New issue

# Command Injection Vulnerability in Git Integration (CWE-78) #31

Open

Labels bug

wing3e opened last month · edited by wing3e Edits

## Command Injection Vulnerability in @context-sync/server

### 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: Intina47
- Product: @context-sync/server
- Repository: https://github.com/Intina47/context-sync
- Affected component(s):

- `src/server.ts`
- `src/git-integration.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: 2.0.0
- 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 @context-sync/server, specifically within the git-integration.ts component. An attacker with network access to the MCP/HTTP interface can supply maliciously crafted input through request parameters that flow unsanitized into OS command execution sinks (e.g., git blame). 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 2.0.0 are confirmed affected.

## 7) Technical Root Cause

1. `js/command-injection-from-request`
   - Source: `src/server.ts:630` (`request`)
   - Sink: `src/git-integration.ts:380`
   - Sink code: `const output = this.exec(\` git blame --line-porcelain "${filepath}"`);`
2. `js/command-injection-from-request`
   - Source: `src/server.ts:630` (`request`)
   - Sink: `src/git-integration.ts:500`
   - Sink code: `return execSync(command, {`

## 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 repository-grounded reproduction snippet for the reported issue.

1. Git Repository Preparation

```
mkdir -p /tmp/context-sync-lab
cd /tmp/context-sync-lab
git init
git config user.email repro@example.com
git config user.name repro
printf 'hello\n' > safe.txt
git add safe.txt
git commit -m init
```

2. Set project

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

3. Reproduction request

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

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

## 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/Intina47/context-sync
- Reviewed source file: `src/server.ts`
- Reviewed source file: `src/git-integration.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.

More details: wing3e/public_exp#22

**wing3e** added **bug** last month

Intina47 2 weeks ago Owner

@**codex** review this issue and create a pr on its patch

chatgpt-codex-connector bot 2 weeks ago – with ChatGPT Codex Connector

## Summary

- Patched `GitIntegration` to stop using shell-interpreted command strings and instead execute git via `execFileSync('git', args)`, removing the command-injection sink pattern from request-driven git operations.
- Converted dynamic git calls (`diff`, `blame`, `ls-files`, file-specific `log`, etc.) to argument-array form and added `--` separators before user-influenced paths to prevent command/option injection through crafted filenames.
- Added defensive normalization for `getRecentCommits(count)` so invalid or non-positive values fall back safely to `5` rather than flowing unchecked into command construction.
- Committed changes on the current branch with message: `fix(security): prevent shell command injection in git integration` (commit `4d61400`), and created a follow-up PR via the `make_pr` tool summarizing this patch on top of the original vulnerability report.

### Testing

- ⚠️ `npm run build` (fails in this environment due to missing project dependencies/type packages such as `@types/node`, `simple-git`, and others; failure is not caused by this patch).
- ✅ `npm test`

View task →



## Metadata

**Assignees**

No one assigned

**Labels**

bug

**Projects**

No projects

**Milestone**

No milestone

**Relationships**

None yet

**Development**

No branches or pull requests

**Participants**