RCE SECURITY

Stackfield Desktop App for Windows and macOS <= 1.10.1 Path Traversal Remote Code Execution

Mar 23, 2026 · By Julien Ahrens

ADVISORY INFORMATION

- **Product**: Stackfield Desktop App for Windows and macOS
- **Vendor URL**: https://www.stackfield.com
- **CWE**: Path Traversal [CWE-22]

- **Date found**: 2026-02-17
- **Date published**: 2026-03-23
- **CVSSv4 Score**: 7.1 (CVSS:4.0/AV:N/AC:L/AT:P/PR:L/UI:A/VC:N/VI:H/VA:N/SC:H/SI:H/SA:H)
- **CVE**: CVE-2026-28373

CREDITS

This vulnerability was discovered and researched by Julien Ahrens from RCE Security.

VERSIONS AFFECTED

Stackfield Desktop App for Windows and macOS version 1.10.1 and below

INTRODUCTION

All-in-One project management tool for organizations with sensitive data. Ideal for government agencies, banks, insurance companies, law firms, universities.

(from the vendor’s homepage)

VULNERABILITY DETAILS

The Stackfield Desktop Apps for Windows and macOS are vulnerable to a path traversal in the organization data export decryption workflow. When a user imports a folder containing an encrypted organization data export, attacker-controlled metadata is processed by `DecryptBackup()` and used to construct output paths for decrypted files without proper sanitization.

The relevant path is built from attacker-controlled `filePath` and `fileGuid` values before being passed to `addLocalFile()`:

```
e = i.filePath.replace(i.fileGuid, "");
p.addLocalFile(t, e);
```

`addLocalFile()` then joins this untrusted path with the export base directory and writes attacker-controlled file content:

```
this.addLocalFile = function(e, t) {
    var i = l.basename(e),        // filename from the temp file
        a = l.join(r, t);         // r = export base dir, t = attacker's path
    d.existsSync(a) || d.mkdSync(a);   // creates directories as needed
    d.copyFileSync(e, l.join(a, i));   // writes the file
}
```

By using a `filePath`/`fileGuid` pair like the following:

```
filePath = foo/foo//../../AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Start
fileGuid = foo/foo//
```

An attacker can keep chunk reads inside the expected export directory, while forcing chunk writes to resolve outside that directory. This results in arbitrary file write to attacker-chosen locations.

Because the attacker controls file contents, this can be escalated to remote code execution, for example by writing into the Windows Startup folder or environment files like `~/.zshenv` on macOS.

Successful exploitation requires the attacker and victim to share at least one room (any permission level, including the “external” role). In that case, both parties can access the room encryption key, which can be used to produce a malicious encrypted export for import by the victim.

PROOF OF CONCEPT

See our linked blog article and GitHub repo.

SOLUTION

Update to version 1.10.2

REPORT TIMELINE

- 2026-02-17: Discovery of the vulnerability
- 2026-02-27: MITRE assigns CVE-2026-28373
- 2026-03-02: Vendor contacted via established disclosure channel

- 2026-03-05: Vendor responded that the fix is introduced with version 1.10.2 of the desktop apps which has been deployed on 2026-03-03. Vendor also asks to add a clarification about the export type to the advisory wording and to “limit” the publication of details about the exploit
- 2026-03-06: Confirmed the deployed fix is working; we kindly rejected any editorial changes to the blog post
- 2026-03-23: Public Disclosure

REFERENCES

- https://www.rcesecurity.com/2026/03/stackfield-desktop-app-rce-via-path-traversal-and-arbitrary-file-write-cve-2026-28373/
- https://github.com/rcesecurity/exploits/CVE-2026-28373