# MCP SECURITY & COMPLIANCE AUDIT

## @modelcontextprotocol/server-filesystem

https://github.com/modelcontextprotocol/servers/tree/main/src/filesystem

| SECURITY HYGIENE | AGENT TRUST | OVERALL RISK |
|---|---|---|
| 7/100<br>Grade F | 21.8/100<br>Grade F | CRITICAL |

| | |
|---|---|
| **Report ID** | LF-2026-DEMO |
| **Audit Date** | 2026-03-05 |
| **Prepared For** | LuciferForge Security (Public Demo) |
| **Prepared By** | LuciferForge Security |
| **Methodology** | mcp-security-audit v0.2.0 + AgentCred v0.1.0 |
| **Standards** | EU AI Act (2024/1689) + NIST AI RMF 1.0 |

# 1. Executive Summary

This report presents the results of an independent security and compliance audit of @modelcontextprotocol/server-filesystem, conducted on 2026-03-05 by LuciferForge Security.

The audit assessed the server across two dimensions: (1) MCP Security Hygiene, using the mcp-security-audit framework — scoring documentation completeness, input schema rigor, injection safety, scope discipline, and metadata hygiene; and (2) Agent Trust Posture, using the AgentCred framework — scoring identity completeness, security posture, reliability signals, and behavioral reputation.

The server received a Security Hygiene Score of 7/100 (Grade F) and an Agent Trust Score of 21.8/100 (Grade F). The overall risk classification is CRITICAL, indicating an unacceptable level of residual risk under the EU AI Act risk management framework.

The audit identified 7 finding(s): 0 Critical, 2 High, 3 Medium. Deployment is not recommended until all Critical and High findings are resolved.

Each finding in this report is mapped to the applicable EU AI Act article and NIST AI Risk Management Framework (AI RMF) function to enable direct integration with your compliance program.

# 2. Score Breakdown

### 2.1 MCP Security Hygiene (mcp-security-audit)

| Category | Score | Max | Description |
|---|---|---|---|
| Documentation | 0.0 | 25 | Tool/parameter descriptions present and substantive |
| Schema Rigor | 0.0 | 25 | Input validation, constraints, required fields |
| Injection Safety | 0.0 | 25 | No injection patterns in tool/prompt/resource text |
| Scope & Least Privilege | 3.0 | 15 | Tool count, destructive tool discipline, shell safety |
| Metadata | 4.0 | 10 | Server name, version, naming consistency |
| **TOTAL** | **7** | **100** | **Grade: F** |

### 2.2 Agent Trust Score (AgentCred)

| Bucket | Weight | Score | Weighted |
|---|---|---|---|
| Identity Completeness | 20% | 15.8/100 | 3.2 |
| Security Posture | 25% | 46.8/100 | 11.7 |
| Reliability | 35% | 0.0/100 | 0.0 |
| Behavioral Reputation | 20% | 35.0/100 | 7.0 |
| **COMPOSITE** | **100%** | | **21.8/100 — Grade F** |

No KYA (Know Your Agent) identity card was found for this server. Under EU AI Act Article 13 (Transparency) and Article 11 (Technical Documentation), high-risk AI systems must provide sufficient documentation for users to interpret outputs and exercise appropriate oversight. A KYA card provides machine-readable identity, ownership declaration, capability scope, and compliance framework declarations. Its absence reduces the agent trust score and is a documentation gap.

## 3. Detailed Findings

| Severity | Count |
|---|---|
| HIGH | 2 |
| MEDIUM | 3 |
| LOW | 2 |
| **TOTAL** | **7** |

| **Finding 1: 13 undocumented tool(s)** | **MEDIUM** |
|---|---|
| **Category** | Documentation |
| **Detail** | Tools without descriptions: ['read_file', 'write_file', 'edit_file', 'create_directory', 'list_directory', 'list_directory_with_sizes', 'directory_tree', 'move_file', 'search_files', 'get_file_info', 'list_allowed_directories', 'delete_file', 'patch_file'] |
| **EU AI Act** | Article 13(3)(b) — Transparency — Instructions for Use |
| **Requirement** | Instructions must include the purpose, accuracy, and limitations of the system. |
| **Compliance Implication** | Undocumented tools prevent downstream users from assessing system capabilities and limitations. |
| **NIST AI RMF** | GOVERN 2.2 / MAP 2.3 |
| **Remediation Effort** | 1-2 weeks |
| **Priority** | Next sprint |

| **Finding 2: 3 unconstrained object parameter(s)** | **MEDIUM** |
|---|---|
| **Category** | Schema |
| **Detail** | Parameters typed as bare 'object' with no properties defined accept arbitrary input |
| **EU AI Act** | Article 15(1) — Accuracy, Robustness and Cybersecurity |
| **Requirement** | High-risk AI systems must be designed to achieve an appropriate level of accuracy, robustness and cybersecurity. |
| **Compliance Implication** | Lack of input constraints reduces robustness against malformed or adversarial inputs. |
| **NIST AI RMF** | MANAGE 2.2 / MEASURE 2.6 |
| **Remediation Effort** | 1-2 weeks |
| **Priority** | Next sprint |

| Finding 3: No string parameters use constraints | LOW |
|---|---|
| **Category** | Schema |
| **Detail** | 28 string params lack enum, pattern, maxLength, or format constraints |
| **EU AI Act** | Article 17(1)(d) — Quality Management — Data Governance |
| **Requirement** | Providers must implement data governance and management practices. |
| **Compliance Implication** | Parameter constraints are a data quality control — their absence is a gap in the quality management system. |
| **NIST AI RMF** | MANAGE 2.2 / MEASURE 2.6 |
| **Remediation Effort** | Ongoing |
| **Priority** | Technical debt queue |

| Finding 4: Destructive tools lack descriptions: delete_file, write_file, move_file | HIGH |
|---|---|
| **Category** | Scope |
| **Detail** | Tools with destructive operations (delete, write, move) have no descriptions to guide safe use |
| **EU AI Act** | Article 9(5) — Risk Management — Least Privilege |
| **Requirement** | Risk management must consider the reasonably foreseeable misuse of the system. |
| **Compliance Implication** | Unexpected high-risk capabilities beyond stated server purpose indicate scope creep that must be risk-assessed. |
| **NIST AI RMF** | MAP 1.6 / MANAGE 1.3 |
| **Remediation Effort** | 2-5 days |
| **Priority** | Sprint 0 — before production |

| Finding 5: search_files: path parameter accepts unconstrained regex with no validation | HIGH |
|---|---|
| **Category** | Injection |
| **Detail** | The pattern parameter for search_files accepts arbitrary regex with no maxLength or sanitization — adversarial ReDoS or path traversal patterns are feasible |
| **Tool** | search_files |
| **EU AI Act** | Article 9(2)(b) — Risk Management — Risk Estimation |
| **Requirement** | Providers must estimate and evaluate risks that may emerge when the system is used. |
| **Compliance Implication** | High-severity injection patterns indicate failure to estimate adversarial misuse potential. |
| **NIST AI RMF** | GOVERN 1.2 / MAP 1.5 / MEASURE 2.5 |
| **Remediation Effort** | 2-5 days |
| **Priority** | Sprint 0 — before production |

| Finding 6: Server does not declare a version string | MEDIUM |
|---|---|
| **Category** | Metadata |
| **Detail** | Server initialization did not include a version — audit trail non-identifiable |
| **EU AI Act** | Article 11(2)(a) — Technical Documentation — System Description |
| **Requirement** | Documentation must include a general description of the AI system. |
| **Compliance Implication** | Missing server name/version makes the system non-identifiable in a compliance audit trail. |
| **NIST AI RMF** | GOVERN 1.7 |
| **Remediation Effort** | 1-2 weeks |
| **Priority** | Next sprint |

| Finding 7: No rate limiting or max-path-depth constraints declared | LOW |
|---|---|
| **Category** | Scope |
| **Detail** | Recursive directory traversal tools (directory_tree) have no declared depth limits |
| **Tool** | directory_tree |
| **EU AI Act** | Article 9(5) — Risk Management — Least Privilege |
| **Requirement** | Risk management must consider the reasonably foreseeable misuse of the system. |
| **Compliance Implication** | Unexpected high-risk capabilities beyond stated server purpose indicate scope creep that must be risk-assessed. |
| **NIST AI RMF** | MAP 1.6 / MANAGE 1.3 |
| **Remediation Effort** | Ongoing |
| **Priority** | Technical debt queue |

## 4. Remediation Roadmap

Findings are grouped by remediation priority. Address items in order — Critical items represent the highest regulatory and security risk.

**Sprint 0 — before production (2 finding(s))**

| # | Finding | Category | Effort |
|---|---|---|---|
| 1 | Destructive tools lack descriptions: delete_file, write_file, move_file | Scope | 2-5 days |
| 2 | search_files: path parameter accepts unconstrained regex with no validation | Injection | 2-5 days |

**Next sprint (3 finding(s))**

| # | Finding | Category | Effort |
|---|---|---|---|
| 1 | 13 undocumented tool(s) | Documentation | 1-2 weeks |
| 2 | 3 unconstrained object parameter(s) | Schema | 1-2 weeks |
| 3 | Server does not declare a version string | Metadata | 1-2 weeks |

**Technical debt queue (2 finding(s))**

| # | Finding | Category | Effort |
|---|---|---|---|
| 1 | No string parameters use constraints | Schema | Ongoing |
| 2 | No rate limiting or max-path-depth constraints declared | Scope | Ongoing |

**Agent Trust Improvement Recommendations**

- Sign the KYA card with Ed25519: kya keygen && kya sign
- Run injection testing: pip install ai-injection-guard
- Enable decision logging for audit trails
- Implement a kill switch for emergency shutdown
- Add created_at timestamp to metadata

## 5. Tool Inventory

| Tool Name | Risk Category | Purpose Aligned | Matched Patterns |
|---|---|---|---|
| read_file | File System | Yes | read, file |
| read_multiple_files | File System | Yes | read, file |
| write_file | File System | Yes | write, file |
| edit_file | File System | Yes | edit, file |
| create_directory | File System | Yes | create, directory |
| list_directory | File System | Yes | list, directory |
| list_directory_with_sizes | File System | Yes | list, directory |
| directory_tree | File System | Yes | directory, tree |
| move_file | File System | Yes | move, file |
| search_files | File System | Yes | search, file |
| get_file_info | File System | Yes | file, info |
| list_allowed_directories | File System | Yes | list, directories |
| delete_file | File System | Yes | delete, file |
| patch_file | File System | Yes | patch, file |

## 6. Methodology

**Audit Tools**

**mcp-security-audit v0.2.0** (github.com/LuciferForge/mcp-security-audit): Connects to the MCP server via the standard stdio protocol, enumerates all tools, resources, and prompts, classifies each tool by risk category using pattern matching against 40+ patterns, scans tool/prompt/resource text for 22 injection patterns using ai-injection-guard, and scores the server on 5 hygiene categories totaling 100 points. No live tool invocations were performed — audit is static analysis only.

**AgentCred v0.1.0** (github.com/LuciferForge/agentcred): Scores agent trust across 4 buckets — Identity Completeness (20%), Security Posture (25%), Reliability (35%), Behavioral Reputation (20%) — using static signals from KYA cards and security audit results.

**Regulatory Framework**

Findings are mapped to the EU Artificial Intelligence Act (Regulation 2024/1689), specifically Articles 9 (Risk Management), 11 (Technical Documentation), 13 (Transparency), 15 (Accuracy and Robustness), and 17 (Quality Management System). NIST AI Risk Management Framework 1.0 function references are also provided (GOVERN, MAP, MEASURE, MANAGE).

**Limitations**

This audit assesses static properties observable at audit time. It does not assess: runtime behavior, authentication/authorization mechanisms, network security, data handling practices, or third-party dependency security. Dynamic testing (live tool invocation) was not performed. A clean audit does not guarantee absence of all security issues.