New issue

Command Injection Vulnerability in Vale-MCP #27

Open

BruceJqs opened 2 weeks ago

Command Injection Vulnerability in Vale-MCP

1) CNA / Submission Type

- Submission type: Report a vulnerability (CVE ID request)
- Reporter role: Independent security researcher
- Report date: March 17, 2026

2) Reporter Contact (fill before submit)

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

3) Vendor / Product Identification

- Vendor: ChrisChinchilla
- Product: vale-mcp
- Repository: https://github.com/ChrisChinchilla/Vale-MCP
- Affected component(s):
- `src/index.ts`
- `src/vale-runner.ts`

4) Vulnerability Type

- CWE: CWE-78 (OS Command Injection)
- Short title: OS command injection in MCP/HTTP request handling

5) Affected Versions

- Confirmed affected: 0.1.0
- Suspected affected range: revisions containing the same request-to-sink flows listed below
- Fixed version: Not available at time of report (March 17, 2026)

6) Vulnerability Description

A command injection vulnerability (CWE-78) has been identified in vale-mcp, specifically within the vale-runner.ts component. An attacker with network access to the MCP/HTTP interface can supply maliciously crafted input through the config_path or other tool arguments, which flows unsanitized into OS command execution via exec or execAsync. This allows arbitrary system commands to be executed with the privileges of the server process, leading to full host compromise, including data exposure, integrity loss, and potential service disruption. Versions up to and including 0.1.0 are confirmed affected.

7) Technical Root Cause

1. `js/command-injection-from-request`
   - Source: `src/index.ts:278` (`request`)
   - Sink: `src/vale-runner.ts:309`
   - Sink code: `const result = await execAsync(command, execOptions);`
2. `js/command-injection-from-request`
   - Source: `src/index.ts:278` (`request`)
   - Sink: `src/vale-runner.ts:389`
   - Sink code: `const child = exec(command, execOptions, (error, stdout, stderr) => {`

8) Attack Prerequisites

- Attacker can invoke the MCP/HTTP endpoint or tool handler that reaches the vulnerable sink.
- No effective runtime policy strips or constrains attacker-controlled values before sink usage.
- If SSRF applies: server has network egress to attacker-chosen or internal targets.

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":"check_file","arguments"
```

```
{"jsonrpc":"2.0","id":1,"method":"tools/call","params":{"name":"check_text","arguments"
```

10) Security Impact

- Confidentiality: High (host/system data exposure possible).
- Integrity: High (command execution may alter server state).
- Availability: High (service disruption via command abuse possible).
- Scope: Changed.

11) CVSS v3.1 Suggestion

- Suggested vector: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H`
- Suggested base score: 10.0 (Critical)
- Adjust `PR` upward if the vulnerable tools are strictly admin-only and strongly authenticated.

12) Workarounds / Mitigations

- Remove direct shell-string execution from request-driven paths.
- Replace free-form commands with fixed allowlists and validated argument schemas.
- Prefer argument-array process execution without shell interpretation.
- Add authentication, authorization, logging, and rate limiting on sensitive MCP/HTTP handlers.

13) Recommended Fix

- Eliminate the request-to-sink data flow documented above.
- Add input schema validation at MCP/HTTP boundaries.
- Add regression tests proving attacker-controlled values cannot reach sensitive sinks.
- Publish a maintainer security advisory once a patch is released.

14) References

- Repository: https://github.com/ChrisChinchilla/Vale-MCP
- Reviewed source file: `src/index.ts`
- Reviewed source file: `src/vale-runner.ts`
- CWE-78: https://cwe.mitre.org/data/definitions/78.html

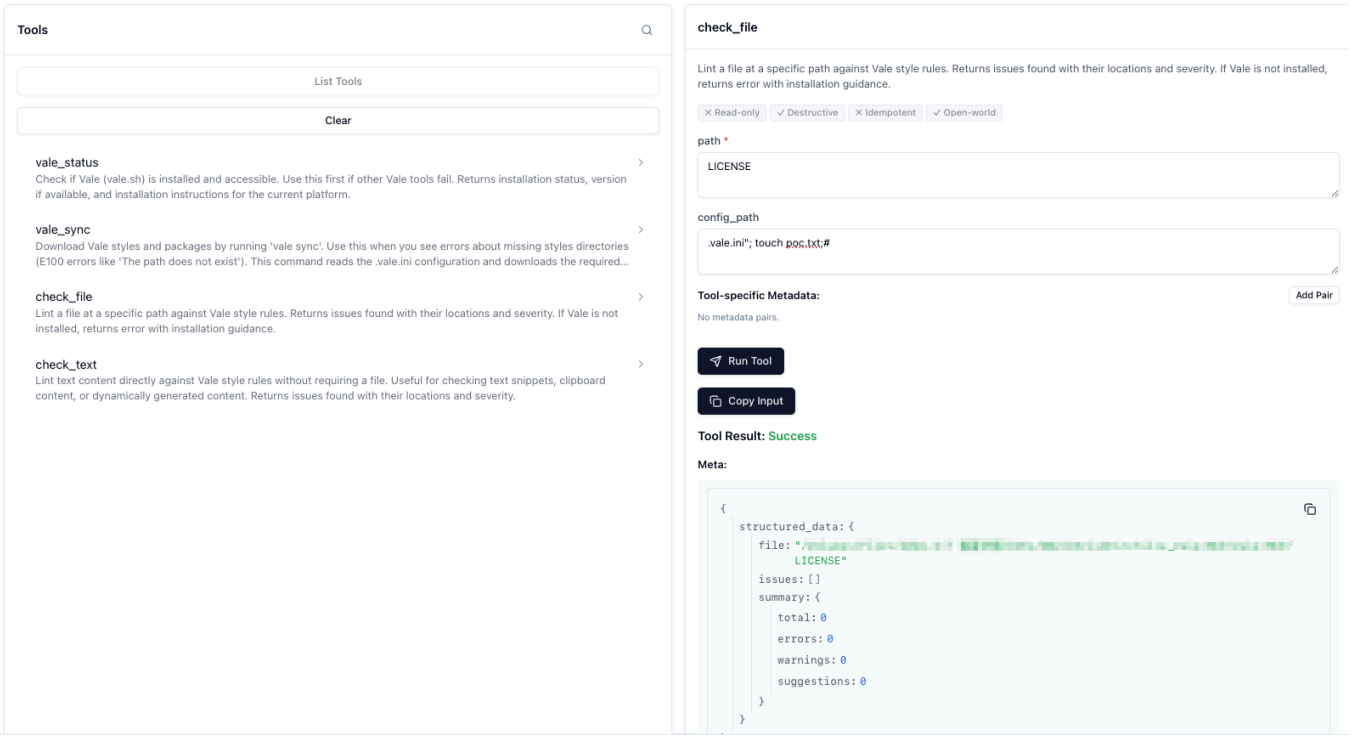
15) Credits

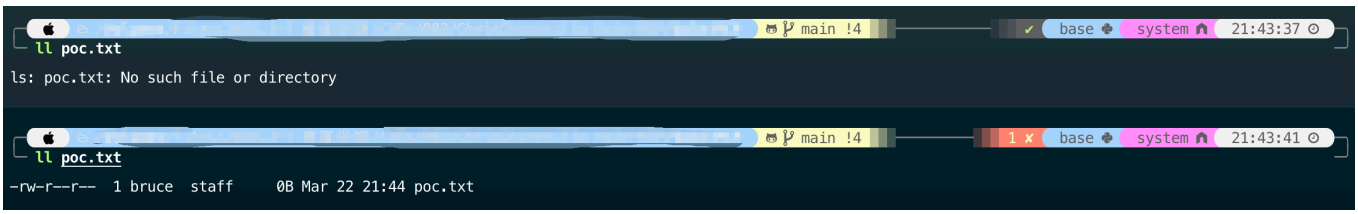
- Discoverer: `BruceJin`
- Discovery method: Static analysis (CodeQL) plus repository source-code audit

16) Additional Notes for Form Mapping

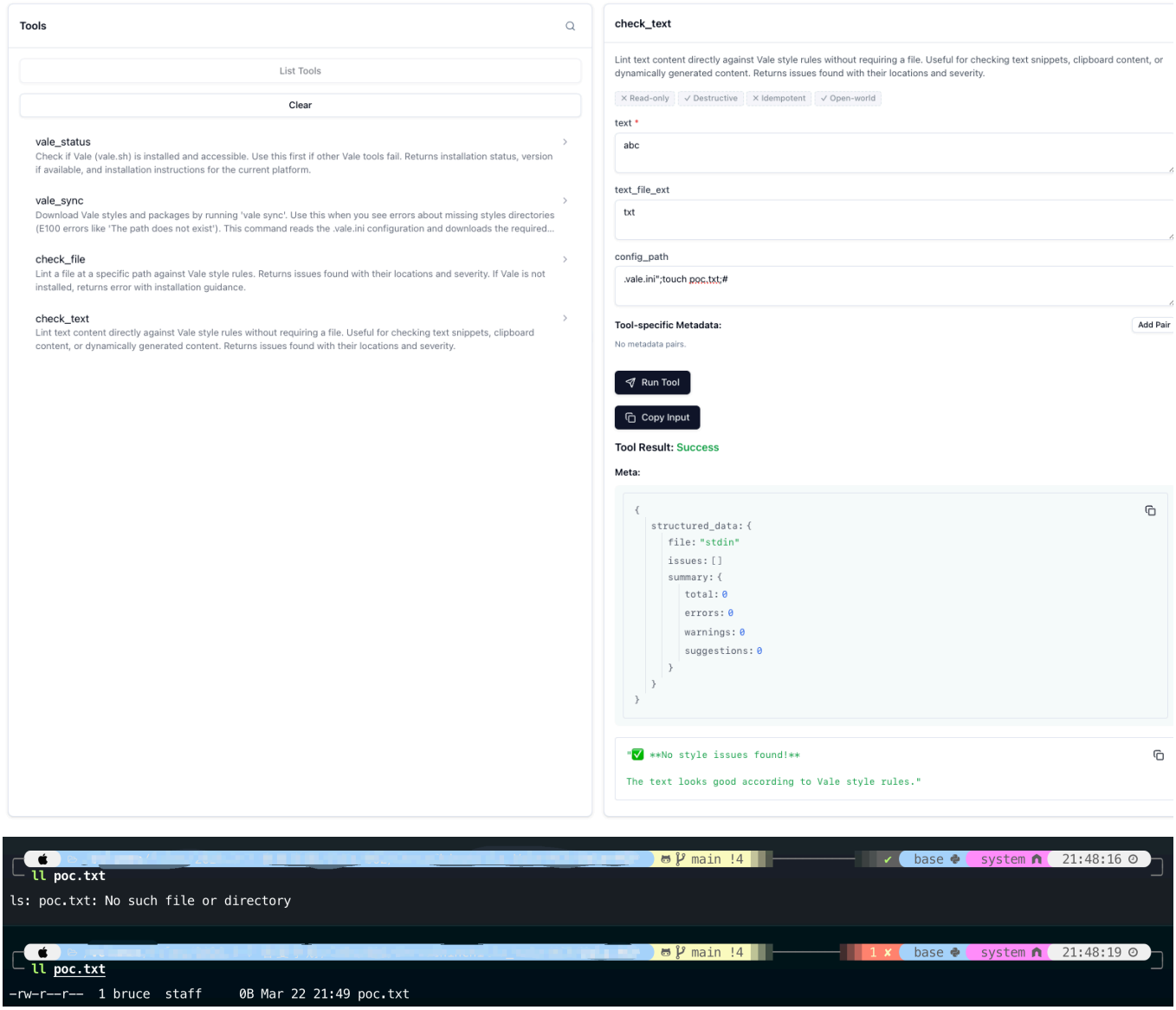
- Audit verdict: Likely exploitable: command injection path reaches OS execution sink.
- Dynamic exploit replay status: not completed in this batch run.
- Maintainer should validate release mapping before coordinated disclosure.

BruceJqs 2 weeks ago Author • • •





After executing command via the command injection vulnerability, the file "poc.txt" has been successfully created.



After executing command via the command injection vulnerability, the file "poc.txt" has been successfully created.

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

No branches or pull requests

Participants