MCP Server Security Review

Five checks before an MCP server reaches production

blog.andreladem ann.de

> Installing an MCP server gives an AI system – and whoever controls it – a tool that can act in your name. This is the middle ground between **ship it** and a full penetration test.

REVIEW CHECKLIST

- [ ] **Network calls** Does it phone home? Does anything go to a third-party endpoint? A server that wraps a local CLI and makes zero outbound calls is a far smaller risk than one with its own cloud backend.
- [ ] **Credential handling** How does it receive and store API keys? Environment variables are fine; config files that can end up in version control are not. Check that nothing is written to logs.
- [ ] **Scope of permissions** Does the tool set match what you actually need? A Jira server that can also delete projects offers attack surface you never asked for. Prefer least privilege – and if the server does not implement it, fork it and strip it down.
- [ ] **Source availability and activity** Is the code open? Is it maintained? A single-file project with no issues, no pull requests and a two-year-old last commit is a very different risk from an actively maintained one.
- [ ] **Dependencies** Run an audit. Not because it catches everything, but because it costs almost nothing and filters out the obvious.

```
cd your-mcp-server && npm audit --audit-level=high
```

Takes 10 seconds, occasionally saves you from yourself.

WHY ENTERPRISE CHANGES THE CALCULUS

Compliance GDPR, SOC 2, ISO 27001 and sector-specific frameworks do not bend because a developer found a convenient new tool. A server that logs tool calls – even only for debugging – may be storing data you are not allowed to hand to a third party.

Blast radius An enterprise server is usually authenticated against more: the cloud provider, internal APIs, communication tools. A compromise is an incident, not a personal inconvenience.

Visibility If nobody can say which MCP servers your team is running, you already have a shadow IT problem. Adoption outpaces the speed at which security teams hear about it.

> **Prompt injection through tool responses is a real attack vector.** A malicious server can return crafted content that nudges the model into doing something unintended – and if that model also holds a file system or a code executor, the risk compounds.

MAKE IT A HABIT, NOT A GATE

Keep a register One list of the MCP servers in use, who owns each, and when it was last reviewed.

Re-review on upgrade A server that passed six months ago is a different piece of software today.

Make reporting safe Aim for a culture where “I added a new MCP server” is said out loud, not hidden.

SOURCES

Model Context Protocol – Security Best Practices
https://modelcontextprotocol.io/docs/tutorials/security/security_best_practices

Protecting against indirect prompt injection attacks in MCP – Microsoft for Developers
https://developer.microsoft.com/blog/protecting-against-indirect-injection-attacks-mcp

MCP Tool Poisoning – OWASP
https://owasp.org/www-community/attacks/MCP_Tool_Poisoning

The MCP Attack Surface: Top-20 Documented Attacks (2026)
https://agyn.io/blog/mcp-attack-surface