Sandbox Bypass via Early-Exit Logic Flaw Allows Path Traversal

High **kevincodex1** published **GHSA-m6rx-7pvw-2f73** 18 hours ago

Package

openclaude

Affected versions

v0.1.7

Patched versions

None

Description

A logic flaw exists in `bashToolHasPermission()` inside `src/tools/BashTool/bashPermissions.ts`. When the sandbox auto-allow feature is active and no explicit deny rule is configured, the function returns an `allow` result immediately — before the path constraint filter (`checkPathConstraints`) is ever evaluated. This allows commands containing path traversal sequences (e.g., `../../../../../etc/passwd`) to bypass directory restrictions entirely.

Affected Component

- **File:** `src/tools/BashTool/bashPermissions.ts`
- **Function:** `bashToolHasPermission`
- **Location:** ~line 1445 (sandbox auto-allow block)

Vulnerable Code Flow

```
bashToolHasPermission()
    │
    ├─ [~1445] Sandbox auto-allow block
    │       └─ No deny rule found → return ALLOW  ⚠️ Early exit
    │
    └─ [~1644] checkPathConstraints()             ❌ Never reached
```

The sandbox block was designed to skip interactive permission prompts in sandboxed environments. However, it unintentionally also skips the path traversal filter, which is a separate and critical security control.

Impact

Any process or user operating in a sandboxed session with no explicit deny rules can:

- Read arbitrary files outside the sandbox boundary (e.g., `/etc/passwd`, `/etc/shadow`, `.env` files, SSH private keys)
- Write to arbitrary paths (subject to OS-level permissions)
- Fully defeat the filesystem isolation that the sandbox is intended to enforce

Confidentiality: High
Integrity: High
Availability: None
CVSS v3.1: 9.1 (AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N)

Steps to Reproduce

1. Enable sandbox mode: `SandboxManager.isSandboxingEnabled() = true`
2. Enable auto-allow: `SandboxManager.isAutoAllowBashIfSandboxedEnabled() = true`
3. Ensure no explicit deny rules are configured for the session
4. Submit a bash command with a path traversal payload:

```
cat ../../../../../etc/passwd
```

5. Observe that the command receives `behavior: allow` without triggering `checkPathConstraints`

Recommended Fix

The sandbox auto-allow block should **never short-circuit the full permission pipeline**. It may suppress interactive prompts, but path constraint validation must always execute.

Option 1 — Preferred: Continue pipeline on `allow`

Only return early for `deny` or `ask` behaviors. Let `allow` fall through to `checkPathConstraints`:

```
if (
  SandboxManager.isSandboxingEnabled() &&
  SandboxManager.isAutoAllowBashIfSandboxedEnabled() &&
  shouldUseSandbox(input)
) {
  const sandboxAutoAllowResult = checkSandboxAutoAllow(
    input,
```

```
    appState.toolPermissionContext,
  );
  if (sandboxAutoAllowResult.behavior !== 'allow') {
    // Only block or prompt — never skip path checks on allow
    return sandboxAutoAllowResult;
  }
  // If 'allow', continue to checkPathConstraints below
}
```

Option 2 — Defense in depth: Run path check before returning

Run `checkPathConstraints` explicitly inside the sandbox block before returning:

```
if (sandboxAutoAllowResult.behavior !== 'passthrough') {
  const pathCheck = checkPathConstraints(input, appState.toolPermissionContext);
  if (pathCheck.behavior !== 'allow') {
    return pathCheck; // Block traversal attempts even in sandbox
  }
  return sandboxAutoAllowResult;
}
```

Option 3 — Minimal change: Move sandbox block after path check

Reorder the function so `checkPathConstraints` always runs first, and the sandbox block only handles the prompt-suppression logic afterward.

Credit: Elvin Latifli (**@Rickidevs**)

Severity

High 8.4 / 10

CVSS v3 base metrics

Attack vector	Local
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Changed
Confidentiality	High
Integrity	High
Availability	None

Learn more about base metrics

CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:N

CVE ID

CVE-2026-35570

Weaknesses

- CWE-22
- CWE-284

Credits

Rickidevs Reporter