New issue

Command Injection Vulnerability in website-downloader #7

Open

BruceJqs opened 3 weeks ago

Command Injection Vulnerability in website-downloader

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: pskill9
- Product: website-downloader
- Repository: https://github.com/pskill9/website-downloader
- Affected component(s):
- `src/index.ts`

4) Vulnerability Type

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

5) Affected Versions

- Confirmed affected: 0.1.0, commit `5b399bebad1800ac6df5052b63eaea37117092b6`
- 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 website-downloader version 0.1.0, specifically within the download_website MCP tool in src/index.ts. The tool constructs a wget command by concatenating user-supplied url and outputPath arguments into a shell command string and executes it via child_process.exec without proper escaping or argument separation. An attacker with network access to the MCP interface can inject shell metacharacters through outputPath (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:83` (`request.params.arguments`)
   - Sink: `src/index.ts:122`
   - Sink code: `const { stdout, stderr } = await execAsync(wgetCommand);`
   - Unsafe command construction:
     - `src/index.ts:110`: `const wgetCommand = [...].join(' ');`
     - `src/index.ts:118`: `'--directory-prefix=' + outputPath`
     - `src/index.ts:119`: `url`

8) Attack Prerequisites

- Attacker can invoke the MCP `download_website` tool.
- The server process can execute `wget` and shell commands through `child_process.exec`.
- No effective runtime policy strips shell metacharacters from `outputPath` or replaces shell execution with argument-vector execution.
- 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":"download_website","argu
```

2. Validation

- Build and start the affected MCP server with `npm run build` followed by `npm run inspector`, or connect `mcp-inspector` directly to `build/index.js`.
- Invoke the `download_website` tool with the arguments 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 injecting `id` 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: Changed.

11) CVSS v3.1 Suggestion

- Suggested vector: `CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H`
- Suggested base score: 9.9 (Critical)
- Adjust `PR` to `N` if the affected MCP tool is exposed to unauthenticated remote clients.

12) Workarounds / Mitigations

- Do not expose the MCP server to untrusted clients until a fix is available.
- Restrict access to the `download_website` tool to trusted local users only.
- Reject shell metacharacters in `url` and `outputPath` as a temporary defense-in-depth measure.
- Run the MCP server with a dedicated low-privilege OS account and a restricted working directory.

13) Recommended Fix

- Replace `child_process.exec` with `child_process.execFile` or `spawn` using an argument array and `shell: false`.
- Pass `wget` options as separate arguments instead of building a single shell command string.
- Validate `url` with strict scheme and hostname checks.
- Normalize and constrain `outputPath` to an intended download directory.
- Add regression tests proving that payloads such as `; id #`, `&& id`, backticks, `$()`, and embedded quotes cannot execute additional commands.
- Publish a maintainer security advisory once a patch is released.

14) References

- Repository: https://github.com/pskill9/website-downloader
- Reviewed source file: `src/index.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 tool argument reaches an OS command sink and executes injected shell commands.
- Dynamic exploit replay status: completed with injected `id` command; `mcp-inspector` displayed the `id` command result.
- Maintainer should validate release mapping before coordinated disclosure.

For furthermore information, please refer to BruceJqs/public_exp#31

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants