# Branch Monkey Command Injection via `/api/local-claude/time-machine/preview` #8

Open

wing3e opened 2 weeks ago · edited by wing3e Edits

## Branch Monkey Command Injection via `/api/local-claude/time-machine/preview`

### 1) CNA / Submission Type

- Submission type: Report a vulnerability (CVE ID request)
- Reporter role: Independent security researcher
- Report date: April 14, 2026

### 2) Reporter Contact (fill before submit)

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

### 3) Vendor / Product Identification

- Vendor: gneyal
- Product: Branch Monkey MCP / Kompany local bridge
- Repository: https://github.com/gneyal/p_69_branch_monkey_mcp.git
- Reviewed local source path: `python-batch-04/datasets/gneyal-p_69_branch_monkey_mcp`
- Exposed route: `POST /api/local-claude/time-machine/preview`
- Reviewed source file: `branch_monkey_mcp/bridge_and_local_actions/routes/advanced.py`

## 4) Vulnerability Type

- CWE: CWE-78 / CWE-88 (OS Command Injection / Argument Injection)
- Short title: The `dev_script` request field is copied into a shell command and executed with `subprocess.Popen(..., shell=True)`, allowing arbitrary command execution.

## 5) Affected Versions

- Confirmed affected: `d77e757`
- Suspected affected range: revisions containing the same `dev_script` handling in `routes/advanced.py`
- Fixed version: Not available at time of report (April 14, 2026)

## 6) Vulnerability Description

The time-machine preview feature creates a temporary worktree for a requested commit and optionally starts a dev server. If the caller provides `dev_script`, the server does:

- `command = request.dev_script.replace("{port}", str(port))`
- `subprocess.Popen(command, shell=True, cwd=str(work_path), ...)`

Because the string is executed through the system shell, metacharacters such as `;`, `&&`, `$()`, or shell redirection are interpreted. An attacker only needs to provide any valid local Git repository path and a valid commit SHA to turn this endpoint into arbitrary OS command execution.

## 7) Technical Root Cause

1. `routes/advanced.py:222-234`
   - Untrusted `dev_script` is interpolated and executed with `shell=True`.
2. `routes/advanced.py:183-216`
   - The endpoint only validates that `project_path` is a Git repository and that `commit_sha` exists; it does not restrict the command itself.
3. Additional findings note
   - The same repository contains other shell-executed developer utilities, but the direct, source-confirmed exploit documented here is the `dev_script` field on `/api/local-claude/time-machine/preview`.

## 8) Attack Prerequisites

- The attacker can reach the local bridge HTTP endpoint.
- The attacker can provide a valid `project_path` pointing to a local Git repository and a commit SHA that exists in that repository.

## 9) Proof of Concept / Reproduction Guidance

Use any existing local repository and commit on the target host. The following request causes an arbitrary file creation before starting a preview server:

```
POST /api/local-claude/time-machine/preview HTTP/1.1
Host: target-host
Content-Type: application/json

{
  "commit_sha": "<existing_commit_sha>",
  "project_path": "/path/to/local/git/repo",
  "dev_script": "touch /tmp/codex_cmdi_poc; python3 -m http.server {port}"
}
```

Expected result:

- `/tmp/codex_cmdi_poc` is created on the host.
- The server then continues executing the rest of the attacker-supplied shell command.

## 10) Security Impact

- Confidentiality: High. Arbitrary commands can read local secrets and source code.
- Integrity: High. Arbitrary commands can modify project files or system state.
- Availability: High. Arbitrary commands can kill processes, fill disks, or otherwise disrupt service.
- Scope: Unchanged.

## 11) CVSS v3.1 Suggestion

- Suggested vector: `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H`
- Suggested base score: 9.8 (Critical)

## 12) Workarounds / Mitigations

- Remove shell execution entirely for `dev_script`.
- Replace free-form commands with a small allowlist of known dev-server presets.
- If a customizable command is unavoidable, execute an argv list with strict validation and no shell.

## 13) Recommended Fix

- Replace `subprocess.Popen(command, shell=True, ...)` with a non-shell argv-based invocation.
- Treat the requested preview command as structured configuration rather than a raw shell string.
- Add regression tests covering metacharacters such as `;`, `&&`, backticks, and `$()`.

## 14) References

- Repository: https://github.com/gneyal/p_69_branch_monkey_mcp.git
- Reviewed source file: `branch_monkey_mcp/bridge_and_local_actions/routes/advanced.py`
- CWE-78: https://cwe.mitre.org/data/definitions/78.html
- CWE-88: https://cwe.mitre.org/data/definitions/88.html

## 15) Credits

- Discoverer: `Winegee`
- Discovery method: Static analysis (CodeQL) plus local source-code audit

## 16) Additional Notes for Form Mapping

- Audit verdict: Confirmed vulnerability.
- Total reviewed SARIF results for this repository/rule group: 7
- Dynamic exploit replay status: not completed in this pass.
- The separate path-injection report for this repository was re-reviewed independently during the same batch and is not required to prove the command-execution issue.



### Metadata

**Assignees**

No one assigned

**Labels**

No labels

**Projects**

No projects

**Milestone**

No milestone

**Relationships**

None yet

**Development**

No branches or pull requests

**Participants**