Your MCP Server Is an Attacker's Dream

A Security Playbook From Real-World Assessments

Akash Mahajan

2x Decades of Hacking & Securing Apps, Cloud, APIs and now AI, Agents & MCPs

2x Security Books Author (Bestselling books on Burp Suite & Ansible)

2x Company Builder Founder CEO (Kloudle & Appsecco)

2x Community Builder (null Community & Headstart)

2x Father (Boys 7 & 3)

Love Tea and Coffee as long as its without milk

But why should you pay attention to me?

The NSA is citing our work in MCP security!

Basically, the highest authority in security recognizes the impact of our research. 🤷‍♂️😀

for-ai-thats-bringing-fierce-rivals-together/

[3] Australian Signals Directorate's Australian Cyber Security Centre (ASD's ACSC). Careful adoption of agentic AI services. 2026. https://www.cyber.gov.au/business-government/secure-design/artificial-intelligence/careful-adoption-of-agentic-ai-services

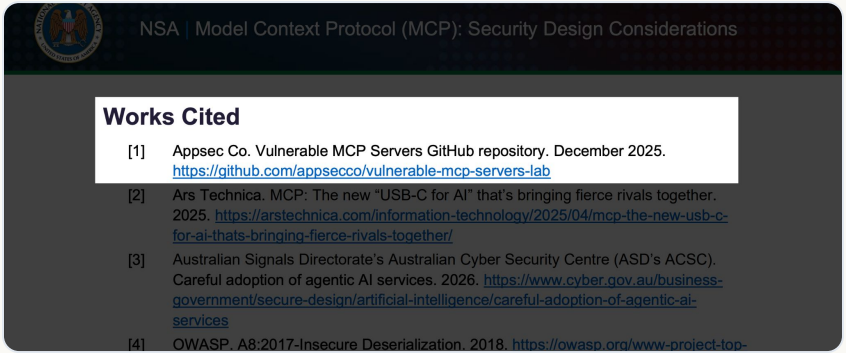
[4] OWASP. A8:2017-Insecure Deserialization. 2018. https://owasp.org/www-project-top-ten/2017/A8_2017-Insecure_Deserialization

[5] Model Context Protocol. Model Context Protocol Specification. 2025. https://modelcontextprotocol.io/specification/

[6] O. Santos. AI Model Context Protocol (MCP) and Security. 2025. https://community.cisco.com/t5/security-blogs/ai-model-context-protocol-mcp-and-security/ba-p/5274394

[7] Model Context Protocol. Authorization. 2025. https://modelcontextprotocol.io/specification/2025-11-25/basic/authorization

NSA Model Context Protocol (MCP) Security Design Considerations



So I Built A Mcp Pentesting Agent, But Why?

Nothing out there could tell me if my customers' SaaS/AI/ Infra was secure if they hooked it up with a MCP server.

So, I had to fix that!

It worked so well that I thought, how does it fare against the SOTA for prod MCP servers.

15+

MCP Servers with **Critical** Authorization Bugs

Shipping Fast, Skipping Security, YOLO 🤘

01

Missing object level authorization (IDOR)

02

Unauthenticated callers read the whole catalog

03

RBAC present but ignored

The Lethal Trifecta As Coined By Simon Willison (2025)

01

Access to private data

Over-privileged tools expose data server shouldn't have

02

Exposure to evil content

Injected content the LLM treats as trusted commands

03

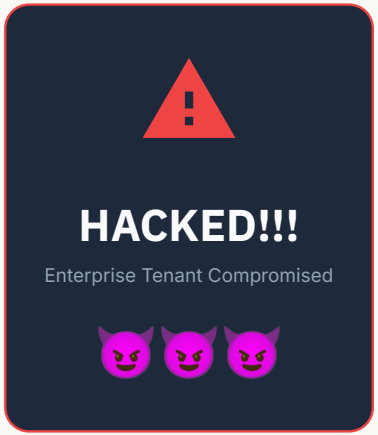
Ability to exfiltrate

SSRF via tool args & transport misconfigs copy it outside

Survivable alone. Lethal in combination.

Trifecta Shows Up in Prod MCP Servers

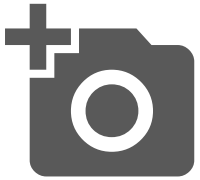
1. **Broken RBAC** Tools list returned 70+ tools
2. **Data Leakage** Credential-list tool gave up metadata on 100+ creds
3. **Unrestricted Egress** HTTP tool lacked an outbound allowlist.
4. **SSRF Exploitation** Attacker server URL + Cred ref used to resolve secrets server-side
5. **Automated Exfil** Looped all IDs to leak the entire enterprise tenant
6. **LEAKED ASSETS** GitHub tokens, AWS keys, Tavily tokens, and more.



Checklist To Assess Any Server Before Launching In Production

My recommendation

Take a photo of the next slide and give it to your coding agent and ask it to build a rubric and audit your MCP server.



7 Steps Security Review for your MCP Server ✅

01. MAP THE SURFACE
Inventory tools, schemas, and annotations.
Test unauthenticated first: what leaks before login?

02. CLASSIFY POWER
Read-only vs. State-changing vs. Destructive.
Flag URL, path, code, and identity fields.

03. ENUMERATE SURFACES
Analyze tool descriptions and return data.
Identify potential prompt injection points.

04. CONFUSED-DEPUTY GRAPH
Map every read → write pair.
SSRF-source to credential-sink = High Risk.

05. TEST AUTHORIZATION
Test with ≥2 principals (cross-user/role).
Use positive/negative controls to catch IDOR.

06. ACTIVE-SAFE PROBING
SSRF callbacks & credential exfiltration.
Operator-owned resources only with cleanup.

07. PROMOTE EVIDENCE → ISSUES
Back claims with request/response logs. Verify against raw schemas.

Secure MCP Servers Hall of Fame - Based on Real Audits

✔ Got it right

Pattern: real authz boundaries, SSRF allowlists, no pre-auth leak.

GitHub

Cross-org access denied; org-scoped fine-grained PATs.

Supabase

Bidirectional cross-principal access denied; boundary enforced.

Linear

Read/destructive hints on all 46 tools; SSRF domain allowlist.

MotherDuck

No promotable vulns; destructive hint set on the SQL tool.

PostHog: Required auth; unauth assessment blocked at transport.

Summary Takeaways

01

Apply the Lethal Trifecta Lens

Any MCP server with private data, untrusted input, and an outbound path is one prompt away from compromise.

02

Common Mistakes are predictable

Tool poisoning, overprivileged definitions, SSRF via arguments, and transport misconfig show up again and again.

03

Code Review Won’t Catch tit

Tool metadata reads like docs, not code. You need a rigorous assessment, not just a skim.

Thank You!

I hope now you won't let your MCP server become an attacker's dream

Follow My Work

https://linktr.ee/makash

Slides are here 👆