mcp-proof DELIVERY REPORT

secure-filesystem-server

✓ **SHIP-READY** — all MUST checks pass, no blocking security findings, no behavioural drift

Server under test	`npx -y @modelcontextprotocol/server-filesystem /tmp/mcp-proof-sandbox`
Generated	2026-08-21 19:50 UTC · mcp-proof v0.7.2
Protocol	negotiated `2025-11-25` · initialize-handshake era · newest initialize-handshake revision
Behaviour fingerprint	`sha256:66fc4a2a430dc1662970e2864ebffea8cadd628c4d59622b468ca17fb73cb67b`

CONFORMANCE

11/11

MUST checks passed · 7/7 SHOULD

SECURITY

1

finding · 0 blocking · 1 advisory

BEHAVIOUR REGRESSION

34/34

replays clean · gate PASS

Evidence scope. This report proves what was observed on the wire: protocol conformance of the served surface, static analysis of advertised tool metadata, and behavioural equality against the recorded fixture baseline. It does not assess deployment, source code, process controls or authorization/OAuth flows — MSSS controls that need such evidence are marked *manual review* (or *partial*), never assumed.

Protocol conformance

CHECK	LEVEL	RESULT	DETAILS
`LIFE-01`	MUST	✓ PASS	**initialize returns protocolVersion, capabilities and serverInfo** protocolVersion, capabilities and serverInfo all present
`LIFE-02`	SHOULD	✓ PASS	**server negotiates the newest handshake revision (2025-11-25)** negotiated 2025-11-25, the newest revision the initialize handshake carries
`LIFE-03`	MUST	✓ PASS	**server answers tools/list after the initialized notification** tools/list returned 14 tool(s) across 1 page(s) after initialized
`LIST-01`	MUST	✓ PASS	**tools/list pagination terminates (no cursor loop)** single page, no pagination cursor
`RPC-01`	MUST	✓ PASS	**unknown method gets a JSON-RPC error response** unknown method rejected with error code -32601

CHECK	LEVEL	RESULT	DETAILS
RPC-02	SHOULD	✓ PASS	**unknown method error code is -32601 (method not found)** expected -32601, got -32601
RPC-03	MUST	✓ PASS	**malformed tools/call params are rejected with an error** malformed params rejected with error code -32603
TOOL-01	MUST	✓ PASS	**every tool has a non-empty name and an inputSchema** all 14 tools have a name and an inputSchema
TOOL-02	SHOULD	✓ PASS	**every tool has a non-empty description** all 14 tools carry a description
TOOL-03	MUST	✓ PASS	**every tool inputSchema compiles as JSON Schema** all 14 inputSchemas compile as JSON Schema draft 2020-12
TOOL-04	MUST	✓ PASS	**calling a nonexistent tool is rejected** rejected as tool result with isError=true
TOOL-05	MUST	✓ PASS	**a call missing required arguments is rejected** read_file with empty args rejected via isError=true
TOOL-06	MUST	✓ PASS	**declared outputSchemas compile as JSON Schema** all 14 declared outputSchema(s) compile
TOOL-08	MUST	– SKIP	**observed structuredContent matches the declared outputSchema** runtime behaviour unobserved — read_file: call yielded no normal result; read_text_file: call yielded no normal result; read_media_file: call yielded no normal result
TOOL-07	SHOULD	✓ PASS	**declared input constraints are enforced** 6 schema-violating input(s) across 3 tool(s), all rejected
RES-01	MUST	– SKIP	**advertised resources capability serves resources/list** resources capability not advertised
RES-02	MUST	– SKIP	**every resource carries a uri and a name** resources/list unavailable
RES-03	MUST	– SKIP	**resources/read returns contents for an advertised resource** no listed resource to read
RES-04	MUST	– SKIP	**resources/list pagination terminates (no cursor loop)** no surface to paginate
PROMPT-01	MUST	– SKIP	**advertised prompts capability serves prompts/list** prompts capability not advertised

CHECK	LEVEL	RESULT	DETAILS
PROMPT-02	MUST	– SKIP	**every prompt has a name and well-formed argument metadata** prompts/list unavailable
PROMPT-03	MUST	– SKIP	**prompts/get rejects a call missing required arguments** no prompt declares required arguments
PROMPT-04	MUST	– SKIP	**prompts/list pagination terminates (no cursor loop)** no surface to paginate
CAP-02	SHOULD	✓ PASS	**declared capabilities match served features (resources)** capabilities.resources not declared and resources/list not served
CAP-03	SHOULD	✓ PASS	**declared capabilities match served features (prompts)** capabilities.prompts not declared and prompts/list not served
HYG-01	MUST	✓ PASS	**stdout carries only JSON-RPC messages** no non-JSON-RPC stdout lines observed
CAP-01	SHOULD	✓ PASS	**declared capabilities match served features (tools)** capabilities.tools declared and tools/list served

Security & hygiene

CHECK	DOMAIN	RESULT	DETAILS
SEC-01	MCP-INPUT-01	✓ PASS	**no prompt-injection patterns in tool descriptions** 0 matches across 14 tools
SEC-02	MCP-INPUT-01	✓ PASS	**no invisible or bidi control characters in tool metadata** 0 invisible characters across 14 tools
SEC-03	MCP-LOG-02	✓ PASS	**no secret-looking strings in tool metadata** 0 secret-like strings across 14 tools
SEC-04	MCP-INPUT-02	! WARN	**injection-surface string params carry constraints** unconstrained injection-surface params: read_file.path, read_text_file.path, read_media_file.path, write_file.path, edit_file.path, create_directory.path, list_directory.path, list_directory_with_sizes.path, directory_tree.path, search_files.path, get_file_info.path **Fix:** Add enum, pattern or maxLength to path/url/command-like string params.

CHECK	DOMAIN	RESULT	DETAILS
SEC-05	MCP-INPUT-01	✓ PASS	**tool descriptions stay under 2000 chars** longest description 457 chars across 14 tools
SEC-06	MCP-EXEC-01,MCP-EXEC-02	✓ PASS	**no tool advertises unconstrained arbitrary execution** 0 exec-style tools with free-form params across 14 tools

MSSS compliance

L1: 1/2 auto-assessable controls met · 1 partial · 4 require manual review. Mapped against MSSS v0.1 (control-level mapping v2.0 (2026-01-20)): 3 of 24 controls are auto-assessable from this audit's deterministic checks (partial = evidence ran clean but cannot prove the control on its own); the remaining 21 need deployment, code or process evidence and are marked manual review — never assessed by this tool.

CONTROL	DOMAIN	LEVEL	RESULT	TITLE
MCP-EXEC-01	Execution	L1	– manual review	**Prohibition of Shell Execution** SEC-06 flags advertised arbitrary-exec tools, but proving the absence of shell invocation requires source review.
MCP-FS-01	Filesystem	L1	– manual review	**Path Allowlisting and Canonical Resolution**
MCP-FS-02	Filesystem	L1	– manual review	**Symlink Resolution Validation**
MCP-NET-01	Network	L1	– manual review	**URL Validation and SSRF Mitigation**
MCP-INPUT-01	Input Validation	L1	✓ met	**JSON Schema Validation** **Evidence checks:** TOOL-01, TOOL-03, RPC-03, TOOL-05, SEC-01, SEC-02, SEC-05 Auto-assessed from advertised schemas (present and valid), live rejection probes, and the tool-metadata poisoning scan.
MCP-LOG-02	Logging	L1	◐ partial	**Secret Redaction in Logs** **Evidence checks:** SEC-03 Supporting evidence only: SEC-03 scans advertised tool metadata for secret-like strings; server log output is not inspected, so a clean scan cannot prove log redaction (a leak still proves a gap).
MCP-SUPPLY-02	Supply Chain	L2	– manual review	**Trusted Package Sources**

CONTROL	DOMAIN	LEVEL	RESULT	TITLE
MCP-INPUT-02	Input Validation	L2	◐ partial	**Input Bounds Enforcement** **Evidence checks:** SEC-04 SEC-04: unconstrained injection-surface params: read_file.path, read_text_file.path, read_media_file.path, write_file.path, edit_file.path, create_directory.path, list_directory.path, list_directory_with_sizes.path, directory_tree.path, search_files.path, get_file_info.path Auto-assessed from advertised schema constraints (enum/pattern/maxLength); runtime payload limits are not probed.
MCP-INPUT-03	Input Validation	L2	– manual review	**Timeout Enforcement**
MCP-NET-03	Network	L2	– manual review	**TLS Enforcement** MSSS marks this N/A for stdio-only deployments with no network access.
MCP-EXEC-02	Execution	L2	– manual review	**Command Allowlisting** See the SEC-06 advisory for advertised command tools; allowlist verification requires source review.
MCP-EXEC-03	Execution	L2	– manual review	**Argument Separator Usage**
MCP-AUTHZ-01	Authorization	L3	– manual review	**OAuth Token Delegation** MSSS marks this N/A for stdio transport with OS-level user isolation.
MCP-AUTHZ-02	Authorization	L3	– manual review	**Per-Tool Scope Definition**
MCP-AUTHZ-03	Authorization	L3	– manual review	**Least Privilege Tool Design**
MCP-AUTHZ-04	Authorization	L3	– manual review	**Resource-Based Access Control**
MCP-LOG-01	Logging	L3	– manual review	**Comprehensive Audit Logging**
MCP-DEPLOY-01	Deployment	L3	– manual review	**Container Hardening**
MCP-FS-03	Filesystem	L4	– manual review	**Filesystem Sandboxing**
MCP-SUPPLY-01	Supply Chain	L4	– manual review	**Package Integrity Verification**

CONTROL	DOMAIN	LEVEL	RESULT	TITLE
MCP-DEPLOY-03	Deployment	L4	– manual review	**Resource Limits and Rate Limiting**
MCP-NET-02	Network	L4	– manual review	**Egress Traffic Filtering**
MCP-DEPLOY-02	Deployment	L4	– manual review	**System Call Filtering (seccomp/AppArmor)**
MCP-DEPLOY-04	Deployment	L4	– manual review	**Runtime Integrity Monitoring** Future control in MSSS v0.1 — implementation details TBD upstream.

Control taxonomy from the MCP Server Security Standard (MSSS), CC BY-SA 4.0, mcp-security-standard.org.

Behaviour regression

FIXTURE	TOOL	VERDICT	DETAIL
0001__read_file__e8cf9eda.json	read_file	OK	
0002__read_file__0c5f5c79.json	read_file	OK	
0003__read_file__fe0dcdc6.json	read_file	OK	
0004__read_file__518341a7.json	read_file	OK	
0005__read_text_file__e8cf9eda.json	read_text_file	OK	
0006__read_text_file__0c5f5c79.json	read_text_file	OK	
0007__read_text_file__fe0dcdc6.json	read_text_file	OK	
0008__read_text_file__518341a7.json	read_text_file	OK	
0009__read_media_file__e8cf9eda.json	read_media_file	OK	
0010__read_media_file__0c5f5c79.json	read_media_file	OK	
0011__read_media_file__fe0dcdc6.json	read_media_file	OK	
0012__read_media_file__518341a7.json	read_media_file	OK	
0013__read_multiple_files__dfea4edf.json	read_multiple_files	OK	

FIXTURE	TOOL	VERDICT	DETAIL
0014__list_directory__e8cf9eda.json	list_directory	OK	
0015__list_directory__0c5f5c79.json	list_directory	OK	
0016__list_directory__fe0dcdc6.json	list_directory	OK	
0017__list_directory__518341a7.json	list_directory	OK	
0018__list_directory_with_sizes__e8cf9eda.json	list_directory_with_sizes	OK	
0019__list_directory_with_sizes__0c5f5c79.json	list_directory_with_sizes	OK	
0020__list_directory_with_sizes__fe0dcdc6.json	list_directory_with_sizes	OK	
0021__list_directory_with_sizes__518341a7.json	list_directory_with_sizes	OK	
0022__directory_tree__e8cf9eda.json	directory_tree	OK	
0023__directory_tree__0c5f5c79.json	directory_tree	OK	
0024__directory_tree__fe0dcdc6.json	directory_tree	OK	
0025__directory_tree__518341a7.json	directory_tree	OK	
0026__search_files__580853a9.json	search_files	OK	
0027__search_files__83e4c665.json	search_files	OK	
0028__search_files__2831e0f1.json	search_files	OK	
0029__search_files__ee546753.json	search_files	OK	
0030__get_file_info__e8cf9eda.json	get_file_info	OK	
0031__get_file_info__0c5f5c79.json	get_file_info	OK	
0032__get_file_info__fe0dcdc6.json	get_file_info	OK	
0033__get_file_info__518341a7.json	get_file_info	OK	
0034__list_allowed_directories__44136fa3.json	list_allowed_directories	OK	

Leave-behind: your regression gate

The recorded fixtures (`demo/fixtures-filesystem`, suite fingerprint `sha256:18cb213983f09d0acd213e66dec2243f42caa915b372f9a9df7c6eecddbd3e2b`) are the behavioural contract of this server. Keep them in the repository and run the workflow below in CI — any breaking or value drift fails the build before it reaches your users.

```
# The recorded fixtures in demo/fixtures-filesystem are the behavioral contract: any drift fails thi
name: mcp-proof regression gate
on:
  push:
  pull_request:
jobs:
  replay:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v5
        with:
          python-version: "3.11"
      - name: Install mcp-proof
        run: pip install git+https://github.com/YuCPbit/mcp-proof
      - name: Replay golden fixtures against the live server
        run: mcp-proof replay --fixtures demo/fixtures-filesystem -- npx -y @modelcontextprotocol/se
```

Recommended next steps

P1 SEC-04 — Add enum, pattern or maxLength to path/url/command-like string params.

Reproducible audit — the behaviour fingerprint covers every check verdict and behavioural replay verdict, never timestamps or latency measurements (those stay outside both fingerprints); the launch command, evidence text and the auditor's version live in the JSON model's separate `run_hash`, which seals the whole document. Re-run with identical server behaviour and the hash matches: `sha256:66fc4a2a430dc1662970e2864ebffea8cadd628c4d59622b468ca17fb73cb67b`
Generated by mcp-proof v0.7.2 · deterministic lanes only, no LLM judging · MCP spec reference 2025-11-25 (initialize-handshake era)