MCP Security Validation Checklist

By the Feluda.ai editorial team

This checklist by Feluda.ai ensures a systematic review of MCP security posture. Critical items should be addressed **before deployment**.

High and Medium risks should be remediated or mitigated within a defined timeline, depending on your organisation's risk appetite.

Use this checklist to assess whether an MCP server has been deployed securely and in line with best practices.

Authentication & Access Control

Authentication and authorization are the first lines of defense. Without them, anyone could connect to your MCP server and use its tools.

- Are strong authentication mechanisms in place (OAuth2, mTLS, or signed tokens)? **(Critical)**
- Is **role-based access control (RBAC)** enforced for clients and tools? **(Critical)**
- Do all users and clients follow the **principle of least privilege**? **(High)**
- Are credentials rotated regularly and revoked when no longer needed? **(High)**

Tool & Resource Protection

Every tool you expose through MCP increases the attack surface. Validators must confirm that dangerous tools are gated, validated, and controlled.

- Has every exposed tool been reviewed for **safety and necessity**? **(Critical)**
- Are sensitive tools (file write, exec, system calls) restricted or sandboxed? **(Critical)**
- Are tool inputs strictly validated against JSON schemas? **(High)**
- Is there a **tool allowlist/denylist** policy in place? **(Medium)**

Network & Transport Security

MCP traffic often includes sensitive prompts and tool outputs. Weak transport security exposes this to interception or tampering.

- Is **TLS 1.2+** enforced for all remote connections? **(Critical)**
- Is plaintext stdio blocked from exposure to untrusted clients? **(High)**
- Are **firewall rules** in place to restrict outbound (egress) connections? **(High)**

- Is **rate limiting or throttling** enabled to prevent abuse? **(Medium)**

Sandboxing & Isolation

The MCP server and its tools should never run with full system access. Sandboxing keeps compromise contained.

- Is the MCP server running in a hardened container, VM, or isolated environment? **(High)**
- Are **AppArmor/SELinux profiles** or equivalent enabled? **(Medium)**
- Is filesystem access restricted (read-only mounts, no access to /etc, no secrets)? **(Critical)**
- Are risky tools sandboxed in separate processes or containers? **(High)**

Logging & Monitoring

Logs provide visibility and accountability. But if not handled properly, they can also leak sensitive information.

- Are all tool calls and client requests logged? **(High)**
- Are logs scrubbed of sensitive data (API keys, passwords, secrets)? **(Critical)**
- Are logs forwarded to a **central SIEM** for monitoring and anomaly detection? **(High)**
- Is there an **audit trail** that records who did what, when, and with what inputs? **(High)**

LLM Guardrails

LLMs are powerful but unpredictable. Guardrails ensure that hallucinations or prompt injections don't cause harm.

- Are all LLM-generated tool requests validated before execution? **(Critical)**
- Is there protection against **prompt injection** and malicious chaining? **(Critical)**
- Is there a rate limit on repeated tool calls from the same session? **(Medium)**
- Are unusual or high-risk requests flagged for human review? **(High)**

Secrets Management

MCP servers often handle API keys, tokens, and credentials. Poor secrets management is a direct path to compromise.

- Are secrets stored in a **vault system** ? **(Critical)**
- Are there no hardcoded credentials in configs or code? **(Critical)**
- Are secrets rotated regularly? **(High)**
- Are environment variables minimal and sanitized? **(Medium)**

Supply Chain Security

MCP servers rely on external packages and plugins. Attackers increasingly target supply chains to inject malware.

- Are all third-party MCP tools and plugins vetted before deployment? **(High)**
- Are dependencies **pinned to specific versions**? **(Medium)**

- Is software composition analysis (SCA) run to identify vulnerable dependencies? **(High)**
- Are package integrity checks (signatures, checksums) in place? **(Medium)**

Deployment & Ops Hardening

A secure MCP server must be deployed on a hardened base. Misconfigured infrastructure can undermine every other control.

- Is the MCP server running under a **non-root user**? **(Critical)**
- Is the OS hardened and patched? **(High)**
- Are backups and disaster recovery plans in place? **(Medium)**
- Is deployment reproducible via Infrastructure-as-Code (IaC)? **(Medium)**

Testing & Incident Response

Security isn't static. Continuous testing and a clear incident response plan are critical to MCP resilience.

- Has the MCP server undergone **penetration testing**? **(High)**
- Are simulated malicious inputs tested against defenses? **(High)**
- Are runbooks in place for incident response if MCP is abused? **(Critical)**
- Is the threat model reviewed and updated regularly? **(Medium)**

Final Validation Question

- If this MCP server were compromised today, would the **blast radius** be contained, monitored, and recoverable? **(Critical)**

Bottom Line

For both security leaders and engineering teams, the decision is clear:

- You can commit to designing, implementing, and continuously maintaining a complex, multi-layered security framework for MCP on your own. This path demands ongoing investment in engineering time, red-teaming, compliance alignment, and constant monitoring.
- Or you can adopt Feluda.ai, where those protections are already built, tested, and integrated into the platform.

For CISOs, Feluda means reduced risk exposure, faster compliance alignment, and assurance that the attack surface is contained. For engineers, it removes the burden of plumbing and patching, allowing focus on delivering features and automation instead of constantly firefighting security gaps.

With Feluda, your organisation gets the full power of MCP without the pain, risk, and cost of DIY security hardening — enabling you to innovate with confidence while we ensure the foundation stays secure and resilient.