

Source dossier

The scope was never shown — a scan of 21,097 MCP servers confirms 150 that read credentials outside their declared contract, and the AI Act article you reach for excludes open-source tools anyway

Edition 2026-10-19 · narrative: scope-vs-approval · compiled 2026-10-19

Claims index

Claim	Statement	Status	Sources
CLAIM-2026-10-19-01	MCP-FSD catalogued 21,097 MCP servers across nine registries and triaged 4,806 of them; 787 of the triaged servers (16.4%) exhibited filesystem capabilities despite carrying non-filesystem functional classifications.	verified	arxiv2026mcpfsd, arxivexport2026record
CLAIM-2026-10-19-02	The authors characterise the 16.4% category mismatch as a coarse initial screening signal rather than a definitive violation metric, because no external ground-truth labels exist across the catalogue.	verified	arxiv2026mcpfsd
CLAIM-2026-10-19-03	Three-case dynamic validation confirmed 150 vulnerable instances; for 136 of the 150 the adjudication relied on exact credential-marker matching, and for the remaining 14 on other transcript evidence such as write side-effects or response-shape changes.	verified	arxiv2026mcpfsd
CLAIM-2026-10-19-04	Coordinated disclosure produced 13 reports, two of which became CVEs in Anthropic's Git MCP Server; both CVEs were patched upstream before the paper appeared.	verified	arxiv2026mcpfsd, nvd2025unscoped, nvd2026pathtrave
CLAIM-2026-10-19-05	CVE-2025-68143: in mcp-server-git before 2025.9.25 the git_init tool accepted arbitrary filesystem paths and created Git repositories without validating the target location, making those directories eligible for subsequent git operations; the tool was removed entirely rather than fixed.	verified	nvd2025unscoped
CLAIM-2026-10-19-06	CVE-2026-27735: in mcp-server-git before 2026.1.14 the git_add tool did not validate that paths in the files argument lay inside the repository, and because it used GitPython's repo.index.add() rather than the Git CLI, relative paths containing ../ were accepted and staged.	verified	nvd2026pathtrave

Claim	Statement	Status	Sources
CLAIM-2026-10-19-07	In typical local MCP deployments, client-side approval of a tool call grants the underlying server process broad host filesystem permissions rather than restricting it to the declared operation; the declared contract of name, description and argument schema is not an enforcement boundary.	verified	arxiv2026mcpfsd
CLAIM-2026-10-19-08	At the sink level 54.3% of static emissions are Node-ecosystem files and 42.6% Python; Python leads the PT-001 missing-validation rule with a 53.8% within-rule share while Node leads every bypass-style rule at 57% to 63%, which the authors attribute to the MCP TypeScript SDK shipping no built-in path-containment check.	verified	arxiv2026mcpfsd
CLAIM-2026-10-19-09	The paper reports that operating-system process isolation reliably enforced the inferred least-privilege boundaries in its three-case validation without disrupting benign tool functionality.	verified	arxiv2026mcpfsd
CLAIM-2026-10-19-10	AI Act Article 25 operates only within the high-risk regime, and Article 25(4)'s written-agreement duty for supplied tools, services, components and processes expressly does not apply to publicly accessible tools, services, processes or components other than general-purpose AI models released under a free and open-source licence.	verified	euartificia2024responsib
CLAIM-2026-10-19-11	GDPR Article 25(2) requires that by default only personal data necessary for each specific purpose are processed, and the obligation extends to the amount collected, the extent of processing, the storage period and their accessibility.	verified	gdprinfoeu2026data
CLAIM-2026-10-19-12	Independent corroboration: in 23 controlled local validations reported on 16 October 2026, the original configuration reached the approval-token transition in 9 of 9 reruns and completed the exact target operation in 8 of 9, evidencing an authorization-context gap in which token possession is treated as sufficient authorization even where resource scope is not jointly enforced.	verified	arxiv2026authorization
CLAIM-2026-10-19-13	The NVD record scores the same git_init flaw at CVSS v3.1 base 8.8 HIGH and CVSS v4.0 base 6.5 MEDIUM.	verified	nvd2025unscoped
CLAIM-2026-10-19-14	Author affiliations for arXiv:2610.22301, taken from the work's own front matter, are the University of Illinois Urbana-Champaign, Santa Clara University, Auburn University at Montgomery, and two independent researchers.	verified	arxiv2026mcpfsd

Claim	Statement	Status	Sources
CLAIM-2026-10-19-15	The scan's dynamic results are conditional on servers the pipeline could install and run, the paper does not measure the accuracy of its own inferred least-privilege boundaries, and a boundary inferred too tightly can flag legitimate access.	verified	arxiv2026mcpfsd

Only claims marked **verified** may carry a post. **single-source** may appear as hedged context; **disputed** must not be published at all.

MCP-FSD: Detecting Filesystem Capability Laundering in MCP Servers at Scale (My Credentials, Please)

PRIMARY

Publisher	arXiv
Published	2026-10-15
Retrieved	2026-10-19
Cite key	arxiv2026mcpfsd
URL	https://arxiv.org/abs/2610.22301

Assessment. Primary. Front matter read from arxiv.org/html/2610.22301v1; affiliations taken from the work's own author block.

Claims drawn from this source. CLAIM-2026-10-19-01, CLAIM-2026-10-19-02, CLAIM-2026-10-19-03, CLAIM-2026-10-19-04, CLAIM-2026-10-19-07, CLAIM-2026-10-19-08, CLAIM-2026-10-19-09, CLAIM-2026-10-19-14, CLAIM-2026-10-19-15

Abstract (verbatim from the work)

The Model Context Protocol (MCP) standardizes how AI agents invoke external tools. MCP servers expose tools through a declared contract consisting of a name, description, and argument schema. In typical local deployments, client-side approval of a tool call grants the underlying server process broad host filesystem permissions rather than restricting it to the declared operation. We study the resulting vulnerability class, filesystem capability laundering, where attacker-steered arguments cause a tool to access sensitive host files outside its declared contract and deployment scope. We analyze this capability gap across three layers: server-category screening (L1), per-tool contract analysis (L2), and bypassable path-validation checks (L3). We design MCP-FSD to catalog MCP servers, infer least-privilege boundaries per tool, plant credential fixtures outside those boundaries, and evaluate exploitability and sandbox containment via paired dynamic execution. Across 21,097 MCP servers from nine registries, 787 of 4,806 triaged servers (16.4%) exhibit filesystem capabilities despite non-filesystem functional classifications. Dynamic three-case validation confirms 150 vulnerable instances under tested sandbox policies. Coordinated disclosure yielded 13 security advisories, including two CVEs in Anthropic's Git MCP Server.

Front matter — authors and affiliations

Zeqian Huang (Independent Researcher); Mingyue Tang (University of Illinois Urbana-Champaign); Aonan Guan (Independent Researcher); Qiming Sun (Santa Clara University); Pengzhou He (Auburn University at Montgomery). Licence CC BY 4.0. arXiv:2610.22301v1 [cs.CR], 15 Oct 2026.

Section 4.1 — the authors' own caveat on the 16.4%

"...and absent external ground-truth labels across the entire catalog, this category mismatch serves as a coarse initial screening signal rather than a definitive violation metric."

Section 4.3 — RQ2: Confirmed Instances

"MCP-FSD confirms 150 instances, each adjudicated under the three-case dynamic validation against the inferred least-privilege boundary. For 136 of the 150 instances, adjudication relies on exact credential-marker matching: PoC 1's response contains the planted secret string, whereas PoC 3's response contains no marker bytes. For the remaining 14, the adjudicator evaluates corresponding evidence from other transcript features (write side-effects, response-shape changes, or external-CLI outputs)."

Section 4.4 — RQ3: Disclosed Outcomes

"We reported 13 findings, including two CVEs in Anthropic's Git MCP Server. MCP-FSD dynamically validates `git_add` with a full three-case PoC. The `git_init` finding is static because its disclosure is indirect: it seeds a repository that downstream Git tools then read. Both CVEs have been patched upstream. Among the 13 packages, 11 cite source-level mechanisms or configuration as evidence... Two packages rely on contract-implied scope boundaries without explicit path validation at the read sink."

Section 4.5 — RQ4: Violation Distribution

"At the sink level, 54.3% of emissions are Node-ecosystem files and 42.6% are Python files. All other languages together account for under 3.1%. Python leads PT-001 missing-validation (53.8% within-rule share), while Node leads every bypass-style rule at 57%-63%. This distribution reflects structural tooling gaps: the MCP TypeScript SDK provides no built-in path-containment [check]." Static emission level: 10,304 findings across 2,818 flagged servers (65.7% of the 4,292 with results); defence-bypass rules PT-002-PT-005 account for 61.6% against 36.4% for PT-001 missing validation.

Section 6.1 — Mitigations

"At L3, developers should enforce canonical-path containment, validate schema fields, and address symlink races at access time. An SDK-level containment helper would reach the many Node servers that currently ship no such check. At L2, a protocol extension could expose a per-tool `fs_scope` field in `tools/list`, allowing clients to inspect the requested read/write scope before approval and enforce the approved policy on server execution... Our three-case validation demonstrates that operating-system process isolation reliably enforces least-privilege boundaries without disrupting benign tool functionality. At L1, registry screening can identify filesystem capabilities for further review."

Section 6.2 — Limitations (carried into the edition)

"First, the pipeline only validates servers it can install and run. The dynamic results are conditional on successful startup and the selected runtime configurations. Second,

dynamic validation may miss a violation whose out-of-scope effect never appears in a response... Third, we do not measure the accuracy of the inferred boundaries themselves. A policy that is too restrictive can flag legitimate access, while one that is too permissive can miss a violation."

Case studies

5.1 Case 1 (L2): Unscoped Repository Creation in `git_init` (CVE-2025-68143). 5.2 Case 2 (L2): Path-Traversal Staging in `git_add` (CVE-2026-27735). A separate Memory MCP Server report demonstrated configuration injection through a persistence write when an attacker controls the launch-time persistence path; its fix was merged upstream without a CVE.

Record of arXiv identifier 2610.22301 (export API query)

PRIMARY

Publisher	arXiv export API
Published	2026-10-15
Retrieved	2026-10-19
Cite key	arxivexport2026record
URL	https://export.arxiv.org/api/query?id_list=2610.22301

Assessment. Identifier verification. A keyword search returns a paper on the subject and not necessarily the one; this record pins the ID, the version and the submission date.

Claims drawn from this source. CLAIM-2026-10-19-01

Queried 2026-10-19.

ID: <http://arxiv.org/abs/2610.22301v1> PUBLISHED: 2026-10-15T17:48:10Z UPDATED: 2026-10-15T17:48:10Z TITLE: My Credentials, Please: Detecting Filesystem Capability Laundering in MCP Servers at Scale AUTHORS: Zeqian Huang; Mingyue Tang; Aonan Guan; Qiming Sun; Pengzhou He COMMENT: None

The 2610 prefix matches the story's month; v1 is the only version at the time of retrieval.

Unscoped repository creation in mcp-server-git git_init (CVE-2025-68143)

PRIMARY

Publisher	NVD
Published	2025-12-17
Retrieved	2026-10-19
Cite key	nvd2025unscoped
URL	https://nvd.nist.gov/vuln/detail/CVE-2025-68143

Assessment. Primary vendor-adjacent record. Status Analyzed. Confirms the fix predates the paper.

Claims drawn from this source. CLAIM-2026-10-19-04, CLAIM-2026-10-19-05, CLAIM-2026-10-19-13

NVD record, retrieved 2026-10-19. Published 2025-12-17. vulnStatus: Analyzed.

Description. "Model Context Protocol Servers is a collection of reference implementations for the model context protocol (MCP). In mcp-server-git versions prior to 2025.9.25, the git_init tool accepted arbitrary filesystem paths and created Git repositories without validating the target location. Unlike other tools which required an existing repository, git_init could operate on any directory accessible to the server process, making those directories eligible for subsequent git operations. The tool was removed entirely, as the server is intended to operate on existing repositories only. Users are advised to upgrade to 2025.9.25 or newer to remediate this issue."

Scoring. CVSS v3.1 base 8.8 HIGH; CVSS v4.0 base 6.5 MEDIUM. The same flaw therefore reports as high or medium depending on which scoring version the scanner speaks.

References. github.com/modelcontextprotocol/servers commit eac56e7bcde48fb64d5a973924d05d69a7d876e6; advisory GHSA-5cgr-j3jf-jw3v.

Path-traversal staging in mcp-server-git git_add (CVE-2026-27735)

PRIMARY

Publisher	NVD
Published	2026-02-26
Retrieved	2026-10-19
Cite key	nvd2026pathtrave
URL	https://nvd.nist.gov/vuln/detail/CVE-2026-27735

Assessment. Primary vendor-adjacent record. Status Analyzed. Fixed in 2026.1.14, before the paper appeared.

Claims drawn from this source. CLAIM-2026-10-19-04, CLAIM-2026-10-19-06

NVD record, retrieved 2026-10-19. Published 2026-02-26. vulnStatus: Analyzed.

Description. "Model Context Protocol Servers is a collection of reference implementations for the model context protocol (MCP). In mcp-server-git versions prior to 2026.1.14, the git_add tool did not validate that file paths provided in the files argument were within the repository boundaries. Because the tool used GitPython's repo.index.add() rather than the Git CLI, relative paths containing ../ sequences that resolve outside the repository were accepted and staged into the Git index. Users are advised to upgrade to 2026.1.14 or newer to remediate this issue."

Scoring. CVSS v3.1 base 6.5 MEDIUM; CVSS v4.0 base 6.4 MEDIUM.

References. [github.com/modelcontextprotocol/servers pull/3164](https://github.com/modelcontextprotocol/servers/pull/3164); advisory GHSA-vjqx-cfc4-9h6v.

Data protection by design and by default - Art. 25 GDPR

SECONDARY

Publisher gdpr-info.eu

Published 2026

Retrieved 2026-10-19

Cite key gdprinfoeu2026data

URL <https://gdpr-info.eu/art-25-gdpr/>

Assessment. Secondary rendering of the Regulation text; relied on for the paragraph 2 limb only. Extraction is a paraphrase with short exact phrases, not a verbatim archive.

Claims drawn from this source. CLAIM-2026-10-19-11

Extracted 2026-10-19.

Paragraph 1. Taking into account the state of the art, implementation costs, and the nature, scope, context, purposes and risks of the processing, the controller must put in place appropriate technical and organisational measures, such as pseudonymisation, to implement the data protection principles including data minimisation, both when determining the means of processing and during the processing itself. Key phrase: "appropriate technical and organisational measures."

Paragraph 2. The controller must ensure that, by default, only the personal data necessary for each specific purpose is processed. This obligation applies to the amount of data collected, the extent of their processing, the period of their storage and **their accessibility**. Personal data must not be made accessible to an indefinite number of natural persons without the individual's intervention. Key phrase: "only personal data which are necessary for each specific purpose of the processing are processed."

Paragraph 3. An approved certification mechanism under Article 42 may be used as an element to demonstrate compliance with paragraphs 1 and 2.

Relevance to this edition. The accessibility limb of paragraph 2 is the one that reaches a default local configuration in which approving one narrowly declared tool call grants the server process reach over the host filesystem. It is a duty on the controller's own defaults, not on the supplier.

Responsibilities along the AI value chain - Article 25

SECONDARY

Publisher EU Artificial Intelligence Act

Published 2024-07-12

Retrieved 2026-10-19

Cite key euartificia2024responsib

URL <https://artificialintelligenceact.eu/article/25/>

Assessment. Secondary consolidated rendering. Used for the scope of Article 25 and the paragraph 4 open-source carve-out.

Claims drawn from this source. CLAIM-2026-10-19-10

Extracted 2026-10-19.

Paragraph 1. A distributor, importer, deployer or other third party is treated as the provider of a high-risk AI system where it (a) puts its own name or trademark on a high-risk system already placed on the market, (b) makes a substantial modification such that the system remains high-risk under Article 6, or (c) changes the intended purpose of a system, including a general-purpose AI system, so that it becomes high-risk under Article 6.

Paragraph 2. The original provider ceases to be the provider of that specific system but must cooperate closely with the new provider, share necessary information and give reasonably expected technical access and assistance for conformity assessment. The duty does not apply where the original provider clearly excluded conversion of its system into a high-risk one.

Paragraph 3. For high-risk systems that are safety components of Annex I Section A products, the product manufacturer is treated as the provider and bears the Article 16 obligations.

Paragraph 4 (the limb that matters here). Providers of high-risk systems and third parties supplying AI models, tools, services, components or processes used in or integrated into a high-risk system must agree in writing on the information, capabilities, technical access and other assistance needed for the provider to comply fully, based on the generally acknowledged state of the art. **The paragraph does not apply to publicly accessible tools, services, processes or components - other than general-purpose AI models - released under a free and open-source licence.** The AI Office may publish voluntary model contract terms.

Paragraph 5. Paragraphs 2 and 3 do not override protection of intellectual property, confidential business information and trade secrets.

Scope. Every operative provision of Article 25 is tied to high-risk AI systems; paragraph 1(c) reaches general-purpose AI systems only once they become high-risk under Article 6. The article sets no obligations for systems that are not high-risk. The Annex III

high-risk obligations were moved to 2 December 2027 by Regulation (EU) 2026/1744, in force 27 July 2026.

Authorization-Grounded Workflow Attacks on Tool-Using Agents (When Approval Becomes Execution)

PRIMARY

Publisher arXiv
Published 2026-10-16
Retrieved 2026-10-19
Cite key arxiv2026authoriza
URL <https://arxiv.org/abs/2610.23077>

Assessment. Primary. Independent corroboration from the day after, approached from the authorization side rather than the filesystem side. Used for context, not for the edition's headline numbers.

Claims drawn from this source. CLAIM-2026-10-19-12

Abstract (verbatim from the work)

Tool-using language-model agents increasingly rely on local services for document access, approval management, and enterprise workflow execution. This paper studies an authorization-grounded workflow attack in the OpenClaw Agent Security Attack Competition, a controlled benchmark derived from SeClaw-style security task synthesis. Rather than using explicit policy-override instructions, the attack guides the agent through a locally coherent approval workflow: a protected document request is denied, the agent requests a normal approval token from the local approval service, and the returned token is supplied to the document service through `authorized_by`. Official submission traces show this pattern succeeding on three Enterprise OA document targets. We also conducted 23 controlled local validation experiments in the official OpenClaw execution environment: the original configuration reached the approval-token transition in 9/9 reruns and completed the exact target operation in 8/9 reruns, while single-component variants remained successful on the three document targets. The results provide concrete case evidence of an authorization-context gap: token possession can be treated as sufficient authorization even when requester identity, approver authority, resource scope, action type, destination or recipient, and business purpose are not jointly enforced. We present the attack pattern, controlled trace analysis, and defensive implications for context-bound tokens, service-side policy binding, agent-side authorization checks, and trace-aware benchmark design.

Authors: Chenggong Shi; Hongmei Mou. arXiv:2610.23077v1, 16 Oct 2026.

Note on scale. 23 controlled local validations in one competition environment is a case study, not a population estimate. Cited in this edition only as independent corroboration that the authorization context, rather than the tool contract, is where the gap sits.