The OWASP MCP Top 10 Through an Attacker's Eyes



Meet the Speakers



Mugoya Hillarious

AI Security & Governance Researcher,
CCEP, ISO 42001 Lead Auditor, ISO/IEC 27701:2025 Lead Auditor

About

Emerging technology professional focused on AI security, governance, and cybersecurity awareness

, contributed to multiple developer and research communities, including the Cloud Security Alliance AI Safety Working Group and Indabax Uganda, various Google Developer Groups among others.



Mohammed Kateregga

About

Mohammed Kateregga is a cybersecurity analyst, cloud engineer, data scientist, full stack developer, and open-source community leader based in Uganda. His work focuses on cloud security, identity and access management, secure AI applications, and data protection. He leads GDG Cloud Jinja and serves as CEO and Technical Team Lead at Hello World Africa.

MCP
Dev Summit
Nairobi

Session Overview

1. What is OWASP
2. What's the OWASP MCP Top 10
3. The 10
4. MCP01: Token Mismanagement
5. MCP02: Privilege Escalation via Scope Creep
6. MCP03: Tool Poisoning
7. MCP04: Software Suppy Chain Attacks
8. MCP:05 Command Injection & Execution
9. MCP06: Intent Flow Subversion
10. MCP07: Insufficient Authentication & Authorization
11. MCP08: Lack of Audit & Telemetry
12. MCP09: Shadow MCP Servers
13. MCP10: Context Injection & Over-Sharing
14. Governance & Lifecycle Controls
15. Least Privilege Access Boundaries
16. Automated Policy Enforcement
17. Continuous Monitoring & Telemetry
18. References
19. Feedback Request

MCP Dev Summit Nairobi

What is OWASP

OWASP stands for the **Open Worldwide Application Security Project**, a global non-profit foundation dedicated to improving the security of software. Founded in 2001, it operates as an open community that provides free, vendor-neutral security tools, documentation, and standards to help developers and organizations build safer applications.

What's the OWASP MCP Top 10

The first industry-standard security framework dedicated to identifying and mitigating the most critical vulnerabilities in Model Context Protocol (MCP) implementations.

The 10

Risk ID & Name	Classification	Primary Failure Mode
MCP01: Token Mismanagement	Technical	Hardcoding API credentials, caching keys in memory, or passing tokens in prompts
MCP02: Privilege Escalation via Scope Creep	Both	Accumulating excessive operational permissions across multiple connected tools
MCP03: Tool Poisoning	Both	Altering tool descriptions and schemas to trick models into malicious behaviors
MCP04: Software Supply Chain Attacks	Governance	Importing untrusted packages, malicious dependencies, or abandoned open-source servers
MCP05: Command Injection & Execution	Technical	Passing unsanitized agent arguments into system shells or database drivers

The 10

Risk ID & Name	Classification	Primary Failure Mode
MCP06: Intent Flow Subversion	Technical	Contextual prompt injection manipulating model reasoning and tool selection
MCP07: Insufficient AuthN & AuthZ	Both	Relying on ambiguous identities, missing MFA, or accepting unvalidated tokens
MCP08: Lack of Audit & Telemetry	Governance	Operating without SIEM retention standards or oversight policies for agent activity
MCP09: Shadow MCP Servers	Governance	Running unsanctioned local stdio or third-party servers without security visibility
MCP10: Context Injection & Over-Sharing	Technical	Injecting unauthorized data into prompts or leaking cross-tenant memory

MCP01: Token Mismanagement

ATTACK MECHANISM

Contextual secret leakage happens when embedded credentials in MCP configs, environment variables, or prompts are persisted in agent memory, vector DBs, trace dumps, and logs. This leaks sensitive data across conversations, enabling lateral movement across CI/CD and cloud APIs.

DETECTION CHECKLIST

- Hardcoded keys in MCP configs (local stdio or JSON manifests)
- Unredacted prompt/completion telemetry and log exports
- Static shared service accounts reused across disparate users
- Token lifetimes exceeding session duration without expiry

ARCHITECTURAL REMEDIATION

- Enforce runtime-only secret injection through dedicated external secret managers (HashiCorp Vault, AWS Secrets Manager, or HSMs).
- Implement automated ephemeral context scrubbing and real-time redaction pipelines.
- Issue short-lived, session-bound tokens tied strictly to task duration.

MCP02: Privilege Escalation via Scope Creep

ATTACK MECHANISM

Cumulative expansion of autonomous agent access via drift, coarse permissions, and unmonitored parameter changes, enabling IaC modification, unreviewed deployments, or privilege escalation beyond the original mandate.

DETECTION CHECKLIST

- Manual permission adjustments to agent deployment manifests without change logs
- Multiple agents sharing a single overarching identity
- Missing expiration timestamps on elevated operational privileges
- Ad-hoc test promotions without formal approval gates

ARCHITECTURAL REMEDIATION

- Fine-grained, resource-specific OAuth scopes
- Policy-as-Code enforced in CI/CD PR checks.
- Just-in-Time (JIT) permission elevation requiring explicit human authorization.
- Automated entitlement drift audits and continuous scope reconciliation.

MCP03: Tool Poisoning

ATTACK MECHANISM

Adversaries tamper with tool definitions, JSON schemas, parameter annotations, or operational descriptions. By manipulating prompt instructions embedded inside schema metadata, benign operations map to destructive actions (e.g., archive mapping to DELETE), bypassing superficial validation and eroding audit trust.

PREVENTION & HARDENING CONTROLS

- Sign all schemas using JWS (JSON Web Signature) or corporate PKI.
- Content-addressable hash verification across immutable Git registries.
- Open Policy Agent (OPA) / Rego semantic invariant checks prior to runtime registration.

PRE-INSTALL STATIC DETECTION INDICATORS

- Model-Directed Imperatives: Directives embedded in descriptions like "ignore previous instructions" or "system override".
- Sensitive-Path References: References pointing to sensitive paths (~/.ssh, id_rsa, .env, .aws/credentials, /etc/passwd).
- Exfiltration Endpoints: Descriptions routing diagnostic outputs or payloads to untrusted external URLs or webhooks.
- Hidden/Zero-Width Unicode: Evasion characters (U+200B–U+200F, U+202A–U+202E, U+2060, U+FEFF).
- Comment Smuggling: Payloads concealed inside HTML/Markdown comments (<!-- ... -->) within schema text.

MCP04: Software Supply Chain Attacks

ATTACK MECHANISM

Compromised upstream MCP SDKs, third-party server plugins, community connectors, or vector database client libraries containing backdoors, rogue schemas, or floating "latest" dependencies introducing cross-tenant data exfiltration pathways.

DETECTION CHECKLIST

- Missing cryptographic signature validation checks upon SDK/server startup
- Incomplete or absent Software Bill of Materials (SBOM) and Cryptographic Bill of Materials (CBOM)
- Unisolated plugin components initiating unauthorized outbound network sockets

PREVENTION & HARDENING CALLOUT

- Mandatory cryptographic signing for all MCP SDKs, plugins, and container images.
- Generate and enforce continuous SBOM and CBOM snapshots across all build pipelines.
- Pin exact dependency versions via internal, air-gapped package mirrors.
- Strictly isolate third-party plugins in WebAssembly (WASM) or container sandboxes.

MCP05: Command Injection & Execution

ATTACK MECHANISM

Model-mediated translation of untrusted conversational prompts or retrieved context into raw shell commands, SQL queries, or filesystem path traversal operations (../../../etc/passwd), chained via shell metacharacters.

RUNTIME DETECTION INDICATORS

- **Syscall Anomalies:** Runtime traces of `execve` invocations via Falco, auditd, or osquery.
- **Privilege Escalation:** Sudo / SUID binary execution by agent worker processes.
- **Network Egress:** Outbound connections initiated by local tools to unmonitored external IPs.
- **Resource Saturation:** Unexplained CPU or disk I/O saturation spikes during execution.

ARCHITECTURAL MITIGATION & HARDENING

- Execution Policy: Strictly prohibit shell=True, eval(), and exec(); enforce structured arrays and parameterized SQL.
- Runtime Sandbox: Isolate tool processes in micro-VMs (Kata Containers, Firecracker) or gVisor sandboxes running strictly as non-root.

MCP06: Intent Flow Subversion

Attack Mechanism:

In-flow goal hijacking where hidden instructions inside retrieved resources/ or tool outputs override the user's original intent (e.g., pivoting from 'Summarize logs' to 'Exfiltrate logs to attacker webhook') and persist across multi-turn sessions.

ARCHITECTURAL FLAW INDICATORS

- Merging system instructions, user intent, and untrusted external resources into a flat prompt window.
- Blind agent re-planning after reading external or tool-supplied data.

MCP06: Intent Flow Subversion

ARCHITECTURAL PREVENTION FRAMEWORK

- SEMANTIC ANCHORING
  Continuous relevance scoring against original user goal at every planning step; block unaligned verbs (e.g., reject Delete/Write on a Read request).
- THE CHECKER PATTERN
  Interpose an independent Guardrail Model that sees only User Intent + Proposed Action (completely isolated from poisoned MCP context).
- STRICT CONTEXT TAGGING
  Wrap external outputs in [UNTRUSTED_CONTEXT] metadata tags; immediately pause sessions upon detecting Intent Drift.

MCP07: Insufficient Authentication & Authorization

THREAT MECHANISM & EXPLOIT PATH

Absence of mutual cryptographic authentication between MCP clients, orchestrator agents, and servers, leading to reliance on unverified caller identity headers, client-side permission checks, or shared static tokens enabling cross-agent impersonation and unauthorized workflow chaining.

DETECTION & OPERATIONAL CONTROLS

- Mandatory Mutual TLS (mTLS): Enforce cryptographic identity verification across all client-to-agent and agent-to-server network communication channels.
- Dynamic Per-Request Evaluation: Implement Attribute-Based Access Control (ABAC) and Role-Based Access Control (RBAC) evaluated at every tool invocation under an explicit deny-by-default posture.
- Enterprise IAM Integration: Integrate all agent identities directly with enterprise OpenID Connect (OIDC) and centralized IAM domains.
- SOC Alerting Triggers: Configure real-time alerts on HTTP 403 authorization denial spikes, attempts to reuse expired tokens, and cross-IP token presentation anomalies.

MCP08: Lack of Audit & Telemetry

THREAT MECHANISM & EXPLOIT PATH

Autonomous tool invocations and state changes executing without structured, immutable, and centralized telemetry. Results in severe visibility blind spots, non-compliance with regulatory mandates (GDPR, PCI DSS 1-year log retention, ISO 27001), and inability to conduct post-incident forensics.

DETECTION & TELEMETRY HARDENING

- Structured Schema Telemetry: Standardize all logging to emit structured JSON, Common Event Format (CEF), or OpenTelemetry (OTel) traces containing: timestamp, agent_id, session_id, tool_invoked, parameters_used, response_summary, user_identity, and schema_version.
- Cryptographic Log Integrity: Ensure immutability by signing log batches using HMAC or SHA-256 hashes, streaming them to Write-Once-Read-Many (WORM) storage (e.g., AWS S3 Object Lock in compliance mode).
- Field-Level Privacy Redaction: Apply automated field-level encryption and real-time PII tokenization prior to log persistence.
- Verification Drills: Conduct recurring adversarial log-tampering and reconstruction audit drills.

MCP09: Shadow MCP Servers

ATTACK MECHANISM

Proliferation of unsanctioned, unmanaged MCP server instances spun up by internal development teams or autonomous agent experimentation. Rogue servers operate on unmonitored local ports (8000, 8080) or custom routes (/mcp, /agent/tools, /context) with default credentials connected to sensitive internal databases and production APIs.

DETECTION & OPERATIONAL CONTROLS

- Centralized Enterprise Registry: Require all MCP servers to be formally registered in an enterprise catalog detailing owner, business purpose, Git repository version, and compliance certification.
- Automated Network Discovery: Execute weekly subnet scans using Nmap, Cloud Security Posture Management (CSPM), and External Attack Surface Management (EASM) tools, coupled with passive network sensors sniffing for /mcp routes and self-signed certificates.
- Golden Experimentation Sandboxes: Provide pre-configured, hardened developer sandboxes to disincentivize ad-hoc server deployment.
- Automated Quarantine: Implement instant container/process snapshotting and network isolation upon discovering rogue listeners.

MCP10: Context Injection & Over-Sharing

ATTACK MECHANISM

Unrestricted context accumulation leading to cross-user, cross-team (e.g., Marketing agent reading Support dispute transcripts), or multi-tenant vector store bleed caused by shared context buffers and missing Time-to-Live (TTL) purge rules.

DETECTION & ARCHITECTURAL ISOLATION CONTROLS

- Ephemeral Session Lifecycles: Enforce strict Time-To-Live (TTL) expiration policies across all context stores (purge upon session end, 30m idle, or 24h hard limit).
- Strict Namespace Segmentation: Logically and cryptographically isolate vector database partitions and indexes by Tenant ID, Workflow ID, Agent Identity, and User Role.
- Granular Data Tagging: Implement classification tags (Public, Internal, Confidential, Restricted) across context inputs, running real-time PII and secret scanners prior to embedding.
- Human-in-the-Loop Verification: Require explicit HITL preview before context data is shared across disparate agent workflows.

PILLAR 01 · MCP04, MCP07 & MCP09

Governance & Lifecycle Controls

01

Signed Manifests

Cryptographic identity replaces ad-hoc scripts.

02

Pinned Supply Chain

SBOM verification blocks unverified packages.

03

Fail-Closed Boot

Mandatory token validation at startup.

PILLAR 02 · MCP01, MCP02 & MCP10

Least-Privilege Access Boundaries

Unscoped Exposure

- Hardcoded Keys: Static secrets inside source code.
- Scope Creep: Debug flags leak runtime state.
- Raw Dumps: Unfiltered PII in model memory.

Enforced Minimization

Runtime Vaults: Short-lived scoped session tokens.

Role Isolation: Admin states strictly decoupled.

Field Filtering: Curated attributes per request.

PILLAR 03 · MCP03, MCP05 & MCP06

Automated Policy Enforcement

Regex Input Gate

Strict alphanumeric parameter allowlisting.

Shell-Free Execution

Native tarfile APIs replace shell calls.

Context Encapsulation

Boundary tags isolate untrusted notes.

Continuous Monitoring & Telemetry

Telemetry Surface	Silent STDIO Risk	Hardened Audit Control
Log Verbosity	CRITICAL suppression only	Structured INFO/WARN streams
Caller Attribution	Anonymous local pipe	Verified token attribution
Blocked Payloads	Silent regex rejection	Real-time security alerts
Tool Schemas	Unwatched docstring drift	Cryptographic hash logging

References

[1] OWASP Foundation, "MCP01-2025: Token Mismanagement and Secret Exposure," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP01-2025-Token-Mismanagement-and-Secret-Exposure.md

[2] OWASP Foundation, "MCP02-2025: Privilege Escalation via Scope Creep," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP02-2025–Privilege-Escalation-via-Scope-Creep.md

[3] OWASP Foundation, "MCP03-2025: Tool Poisoning," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP03-2025–Tool-Poisoning.md

[4] OWASP Foundation, "MCP04-2025: Software Supply Chain Attacks & Dependency Tampering," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP04-2025–Software-Supply-Chain-Attacks%26Dependency-Tampering.md

-

[5] OWASP Foundation, "MCP05-2025: Command Injection & Execution," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP05-2025–Command-Injection%26Execution.md
[6] OWASP Foundation, "MCP06-2025: Intent Flow Subversion," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP06-2025–Intent-Flow-Subversion.md
[7] OWASP Foundation, "MCP07-2025: Insufficient Authentication & Authorization," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP07-2025–Insufficient-Authentication%26Authorization.md
[8] OWASP Foundation, "MCP08-2025: Lack of Audit and Telemetry," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP08-2025–Lack-of-Audit-and-Telemetry.md

-

[9] OWASP Foundation, "MCP09-2025: Shadow MCP Servers," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP09-2025–Shadow-MCP-Servers.md
[10] OWASP Foundation, "MCP10-2025: Context Injection & Over-Sharing," *OWASP Model Context Protocol Top 10*, GitHub repository, 2025. [Online]. Available: https://github.com/OWASP/www-project-mcp-top-10/blob/165fe0f78ef104459237b4a8e0f6e78db9b02391/2025/MCP10-2025–ContextInjection%26OverSharing.md
[11] International Organization for Standardization, *Information Technology — Artificial Intelligence — Management System*, ISO/IEC Standard 42001:2023, 2023. [Online]. Available: https://www.iso.org/standard/42001.html
[12] National Institute of Standards and Technology, *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, U.S. Department of Commerce, NIST AI 100-1, 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf
[13] Cloud Security Alliance, *AI Model Risk Management Framework*, CSA Research, Jul. 2024. [Online]. Available: https://cloudsecurityalliance.org/artifacts/ai-model-risk-management-framework

-

Your Feedback Matters!



Thank You

AfricanTechVC

-

MCP
Dev Summit
Nairobi