New issue

Command Injection Vulnerability in mcp-test-runner #37

Open

BruceJqs opened 2 weeks ago Owner

Command Injection Vulnerability in mcp-test-runner

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: privsim
- Product: mcp-test-runner
- Repository: https://github.com/privsim/mcp-test-runner
- Affected component(s):
- `src/index.ts`
- Package metadata name: `@modelcontextprotocol/server-test-runner`

4) Vulnerability Type

- CWE: CWE-78 (Improper Neutralization of Special Elements used in an OS Command)
- Short title: Command injection in MCP `run_tests` command execution

5) Affected Versions

- Confirmed affected: 0.2.0, commit `83c84ed053f534774f7de935aeaa7698a5e5f9dc`
- 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-test-runner (package @modelcontextprotocol/server-test-runner) version 0.2.0, specifically within the run_tests MCP tool. The tool accepts a user-supplied command argument and, when a non-generic framework (e.g., jest, pytest) is selected, executes it via child_process.spawn with shell: true without validation or sanitization. An attacker with network access to the MCP interface can inject arbitrary shell commands into the command parameter, 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:119` (`run_tests` tool)
   - Source argument: `src/index.ts:124` (`command`)
   - Source argument: `src/index.ts:128` (`workingDir`)
   - Source argument: `src/index.ts:132` (`framework`)
   - Request argument extraction: `src/index.ts:194`
   - Generic-only validation branch: `src/index.ts:202`
   - Execution call: `src/index.ts:217`
   - Shell execution option: `src/index.ts:339`
   - Process sink: `src/index.ts:364`
   - Sink code: `const childProcess = spawn(cmd, cmdArgs, spawnOptions);`

8) Attack Prerequisites

- Attacker can invoke the MCP `run_tests` tool.
- The attacker can provide `command`, `workingDir`, and a non-`generic` `framework` value.
- The MCP server process has permission to execute shell commands in the selected working directory.

- No effective runtime policy constrains non-`generic` framework commands before `spawn` is called with `shell: true`.

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":"run_tests","arguments":{"m
```

2. Validation

- Start the affected MCP server and connect to it with `mcp-inspector`.
- Invoke the `run_tests` tool with a non-`generic` framework value such as `jest` and a command containing `id`.
- Confirm that the `mcp-inspector` response contains output from the injected `id` command, such as `uid=... gid=...`.
- The reproduction has been manually confirmed with `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 `run_tests` tool to trusted local users only.

- Disable non-`generic` framework execution or apply the same command validation to every framework value.
- Run the MCP server with a dedicated low-privilege OS account and a restricted working directory.
- Monitor unexpected commands, output directories, and test execution logs.

13) Recommended Fix

- Avoid executing caller-supplied command strings through a shell.
- Replace `shell: true` command execution with a fixed command allowlist and argument-array execution using `shell: false`.
- Apply command validation consistently to every framework, not only `framework === "generic"`.
- Restrict framework-specific command execution to known test runner binaries and safe argument patterns.
- Validate and constrain `workingDir` and `outputDir` to intended project directories.
- Add regression tests proving that payloads such as `id`, `; id`, `&& id`, `$()`, backticks, redirections, and pipes cannot execute arbitrary commands.
- Publish a maintainer security advisory once a patch is released.

14) References

- Repository: https://github.com/privsim/mcp-test-runner
- Reviewed source file: `src/index.ts`
- CWE-78: https://cwe.mitre.org/data/definitions/78.html

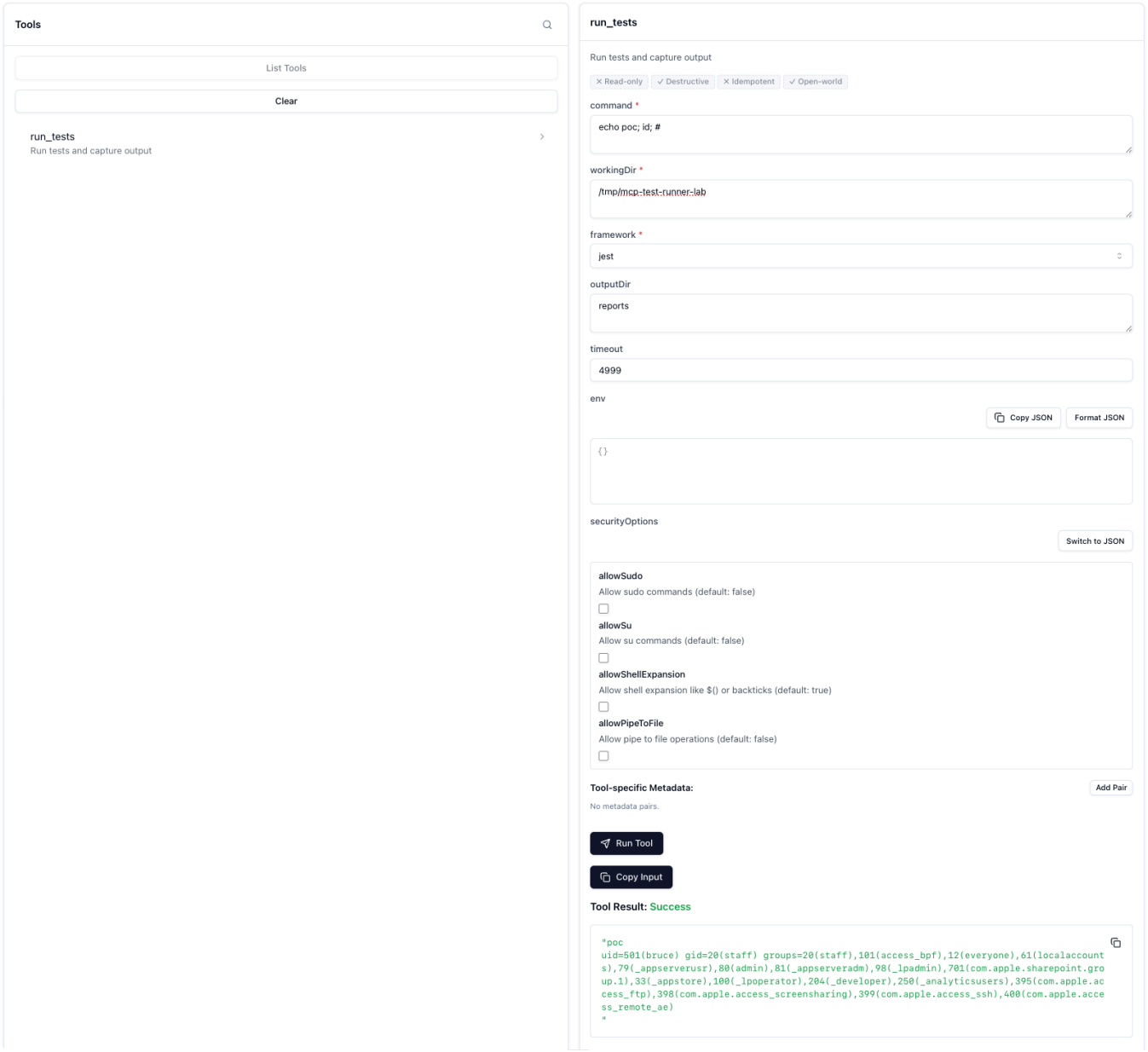
15) Credits

- 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 `command` reaches an OS command sink and executes through `spawn` with `shell: true`.
- Dynamic exploit replay status: completed with injected `id` command through `run_tests`; `mcp-inspector` displayed the `id` command result.
- Maintainer should validate release mapping before coordinated disclosure.

BruceJqs 2 weeks ago Owner Author •••



By exploiting the vulnerability, the command `id` has been successfully executed.

BruceJqs mentioned this 2 weeks ago

Command Injection Vulnerability in mcp-test-runner privsim/mcp-test-runner#24

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