New issue

# Command Injection Vulnerability in mcp-server-rijksmuseum #9

Open

BruceJqs opened 2 weeks ago

## Command Injection Vulnerability in mcp-server-rijksmuseum

### 1) CNA / Submission Type

- Submission type: Report a vulnerability (CVE ID request)
- Reporter role: Independent security researcher
- Report date: Apr 17, 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: r-huijts
- Product: mcp-server-rijksmuseum
- Repository: https://github.com/r-huijts/rijksmuseum-mcp
- Affected component(s):
- `src/index.ts`
- `src/handlers/ToolHandler.ts`

- `src/utils/typeGuards.ts`
- `src/utils/SystemIntegration.ts`

## 4) Vulnerability Type

- CWE: CWE-78 (Improper Neutralization of Special Elements used in an OS Command)
- Short title: Command injection in MCP `open_image_in_browser` URL handling

## 5) Affected Versions

- Confirmed affected: 1.0.4, commit `af9c2a2dba1e709f2193a59bb3fee3a3b66380b5`
- 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 command injection vulnerability (CWE-78) has been identified in mcp-server-rijksmuseum version 1.0.4, specifically within the open_image_in_browser MCP tool. The tool accepts a user-supplied imageUrl argument, performs only a basic type check, and passes it unsanitized into a shell command string executed via child_process.exec. An attacker with network access to the MCP interface can inject shell metacharacters through the imageUrl parameter (e.g., ''; id #) to execute arbitrary operating system commands with the privileges of the server process, leading to full host compromise, including data exposure, integrity loss, and service disruption. No fixed version is available at the time of reporting.

## 7) Technical Root Cause

1. `js/command-injection-from-request`
   - Source: `src/index.ts:291` (`open_image_in_browser` tool)
   - Source argument: `src/index.ts:296` (`imageUrl`)
   - Insufficient validation: `src/utils/typeGuards.ts:30`
   - Source-to-sink transfer: `src/handlers/ToolHandler.ts:175`
   - Sink: `src/utils/SystemIntegration.ts:12`
   - Sink code: `await this.execAsync(`${cmd} "${url}"`);`

## 8) Attack Prerequisites

- Attacker can invoke the MCP `open_image_in_browser` tool.
- The server process runs on an operating system where `open`, `xdg-open`, or `start` is available through a shell.
- No effective runtime policy strips embedded quotes or shell metacharacters from `imageUrl`.
- The injected command runs with the privileges of the MCP server process.

## 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":"open_image_in_browser",
```

2. Validation

- Start the affected MCP server with a configured `RIJKSMUSEUM_API_KEY` value, for example by connecting `mcp-inspector` to `node dist/index.js`.
- Invoke the `open_image_in_browser` tool with the `imageUrl` value shown above.
- Confirm that the `mcp-inspector` response contains output from the injected `id` command, such as `uid=... gid=...`.
- The reproduction has been manually confirmed by using `http://127.0.0.1/"; id 1>&2; exit 1; #` as the `imageUrl` value and observing the `id` command result in `mcp-inspector`.

## 10) Security Impact

- Confidentiality: High (arbitrary command execution can read files and environment variables accessible to the server process).
- Integrity: High (arbitrary command execution can modify files or application state accessible to the server process).
- Availability: High (arbitrary command execution can terminate processes, delete files, or consume system resources).
- 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 tool is 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 the `open_image_in_browser` tool to trusted local users only.
- Disable or remove browser-opening functionality in untrusted deployments.
- Run the MCP server with a dedicated low-privilege OS account.

## 13) Recommended Fix

- Replace `child_process.exec` with `child_process.execFile` or `spawn` using an argument array and `shell: false`.
- Pass the browser-opening command and URL as separate arguments instead of building a single shell command string.
- Strictly validate `imageUrl` as a URL and reject embedded quotes, control characters, shell metacharacters, and non-HTTP(S) schemes.
- Enforce an allowlist for expected Rijksmuseum image hostnames if the tool is intended to open only Rijksmuseum image URLs.
- Add regression tests proving that payloads containing `"`, `;`, `&&`, `|`, backticks, `$()`, and redirections cannot execute additional commands.
- Publish a maintainer security advisory once a patch is released.

## 14) References

- Repository: https://github.com/r-huijts/rijksmuseum-mcp
- Reviewed source files:
- `src/index.ts`
- `src/handlers/ToolHandler.ts`
- `src/utils/typeGuards.ts`
- `src/utils/SystemIntegration.ts`
- CWE-78: https://cwe.mitre.org/data/definitions/78.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 `imageUrl` reaches an OS command sink and executes injected shell commands.
- Dynamic exploit replay status: completed with injected `id` command using `http://127.0.0.1/"; id 1>&2; exit 1; #`; `mcp-inspector` displayed the `id` command result.
- Maintainer should validate release mapping before coordinated disclosure.

For furthermore information, please refer to BruceJqs/public_exp#33

## Metadata

**Assignees**

No one assigned

**Labels**

No labels

**Projects**

No projects

**Milestone**

No milestone

**Relationships**

None yet

**Development**

No branches or pull requests

**Participants**