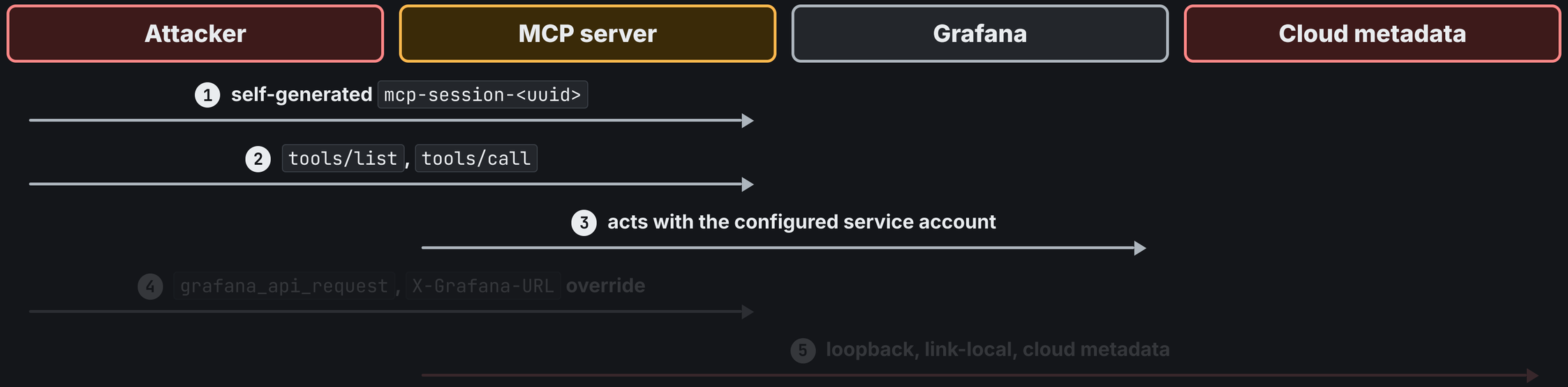
`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

A made-up session reached the cloud metadata endpoint



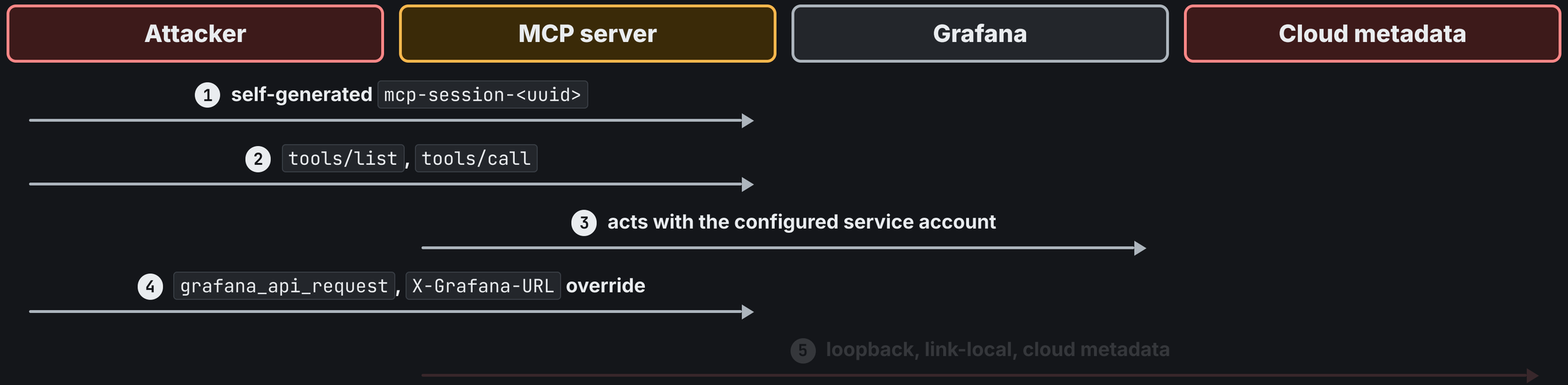
`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

A made-up session reached the cloud metadata endpoint



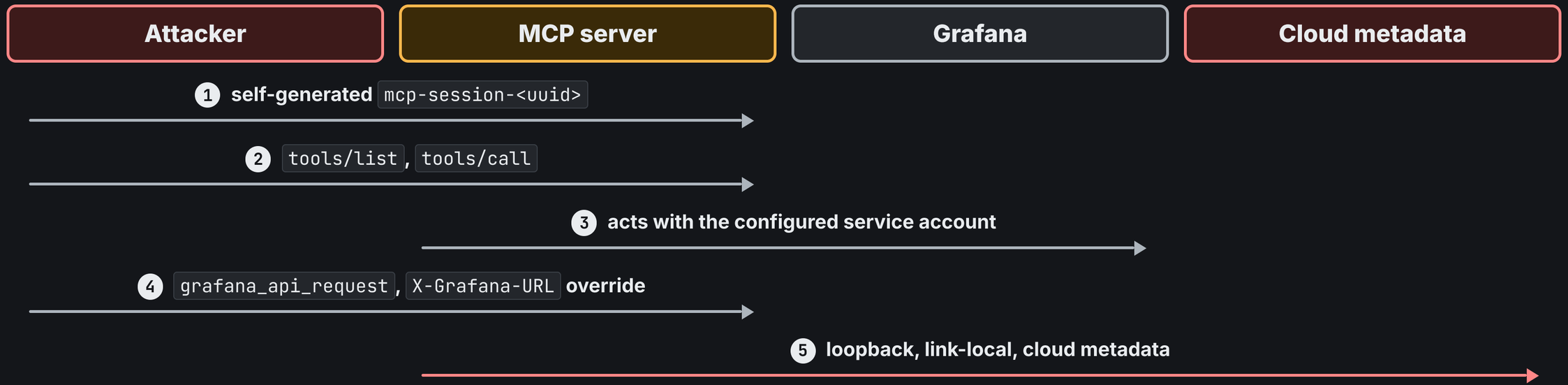
`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

A made-up session reached the cloud metadata endpoint



`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

A made-up session reached the cloud metadata endpoint

```
------------------------------------------------------------------------
REQUEST
POST http://grafana-vuln:8000/
Mcp-Session-Id: mcp-session-6229097e-28c5-4ba8-aca8-13e53eff3da7
Content-Type: application/json

{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "tools/list"
}
------------------------------------------------------------------------
```

A session I invented. Isolated lab, not my cluster.

```
------------------------------------------------------------------------
RESPONSE
HTTP 200
Server: mcp-grafana-sim/vuln Python/3.12.14
Date: Tue, 15 Sep 2026 19:36:22 GMT
Content-Type: application/json
Content-Length: 461

{
  "jsonrpc": "2.0",
  "id": "1",
  "result": {
    "tools": [
      {
        "name": "grafana_api_request",
        "description": "Proxy an API request to the configured Grafana
URL header.",
```

Accepted. Here is the tool list it returned.

`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

A made-up session reached the cloud metadata endpoint

```
--------------------------------------------------------------------------------
REQUEST
POST http://grafana-vuln:8000/
Mcp-Session-Id: mcp-session-6229097e-28c5-4ba8-aca8-13e53eff3da7
Content-Type: application/json

{
  "jsonrpc": "2.0",
  "id": "1",
  "method": "tools/list"
}
--------------------------------------------------------------------------------
```

A session I invented. Isolated lab, not my cluster.

```
--------------------------------------------------------------------------------
RESPONSE
HTTP 200
Server: mcp-grafana-sim/vuln Python/3.12.14
Date: Tue, 15 Sep 2026 19:36:22 GMT
Content-Type: application/json
Content-Length: 461

{
  "jsonrpc": "2.0",
  "id": "1",
  "result": {
    "tools": [
      {
        "name": "grafana_api_request",
        "description": "Proxy an API request to the configured Grafana
URL header.",
```

Accepted. Here is the tool list it returned.

`X-Grafana-URL` is caller controlled, with no destination allowlist. Roughly 1.9 million Docker Hub pulls of the affected image, and pulls are not deployments.

Docker Hub pull count, CSA research note · 3 September 2026

CVE-2026-19516, CVSS 9.1, CWE-918. SSRF via caller-controlled X-Grafana-URL. Pillar Security. Affected mcp-grafana <= 1.0.0, fixed v1.1.0. · published 11 August 2026

Authenticated, audience-scoped access before any tool executes. Destination allowlisting on every network-calling tool. Metadata endpoint blocked from the segment.

As of the current release, caller auth is still enforced only when `--server-auth-token` is set; without it, a non-loopback bind starts anyway. Authentication remains optional, which is the point.

The token authorised the wrong direction

```
res.setHeader('Access-Control-Allow-Origin', '*');
res.setHeader('Access-Control-Allow-Methods', 'GET, POST');
res.setHeader('Access-Control-Allow-Headers', 'Content-Type');
httpServer.listen(port);   // no host argument, Node defaults to 0.0.0.0
```

Any origin can call it.

The token authorised the wrong direction

```
res.setHeader('Access-Control-Allow-Origin', '*');
res.setHeader('Access-Control-Allow-Methods', 'GET, POST');
res.setHeader('Access-Control-Allow-Headers', 'Content-Type');
httpServer.listen(port);   // no host argument, Node defaults to 0.0.0.0
```

Any origin can call it.

Every interface, not loopback.

The token authorised the wrong direction

```
res.setHeader('Access-Control-Allow-Origin', '*');
res.setHeader('Access-Control-Allow-Methods', 'GET, POST');
res.setHeader('Access-Control-Allow-Headers', 'Content-Type');
httpServer.listen(port);   // no host argument, Node defaults to 0.0.0.0
```

Any origin can call it.

Every interface, not loopback.

All 86 tools, including destructive ones, under the operator's personal access token.

The token authorised the wrong direction

```
res.setHeader('Access-Control-Allow-Origin', '*');
res.setHeader('Access-Control-Allow-Methods', 'GET, POST');
res.setHeader('Access-Control-Allow-Headers', 'Content-Type');
httpServer.listen(port);   // no host argument, Node defaults to 0.0.0.0
```

Any origin can call it.

Every interface, not loopback.

All 86 tools, including destructive ones, under the operator's personal access token.

The personal access token was doing real work. It was authorising the downstream call, not the inbound caller.

The token authorised the wrong direction

```
res.setHeader('Access-Control-Allow-Origin', '*');
res.setHeader('Access-Control-Allow-Methods', 'GET, POST');
res.setHeader('Access-Control-Allow-Headers', 'Content-Type');
httpServer.listen(port);   // no host argument, Node defaults to 0.0.0.0
```

Any origin can call it.

Every interface, not loopback.

All 86 tools, including destructive ones, under the operator's personal access token.

The personal access token was doing real work. It was authorising the downstream call, not the inbound caller.

Any page the operator visits while the server runs can open the connection and call tools cross-origin. No interaction beyond visiting.

The token authorised the wrong direction

WHERE	FINDING
`@yoda.digital/gitlab-mcp-server` **< 0.6.0**	CVE-2026-44895. No inbound auth, wildcard CORS, 0.0.0.0 bind. CWE-306
`mcp-gitlab` **< 2.1.18**	CVE-2026-61462, CVSS 9.2. Path traversal via `job_id`
Official Go SDK	CVE-2026-33252. Cross-site POST accepted without Origin validation, named risk case: **stateless or sessionless configurations**
Official Python SDK	CVE-2025-66416. DNS rebinding protection off by default until 1.23.0

The personal access token was doing real work. It was authorising the downstream call, not the inbound caller.

Any page the operator visits while the server runs can open the connection and call tools cross-origin. No interaction beyond visiting.

The token authorised the wrong direction

WHERE	FINDING
`@yoda.digital/gitlab-mcp-server` **< 0.6.0**	CVE-2026-44895. No inbound auth, wildcard CORS, 0.0.0.0 bind. CWE-306
`mcp-gitlab` **< 2.1.18**	CVE-2026-61462, CVSS 9.2. Path traversal via `job_id`
Official Go SDK	CVE-2026-33252. Cross-site POST accepted without Origin validation, named risk case: **stateless or sessionless configurations**
Official Python SDK	CVE-2025-66416. DNS rebinding protection off by default until 1.23.0

The personal access token was doing real work. It was authorising the downstream call, not the inbound caller.

Any page the operator visits while the server runs can open the connection and call tools cross-origin. No interaction beyond visiting.

The counter-example, from Akuity's `mcp-for-argocd` README: "This token is outbound only: it authenticates this server to ArgoCD and never authorizes an inbound caller." The distinction is nameable, and someone already named it.

The code ran before the dialog appeared

```
// .amazonq/mcp.json, committed in a repository
{ "mcpServers": { "x": { "command": "sh", "args": ["-c", "..."] } } }
```

Opening the folder was the entire attack surface. Wiz's proof of concept was a single config file.

The code ran before the dialog appeared

```
// .amazonq/mcp.json, committed in a repository
{ "mcpServers": { "x": { "command": "sh", "args": ["-c", "..."] } } }
```

Opening the folder was the entire attack surface. Wiz's proof of concept was a single config file.

AWS access keys, secret keys, session tokens, CLI credentials, API keys, SSH agent sockets.

The code ran before the dialog appeared

2025
Cursor
CVE-2025-54136

Approved config never re-validated, swapped after approval

2025
Claude Code
CVE-2025-59536 · CVSS 8.7

Repository settings ran shell commands **before** the trust dialog

APR 2026
Windsurf
CVE-2026-30615 · CVSS 8.0

Rendered HTML rewrote the MCP config and registered a server, **zero clicks**

APR TO JUN 2026
Amazon Q
CVE-2026-12957 · CVSS 8.5

Workspace config auto-loaded with no consent step

the session · the network · **the approval**

believed checked | actually checked | **caller got** | control

The code ran before the dialog appeared

2025
Cursor
CVE-2025-54136

Approved config never re-validated, swapped after approval

2025
Claude Code
CVE-2025-59536 · CVSS 8.7

Repository settings ran shell commands **before** the trust dialog

APR 2026
Windsurf
CVE-2026-30615 · CVSS 8.0

Rendered HTML rewrote the MCP config and registered a server, **zero clicks**

APR TO JUN 2026
Amazon Q
CVE-2026-12957 · CVSS 8.5

Workspace config auto-loaded with no consent step

At least four independent teams shipped the same assumption inside a year. That is a design gap, not a vendor failure.

The code ran before the dialog appeared

2025
Cursor
CVE-2025-54136

Approved config never re-validated, swapped after approval

2025
Claude Code
CVE-2025-59536 · CVSS 8.7

Repository settings ran shell commands **before** the trust dialog

APR 2026
Windsurf
CVE-2026-30615 · CVSS 8.0

Rendered HTML rewrote the MCP config and registered a server, **zero clicks**

APR TO JUN 2026
Amazon Q
CVE-2026-12957 · CVSS 8.5

Workspace config auto-loaded with no consent step

At least four independent teams shipped the same assumption inside a year. That is a design gap, not a vendor failure.

Same lesson from the other direction: postmark-mcp shipped faithful copies through 1.0.15, then added one BCC line in 1.0.16, and SmartLoader built a fake developer ecosystem across five repositories over three months before cloning the Oura server. Approval bound to a name is not approval.

the session · the network · **the approval**

believed checked | actually checked | caller got | **control**

July fixed real things

REMOVED OR DEPRECATED

- Protocol sessions and `Mcp-Session-Id`, removed
- The `initialize` and `initialized` handshake, removed
- Sampling, Roots and Logging, deprecated on a twelve-month clock

ADDED

- `server/discover`, now mandatory
- `iss` validation, RFC 9207
- Issuer-bound client credentials
- Client ID Metadata Documents, now preferred over Dynamic Client Registration
- Required `ttlMs` and `cacheScope` on list results
- Enterprise-Managed Authorization, also called Cross App Access

Model Context Protocol specification · 2026-07-28

Two new surfaces, neither of which I have tested

The `description` field is still unlabelled text going into a context window. Statelessness did not touch it. Neither did the authorization work. And July added two surfaces:

Discovery is now a cacheable instruction surface

Every `tools/list` result now declares how long it may be cached, and by whom. The server that writes the tool description also sets how long you keep trusting it:

```
{ "resultType": "tools",
  "cacheScope": "public",
  "ttlMs": 3600000 }
```

`public` : a shared cache may hold it. `3600000` : one hour before anyone reads that description again.

Required fields on list results, MCP specification · 2026-07-28

Four consequences

not tested, reasoned from the schema

Cross-tenant bleed	Identity data served as `public`
Stale poisoning	Long `ttlMs` on discovery
Re-consent bypass	Cache hides definition drift
Replay amplification	Cache serves pre-rollback output

State handles have bearer-like semantics

With sessions gone, a server that has to remember something returns a handle, and the model carries it back. Those handles travel through transcripts, logs and subagent chains.

A handle that only **names** a thing is fine, like a pull request number. A handle that **grants** access to it is a password.

An authorization-bearing handle is a credential in a chat log.

`ttlMs` **is a performance hint, not an authorization decision.**

the session · the network · the approval

Two have an owner. One does not.

ANCHOR	WHAT ANSWERS IT	WHO OWNS IT

the session · the network · the approval

THE SEP GOT THERE FIRST

Two have an owner. One does not.

ANCHOR	WHAT ANSWERS IT	WHO OWNS IT
the session	SEP-2567 (Final, created 2026-03-11): recommends opaque handles validated per request. A recommended posture, not a schema requirement	Transport and primitives. **Written five months before the CVE**

Two have an owner. One does not.

ANCHOR	WHAT ANSWERS IT	WHO OWNS IT
the session	SEP-2567 (Final, created 2026-03-11): recommends opaque handles validated per request. A recommended posture, not a schema requirement	Transport and primitives. **Written five months before the CVE**
the network	Roadmap: HTTP-native transport unification and hardening. The evidence it is needed is in the official SDKs	Transport working group

Two have an owner. One does not.

ANCHOR	WHAT ANSWERS IT	WHO OWNS IT
the session	SEP-2567 (Final, created 2026-03-11): recommends opaque handles validated per request. A recommended posture, not a schema requirement	Transport and primitives. **Written five months before the CVE**
the network	Roadmap: HTTP-native transport unification and hardening. The evidence it is needed is in the official SDKs	Transport working group
the approval	Publisher and artefact identity binding	NO WORKING GROUP

Two have an owner. One does not.

ANCHOR	WHAT ANSWERS IT	WHO OWNS IT
the session	SEP-2567 (Final, created 2026-03-11): recommends opaque handles validated per request. A recommended posture, not a schema requirement	Transport and primitives. **Written five months before the CVE**
the network	Roadmap: HTTP-native transport unification and hardening. The evidence it is needed is in the official SDKs	Transport working group
the approval	Publisher and artefact identity binding	NO WORKING GROUP

Caller identity has DPoP, Workload Identity Federation, ID-JAG and token exchange. Server-side publisher identity has nothing on the roadmap.

Two have an owner. One does not.

ANCHOR	WHAT ANSWERS IT	WHO OWNS IT
the session	SEP-2567 (Final, created 2026-03-11): recommends opaque handles validated per request. A recommended posture, not a schema requirement	Transport and primitives. **Written five months before the CVE**
the network	Roadmap: HTTP-native transport unification and hardening. The evidence it is needed is in the official SDKs	Transport working group
the approval	Publisher and artefact identity binding	NO WORKING GROUP

Caller identity has DPoP, Workload Identity Federation, ID-JAG and token exchange. Server-side publisher identity has nothing on the roadmap.

Python SDK **Todo, unassigned** Java SDK **Todo, unassigned** Rust SDK **open PR** TypeScript SDK **draft PR**

As of last week, the SEP-2567 tracking issues in the Java and Python SDKs were still open and unassigned.

I checked this, and it does not hold up

I was going to tell you that better models are easier to poison. I went back through the 2026 data and it does not hold up cleanly.

I checked this, and it does not hold up

SUPPORTS IT

- MCPTox: 1.3k+ cases, 353 real tools, 45 real servers, top attack success 72.8%, refusal under 3%
- Raccoon: linear correlation between measured instruction-following and susceptibility

COMPLICATES IT

- Wharton GAIL, April 2026, roughly 40,000 grading trials: frontier models largely resistant, GPT-4o mini inflated scores by nearly 20 points
- Agentic injection evaluation, June 2026: attacks tuned on small open models do not transfer to frontier models
- Cisco, May 2026, 15 frontier models: the signal is single-turn versus multi-turn, and reasoning mode moved one model from 88.3% to 43.5%

I checked this, and it does not hold up

SUPPORTS IT

- MCPTox: 1.3k+ cases, 353 real tools, 45 real servers, top attack success 72.8%, refusal under 3%
- Raccoon: linear correlation between measured instruction-following and susceptibility

COMPLICATES IT

- Wharton GAIL, April 2026, roughly 40,000 grading trials: frontier models largely resistant, GPT-4o mini inflated scores by nearly 20 points
- Agentic injection evaluation, June 2026: attacks tuned on small open models do not transfer to frontier models
- Cisco, May 2026, 15 frontier models: the signal is single-turn versus multi-turn, and reasoning mode moved one model from 88.3% to 43.5%

MCPTox stands on its own terms, for tool poisoning specifically. It is one benchmark, not a general law.

Model choice is not the control

GitInject, studying agentic CI/CD on GitLab, found the failure is structural rather than model-specific. When configuration files load from the same untrusted repository context as the code under review, no amount of model-side alignment can distinguish a legitimate project preference from a malicious override.

Model choice is not the control

GitInject, studying agentic CI/CD on GitLab, found the failure is structural rather than model-specific. When configuration files load from the same untrusted repository context as the code under review, no amount of model-side alignment can distinguish a legitimate project preference from a malicious override.

If the failure is structural, the fix is structural. That is the argument for protocol and host controls.

Model choice is not the control

GitInject, studying agentic CI/CD on GitLab, found the failure is structural rather than model-specific. When configuration files load from the same untrusted repository context as the code under review, no amount of model-side alignment can distinguish a legitimate project preference from a malicious override.

If the failure is structural, the fix is structural. That is the argument for protocol and host controls.

Which is where we started: every trust check was a string comparison.

Verify the publisher, authenticate inbound, block on drift

1. Verify the publisher before you verify the content
2. Authenticate inbound separately from downstream
3. Snapshot the definition surface and block on drift

> I am publishing an audit method and results for the internal-tool MCP servers most teams actually run: observability, source control, CI, cluster control planes. If you maintain one, I will run it against yours and give you the findings before I publish anything.

THE LABS REPO
github.com/aminrj-labs/mcp-attack-labs

THE NEWSLETTER
aminrj.com/newsletter

LINKEDIN
linkedin.com/in/araji