New issue

Arbitrary File Write Vulnerability in consciousness-explorer of sublinear-time-solver #19

Open

BruceJqs opened 2 weeks ago

Arbitrary File Write Vulnerability in consciousness-explorer of sublinear-time-solver

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: ruvnet
- Product: sublinear-time-solver / consciousness-explorer
- Repository: https://github.com/ruvnet/sublinear-time-solver
- Affected component(s):
- `src/consciousness-explorer/mcp/server.js`
- `src/consciousness-explorer/index.js`

4) Vulnerability Type

- CWE: CWE-73 (External Control of File Name or Path)
- Short title: Arbitrary file write through MCP `export_state` filepath handling

5) Affected Versions

- Confirmed affected: `sublinear-time-solver` 1.5.0, `consciousness-explorer` 1.1.1, commit `1210646955f33abe5c91f894cc7b04d024f62408`
- Suspected affected range: revisions containing the same request-to-sink flows listed below
- Fixed version: Not available at time of report

6) Vulnerability Description

An arbitrary file write vulnerability (CWE-73) has been identified in the consciousness-explorer component of sublinear-time-solver, specifically within the MCP export_state tool. The tool accepts a user-supplied filepath argument and writes JSON state data to that path using fs.writeFileSync without constraining the destination to a safe directory or validating path traversal sequences. An attacker with network access to the MCP interface can write or overwrite arbitrary files accessible to the server process, leading to integrity loss and potential service disruption. Version 1.1.1 of consciousness-explorer (commit `1210646`) is confirmed affected, and no fixed version is available at the time of reporting.

7) Technical Root Cause

1. `js/file-access-from-request`
   - Source: `src/consciousness-explorer/mcp/server.js:239` (`export_state` tool)
   - Source argument: `src/consciousness-explorer/mcp/server.js:244` (`filepath`)
   - Source-to-sink transfer: `src/consciousness-explorer/mcp/server.js:529`
   - Sink: `src/consciousness-explorer/index.js:288`
   - Sink code: `fs.writeFileSync(filepath, JSON.stringify(state, null, 2));`

8) Attack Prerequisites

- Attacker can invoke the MCP `export_state` tool.
- The MCP server process has filesystem write permissions to the attacker-selected destination.
- No effective runtime policy constrains `filepath` to a dedicated state directory or rejects absolute/traversal paths.
- The write occurs 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":"export_state","argument
```

2. Validation

- Start the `consciousness-explorer` MCP server through `mcp-inspector` without modifying repository code, for example by using `node -e` to import the existing `ConsciousnessExplorer` class and call `startMCPServer()`.
- Invoke the `export_state` tool with the arguments shown above.
- Confirm that `/tmp/sublinear_state_poc.json` is created by the MCP server process.
- Inspect the file and confirm that it contains exported JSON state data.
- The reproduction has been manually confirmed with `mcp-inspector` by writing a controlled file through the `filepath` argument.

10) Security Impact

- Confidentiality: None confirmed for the reproduced `export_state` path.
- Integrity: High (an attacker can write or overwrite files accessible to the MCP server process).
- Availability: High (an attacker may overwrite application files, user files, or other writable targets, depending on process privileges and filesystem permissions).
- Scope: Unchanged.

11) CVSS v3.1 Suggestion

- Suggested vector: `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:H`
- Suggested base score: 7.1 (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 `export_state` tool to trusted local users only.
- Run the MCP server with a dedicated low-privilege OS account and a restricted working directory.
- Configure filesystem permissions so the MCP process cannot write to sensitive locations.

13) Recommended Fix

- Do not accept arbitrary output paths for `export_state`.
- Store exported state under a dedicated application-controlled state directory.
- If callers may choose filenames, accept only a basename and reject path separators, absolute paths, `..` segments, symlinks, and reserved device names.
- Resolve the final path and verify it remains inside the intended export directory before writing.
- Use safe file creation flags where appropriate to avoid unintended overwrite of existing files.
- Add regression tests proving that absolute paths and traversal payloads cannot write outside the intended directory.
- Publish a maintainer security advisory once a patch is released.

14) References

- Repository: https://github.com/ruvnet/sublinear-time-solver
- Reviewed source files:
- `src/consciousness-explorer/mcp/server.js`
- `src/consciousness-explorer/index.js`
- CWE-73: https://cwe.mitre.org/data/definitions/73.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 `filepath` reaches a filesystem write sink and creates a file at an arbitrary process-accessible path.
- Dynamic exploit replay status: completed with `export_state` and `/tmp/sublinear_state_poc.json`; `mcp-inspector` confirmed successful tool execution and the output file was observed locally.
- Maintainer should validate release mapping before coordinated disclosure.

For furthermore information, please refer to BruceJqs/public_exp#32

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants