EU AI Act

Compliance assessment — 2024/1689

Server: @modelcontextprotocol/server-filesystem

Slug: modelcontextprotocol-server-filesystem-20260910034240-4b25df
Scan id: 9bbce6c6-8866-42d6-bf2c-736b6a22d43d

Assessed at: 2026-09-10 03:42:40 UTC
Sentinel version: 0.4.0
Rules version: 2026-04-23

' Non-compliant

DRAFT for review — not legal advice. See attestation block for verification instructions.

Table of contents

1. Executive summary

Assessment of @modelcontextprotocol/server-filesystem against EU AI Act: overall status non compliant. Of 5 controls, 0 met, 3 unmet, 0 partial, 2 not assessed, 0 not applicable. 3 control(s) had complete mapped-assessor execution evidence; 2 lacked sufficient execution evidence and 0 were outside the current assessor mapping. Unmet controls have findings at or above the framework's mandatory severity threshold and should be remediated before relying on this server in a regulated deployment. All claims are traceable to individual finding rows via finding_id and to the governing rule via rule_id; the enclosing signed envelope commits MCP Sentinel to the exact bytes of this report.

2. Coverage & transparency

Coverage band: low

Coverage ratio: 58%

Rules version: 2026-04-23

Analysis techniques applied:

- ast-taint
- composite
- cross-module
- dependency-audit
- similarity
- structural
- stub

3. Controls summary

ControlName	Status	Evidence
Art.9Risk Management System'	Unmet	

7

Art.12Record-Keeping? Not assessed0

Art.13 Transparency & Provision of Information to Deployers' Unmet3

Art.14Human Oversight? Not assessed0

Art.15 Accuracy, Robustness, and Cybersecurity' Unmet2

4. Control details

Art.9 — Risk Management System

' Unmet

20/22 mapped assessor rule(s) are proven to have executed; 7 finding(s) observed (2 high, 5 medium); at least one finding is at or above the high threshold (status: unmet). 2 mapped assessor(s) did not execute, so this finding-backed result does not imply complete control coverage.

Evidence:

[High] Known CVEs in Dependencies (finding fffe29f1-343a-495a-a77f-ff7f89583979, confidence 82%)

SOURCE: external-content at npm:minimatch@10.0.1 — Third-party package dependencies are external content resolved from public registries. A version with a published CVE ships the vulnerable code path

[High] Known CVEs in Dependencies (finding dec259c6-f8f2-45a2-969d-b1c08437640d, confidence 78%)

SOURCE: external-content at npm:vitest@4.1.8 — Third-party package dependencies are external content resolved from public registries. A version with a published CVE ships the vulnerable code path as-i

[Medium] Abandoned Dependencies (finding a2bba225-8bc7-46cc-8780-f86cc154c39e, confidence 70%)

SOURCE: external-content at npm:minimatch@10.0.1 — An unmaintained dependency receives no patches for newly disclosed vulnerabilities. Every published CVE in the package after its last release date re

[Medium]

Abandoned Dependencies (finding 78f558bb-d4f1-4602-86a6-7c693e2cc7e0, confidence 70%)
SOURCE: external-content at npm:@types/diff@5.0.9 — An unmaintained dependency receives no patches for newly disclosed vulnerabilities. Every published CVE in the package after its last release date r

[Medium] Abandoned Dependencies (finding 383eba61-3550-4fab-9bb2-3be8a3a607e2, confidence 70%)
SOURCE: external-content at npm:@types/minimatch@5.1.2 — An unmaintained dependency receives no patches for newly disclosed vulnerabilities. Every published CVE in the package after its last release d

[Medium] Abandoned Dependencies (finding 9b9f02de-f385-4c43-8841-8b08837d5e9a, confidence 70%)
SOURCE: external-content at npm:shx@0.4.0 — An unmaintained dependency receives no patches for newly disclosed vulnerabilities. Every published CVE in the package after its last release date remains p

[Medium] Abandoned Dependencies (finding 7ccf15e0-6d44-4423-8103-15f92399a2b6, confidence 70%)
SOURCE: external-content at npm:typescript@5.8.2 — An unmaintained dependency receives no patches for newly disclosed vulnerabilities. Every published CVE in the package after its last release date re

Required mitigations:

- Update the flagged dependency to a version that resolves every listed CVE. Consult the NVD or OSV advisory linked in the finding for the minimum safe version. Re-run `npm audit` / `pip-audit` / `osv-scanner` after the bump to confirm no residual advisories remain. Where a patched version is not yet released, apply an override (npm `overrides`, `resolutions`, `pnpm.overrides`, or a Python constraints file) to pin a compatible fix, and file a tracking issue so the override is removed when the upstream fix lands.
- Replace the abandoned dependency with an actively maintained alternative. Where no alternative exists, vendor the minimal code the project uses and delete the dependency. If the package is internal/private and legitimately stable, mark it as such in a repository-local allowlist so the scanner can skip it on future runs. Track the abandonment risk in the project's SBOM per ISO 27001 A.8.8.

Art.12 — Record-Keeping

? Not assessed

4/5 mapped assessor rule(s) are proven to have executed; no adverse findings were observed, but the control is not assessed because execution evidence is incomplete. Unexecuted assessors: E3 (connection_metadata).

Art.13 — Transparency & Provision of Information to Deployers

' Unmet

4/14 mapped assessor rule(s) are proven to have executed; 3 finding(s) observed (3 high); at least one finding is at or above the high threshold (status: unmet). 10 mapped assessor(s) did not execute, so this finding-backed result does not imply complete control coverage.

Evidence:

[High] Unsanitized Tool Output (finding 016a8d61-cce2-4a56-8ea4-67d4ed6000d9, confidence 90%)

SOURCE: file-content at dist/index.js:139:24 — External-content read classified as `file-read`. Values returned by this call are outside the server's trust boundary — a web fetch may return attacker-c

[High] Unsanitized Tool Output (finding 98684d5c-ce5c-469c-befd-37f0ccce0d65, confidence 90%)

SOURCE: file-content at dist/index.js:277:35 — External-content read classified as `file-read`. Values returned by this call are outside the server's trust boundary — a web fetch may return attacker-c

[High] Unsanitized Tool Output (finding 3c837148-d579-4cb3-a15a-3869763fe146, confidence 90%)

SOURCE: file-content at dist/index.js:277:35 — External-content read classified as `file-read`. Values returned by this call are outside the server's trust boundary — a web fetch may return attacker-c

Required mitigations:

- Tool responses must not carry raw external content to the AI client. Acceptable mitigations: (1) apply a sanitizer — sanitize(), sanitizeHtml(), escapeHtml(), DOMPurify.sanitize(), he.encode(), validator.escape(), stripTags() — to the returned value BEFORE emission, (2) when returning JSON, ensure every string field originating from external sources is coerced to plain text and never rendered as HTML or Markdown by the client, (3) attach a trust-boundary annotation that the client can use to quarantine the content. CoSAI MCP-T4 and OWASP ASI02 treat unsanitized tool output as a direct tool-poisoning substrate — the AI interprets the response as a trustworthy tool result and gives external content authority it never earned.

Art.14 — Human Oversight

? Not assessed

5/13 mapped assessor rule(s) are proven to have executed; no adverse findings were observed, but the control is not assessed because execution evidence is incomplete. Unexecuted assessors: K4 (tools); I12 (declared_capabilities); M5 (tools); H3 (tools); F1 (tools); F6 (tools); K15 (min_tools(2), tools); Q10 (tools).

Art.15 — Accuracy, Robustness, and Cybersecurity

' Unmet

66/111 mapped assessor rule(s) are proven to have executed; 2 finding(s) observed (2 high); at least one finding is at or above the high threshold (status: unmet). 45 mapped assessor(s) did not execute, so this finding-backed result does not imply complete control coverage.

Evidence:

[High] Unbounded Recursion /

Missing Depth Limits (finding bb914062-81d4-4d7f-ab37-7d890ff18ea0, confidence 88%)

SOURCE: file-content at dist/index.js:476:44 — Recursive call closing a cycle with entry `buildTree`: direct self-recursion on `buildTree`. The entry function declares no depth-comparison guard (Binar

[High] Unbounded Recursion / Missing Depth Limits (finding 85499f5d-d96c-4b09-91ac-033fe6bdbe1a, confidence 88%)

SOURCE: file-content at dist/lib.js:383:27 — Recursive call closing a cycle with entry `search`: direct self-recursion on `search`. The entry function declares no depth-comparison guard (BinaryExpress

Required mitigations:

• Every recursive handler must carry an explicit termination budget. For tree / DAG walks, thread a `depth` parameter with a comparison against an UPPER_SNAKE constant: `if (depth > MAX_DEPTH) return;`. For graphs with cycles, use a visited-set (`new Set()` / `new WeakSet()`) consulted with `.has()` before recursing and updated with `.add()` on entry. For MCP tool-call roundtrips (recursionA!'callTool('B')!'callTool('A')), pair the per-function guard with a session-level tool-call-depth counter carried through the tool arguments; the MCP spec does not enforce this client-side. OWASP ASI08 and EU AI Act Art.15 both require robustness against adversarial input-driven recursion.

5. Multi-step attack chains

No multi-step attack chains were synthesized for this server.

6. Cryptographic attestation

Algorithm: `HMAC-SHA256`
Key ID: `mcps-hkdf-e8990a4e1526`
Signer: `mcp-sentinel/v1`
Signed at: `2026-09-11T10:44:26.961Z`
Canonicalization: RFC8785

HMAC-SHA256 signature (base64, wrapped at 64 chars):

SyRT6HgGnA9qt1gc47plVi1vPwFEfQ4hJ+DAR/5YqTs=

Verification instructions:

```
To verify this report:
  1. Extract the report body (everything except the .attestation field).
  2. Canonicalize the body via RFC 8785 (JCS).
  3. Compute HMAC-SHA256 with the signing key for key_id "mcps-hkdf-e8990a4e1526".
  4. Base64-encode the result and compare with the signature above.
```