

BruceJqs / public\_exp Public[Code](#) [Issues 34](#) [Pull requests](#) [Actions](#) [Projects](#) [Security and quality](#)[New issue](#)

# Path Traversal Vulnerability in mcp-rtfm #35

[Open](#)

BruceJqs opened 2 weeks ago

[Owner](#)

## Path Traversal Vulnerability in mcp-rtfm

### 1) CNA / Submission Type

- Submission type: Report a vulnerability (CVE ID request)
- Reporter role: Independent security researcher
- Report date: Apr 18, 2026

### 2) Reporter Contact

- Reporter name: BruceJin
- Reporter email: brucejin@zju.edu.cn
- Permission to share contact with vendor: Yes

### 3) Vendor / Product Identification

- Vendor: ryanjoachim
- Product: mcp-rtfm
- Repository: <https://github.com/ryanjoachim/mcp-rtfm>
- Affected component(s):
- src/index.ts

### 4) Vulnerability Type

- CWE: CWE-22 (Improper Limitation of a Pathname to a Restricted Directory)

- Short title: Path traversal in MCP documentation file handling

## 5) Affected Versions

---

- Confirmed affected: 0.1.0, commit `054fe515735cb477d4640c20930c04b243e443fc`
- Suspected affected range: revisions containing the same request-to-sink flows listed below
- Fixed version: Not available at time of report

## 6) Vulnerability Description

---

A path traversal vulnerability (CWE-22) has been identified in mcp-rtfm version 0.1.0, specifically within the `get_doc_content`, `read_doc`, and `update_doc` MCP tools. The tools construct filesystem paths by interpolating a user-supplied `docFile` value into a string without normalization or boundary checks, allowing `../` sequences to escape the intended `.handoff_docs` directory. An attacker with network access to the MCP interface can read or modify arbitrary files accessible to the server process, leading to data exposure, integrity loss, and potential service disruption. No fixed version is available at the time of reporting.

## 7) Technical Root Cause

---

1. `js/file-access-from-request`
  - Source: `src/index.ts:442` ( `docFile` for documentation read tools )
  - Source: `src/index.ts:456` ( `projectPath` for `update_doc` )
  - Source: `src/index.ts:460` ( `docFile` for `update_doc` )
  - Source: `src/index.ts:486` ( `projectPath` for `get_doc_content` )
  - Source: `src/index.ts:490` ( `docFile` for `get_doc_content` )
  - Read sink: `src/index.ts:930` ( `read_doc` )
  - Write sink: `src/index.ts:985` ( `update_doc` )
  - Read sink: `src/index.ts:1038` ( `get_doc_content` )
  - Sink path construction: `const filePath = `${projectPath}/.handoff_docs/${docFile}`;`

## 8) Attack Prerequisites

---

- Attacker can invoke the MCP tools `get_doc_content`, `read_doc`, and/or `update_doc`.
- The target path is readable or writable by the MCP server process.
- For `update_doc`, the attacker must first call `read_doc` on the same `docFile` value and provide `searchContent` that exists in the previously read content.
- No runtime policy normalizes `docFile`, rejects `..` segments, or verifies the resolved path remains inside `.handoff_docs`.

## 9) Proof of Concept / Reproduction Guidance

This proof of concept provides a concise, CVE-style reproduction example for the reported issue.

### 1. Reproduction request

```
{"jsonrpc": "2.0", "id": 1, "method": "tools/call", "params": {"name": "get_doc_content", "argum  ":
```

### 2. Validation

- Ensure `/tmp/mcp-rtfm-poc/project/.handoff_docs` exists so that traversal is evaluated from an existing directory.
- Invoke `get_doc_content` with a traversal payload such as `../../../../etc/hosts`.
- Confirm that the `mcp-inspector` response contains the contents of `/etc/hosts`.
- For write verification, create a controlled file outside `.handoff_docs`, call `read_doc` with a traversal payload targeting that file, then call `update_doc` with the same `projectPath` and `docFile`, a matching `searchContent`, and a replacement value.
- Confirm that the controlled file outside `.handoff_docs` is modified.
- The reproduction has been manually confirmed with `mcp-inspector` for `/etc/hosts` read and for `read_doc` plus `update_doc` path traversal write behavior.

## 10) Security Impact

- Confidentiality: High (an attacker can read process-accessible files outside the intended `.handoff_docs` directory).
- Integrity: High (an attacker can modify process-writable files outside the intended `.handoff_docs` directory through `read_doc` plus `update_doc`).
- Availability: High (an attacker may corrupt writable files or application state, depending on process privileges and target path).
- Scope: Unchanged.

## 11) CVSS v3.1 Suggestion

- Suggested vector: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H`
- Suggested base score: 7.8 (High)
- Adjust `AV` to `N` if the affected MCP tools are exposed through a remotely reachable MCP bridge or service.

## 12) Workarounds / Mitigations

- Do not expose the MCP server to untrusted clients until a fix is available.

- Restrict access to documentation read/update tools to trusted local users only.
- Run the MCP server with a dedicated low-privilege OS account.
- Configure filesystem permissions so the MCP process cannot read or write sensitive files.
- Monitor unexpected reads or writes outside project `.handoff_docs` directories.

## 13) Recommended Fix

---

- Resolve `projectPath` and `docFile` with filesystem-safe APIs instead of string interpolation.
- Reject absolute `docFile` values, path separators where not expected, `..` segments, null bytes, and symlink escapes.
- Resolve the final path and verify it remains inside the intended `${projectPath}/.handoff_docs` directory before every read or write.
- Consider accepting only a fixed allowlist of documentation filenames such as the expected default documentation files.
- Apply the same path validation to `read_doc`, `get_doc_content`, `update_doc`, metadata updates, related-document lookups, and any other filesystem tools.
- Add regression tests proving that traversal payloads cannot read or write outside `.handoff_docs`.
- Publish a maintainer security advisory once a patch is released.

## 14) References

---

- Repository: <https://github.com/ryanjoachim/mcp-rtfm>
- Reviewed source file: `src/index.ts`
- CWE-22: <https://cwe.mitre.org/data/definitions/22.html>

## 15) Credits

---

- Discoverer: `BruceJin`
- Discovery method: Static analysis (CodeQL), repository source-code audit, and manual reproduction with `mcp-inspector`

## 16) Additional Notes for Form Mapping

---

- Audit verdict: Manually reproduced: attacker-controlled MCP `docFile` reaches filesystem read and write sinks without path traversal protection.
- Dynamic exploit replay status: completed with `/etc/hosts` read through `get_doc_content`; completed with `read_doc` followed by `update_doc` to modify a traversal-selected file outside `.handoff_docs`.
- Maintainer should validate release mapping before coordinated disclosure.



BruceJqs 2 weeks ago

Owner

Author



**Tools** Q

List Tools

Clear

- analyze\_existing\_docs**  
Analyze existing documentation files with enhanced content analysis and metadata generation
- analyze\_project\_with\_metadata**  
Analyze project structure, create initial documentation files, and enhance with metadata/context
- analyze\_project**  
Analyze project structure and create initial documentation files
- read\_doc**  
Read a documentation file (required before updating)
- update\_doc**  
Update a specific documentation file using diff-based changes
- get\_doc\_content**  
Get the current content of a documentation file

**get\_doc\_content**

Get the current content of a documentation file

Read-only  Destructive  Idempotent  Open-world

projectPath \*

/tmp/mcp-rtfm-lab/project

docFile \*

../etc/hosts

**Tool-specific Metadata:** Add Pair

No metadata pairs.

**Tool Result: Success**

```
###  
# Host Database  
#  
# localhost is used to configure the loopback interface  
# when the system is booting. Do not change this entry.  
##  
127.0.0.1 localhost  
255.255.255.255 broadcasthost  
::1 localhost  
  
# This line is auto added by aTrustAgent, do not modify, or aTrustAgent may unable to work  
127.0.0.1 localhost.sangfor.com.cn  
  
## Adobe Creative Cloud WAM - Start ##  
166.117.29.222 detect-ccd.creativecloud.adobe.com  
## Adobe Creative Cloud WAM - End ##  
10.211.55.16 bruce-s-windows11.shared bruce-s-windows11 #prl_hostonly shared  
"
```

By exploiting vulnerability in get\_doc\_content, the file /etc/hosts has been successfully read.

**Tools** Q

List Tools

Clear

- analyze\_existing\_docs**  
Analyze existing documentation files with enhanced content analysis and metadata generation
- analyze\_project\_with\_metadata**  
Analyze project structure, create initial documentation files, and enhance with metadata/context
- analyze\_project**  
Analyze project structure and create initial documentation files
- read\_doc**  
Read a documentation file (required before updating)
- update\_doc**  
Update a specific documentation file using diff-based changes
- get\_doc\_content**  
Get the current content of a documentation file

**read\_doc**

Read a documentation file (required before updating)

Read-only  Destructive  Idempotent  Open-world

projectPath \*

/tmp/mcp-rtfm-lab/project

docFile \*

../secret.txt

**Tool-specific Metadata:** Add Pair

No metadata pairs.

**Tool Result: Success**

```
"RTFM_SECRET_POC  
"
```

**Tools**

List Tools

Clear

- analyze\_existing\_docs  
Analyze existing documentation files with enhanced content analysis and metadata generation
- analyze\_project\_with\_metadata  
Analyze project structure, create initial documentation files, and enhance with metadata/context
- analyze\_project  
Analyze project structure and create initial documentation files
- read\_doc  
Read a documentation file (required before updating)
- update\_doc  
Update a specific documentation file using diff-based changes
- get\_doc\_content  
Get the current content of a documentation file

**update\_doc**

Update a specific documentation file using diff-based changes

Read-only
  Destructive
  Idempotent
  Open-world

projectPath \*

/tmp/mcp-rtfm-lab/project

docFile \*

../secret.txt

searchContent \*

RTFM\_SECRET\_POC

replaceContent \*

RTFM\_MODIFIED\_POC

continueToNext

Whether to continue to the next file after this update

**Tool-specific Metadata:** Add Pair

No metadata pairs.

Run Tool

Copy Input

**Tool Result: Success**

```

{
  message: "Documentation updated successfully"
  file: "../secret.txt"
  completedFiles: [
    {
      file: "../secret.txt"
    }
  ]
  nextFile: "../secret.txt"
  diff: {
    from: "RTFM_SECRET_POC"
    to: "RTFM_MODIFIED_POC"
  }
}

```

```

> printf 'RTFM_SECRET_POC\n' > /tmp/mcp-rtfm-lab/secret.txt
> cat /tmp/mcp-rtfm-lab/secret.txt
File: /tmp/mcp-rtfm-lab/secret.txt
1 RTFM_MODIFIED_POC

```

By exploiting vulnerability in read\_doc and update\_doc, the secret file has been successfully modified.

- BruceJqs** mentioned this [2 weeks ago](#)
- [Path Traversal Vulnerability in mcp-rtfm ryanjoachim/mcp-rtfm#5](#)

Sign up for free to join this conversation on [GitHub](#). Already have an account? [Sign in to comment](#)

### Metadata

### Assignees

No one assigned

### Labels

No labels

---

### Projects

No projects

---

### Milestone

No milestone

---

### Relationships

None yet

---

### Development

No branches or pull requests

---

### Participants

