New issue

Jump to bottom

TypeError: decodeComponents(...).join is not a function #345

Closed KatsuragiCSL opened this issue on Aug 24 · 12 comments

KatsuragiCSL commented on Aug 24

Version: `v7.1.1`

Crash occured when `# kun%ea%ba%5a%ba` is parsed by `queryString.parse`
This hash is valid and should be parsed correctly as `{ ' kun%ea%ba%5a%ba': null }`
For example in chrome's development tools' console the url `https://google.com# kun%ea%ba%5a%ba` can be parsed without problems:

```
const url = new URL("https://google.com#  kun%ea%ba%5a%ba");
url
```

```
URL {origin: 'https://google.com', protocol: 'https:', username: '', password: '', host: 'google.com', …}
  hash: "#%20%20kun%ea%ba%5a%ba"
  host: "google.com"
  hostname: "google.com"
  href: "https://google.com/#%20%20kun%ea%ba%5a%ba"
  origin: "https://google.com"
  password: ""
  pathname: "/"
  port: ""
  protocol: "https:"
  search: ""
```

Code to reproduce

```
const queryString = require('query-string');

const parsed = queryString.parse("#  kun%ea%ba%5a%ba");
console.log(parsed);
```

Results

```
TypeError: decodeComponents(...).join is not a function
```

sindresorhus commented on Aug 24 Owner

Must be fixed in SamVerschueren/decode-uri-component#5

chrstph-dvx mentioned this issue 4 days ago

dependency decode-uri-component is vulnerable to Denial of Service #349

Closed

wzijden commented 4 days ago

Is it an option to simply get rid of this dependency and use the built in `decodeURIComponent` instead?

Prophet32j commented 4 days ago

my advice is to ditch the vulnerable library. The issue was identified on the project in August and hasn't been fixed yet. The project hasn't had any meaningful activity on it in 3 years. Either fork and fix and use the forked repo or migrate away from it. We are seeing this as well, too.

👍 1

eddilou commented 3 days ago

> Is it an option to simply get rid of this dependency and use the built in `decodeURIComponent` instead?
> I will upset you, but even in the `decodeURIComponent` it does not work correctly

wzijden commented 3 days ago

I looked at this library and all it does is replacing `+` with a space, and only if the native `decodeURIComponent` throws an error, it attempts to parse the string itself. So anything it crashes on is something that `decodeURIComponent` also crashes on.

I guess I anyway don't understand why this is considered a vulnerability.

lexek commented 3 days ago

@wzijden apparentely, someone decided it would be brilliant idea to start filing any unexpected exceptions caused by input (which even in very far fetched cases might be inputted by remote user) in libraries as DoS vulnerabilities. Not the first I see this kind of reasoning

SamVerschueren commented 2 days ago Contributor

I'll try to get a release out today.

People see to be very friendly nowadays. Piling up my Twitter DMs because Snyk seems to mark this as high severity...

❤️ 4

eddilou commented 2 days ago

> I'll try to get a release out today.
>
> People see to be very friendly nowadays. Piling up my Twitter DMs because Snyk seems to mark this as high severity...

The problem is that this is not your vulnerability, but the vulnerability of a native function, which for some reason was shifted to you

eddilou commented 2 days ago

> I'll try to get a release out today.
>
> People see to be very friendly nowadays. Piling up my Twitter DMs because Snyk seems to mark this as high severity...

I understand, there you need another try-catch block

mahesh-kavya commented yesterday

I see that decode-uri-component is upgraded to 0.2.2, but it also has the same issue

SamVerschueren commented 22 hours ago Contributor

Have you actually tried it?

https://stackblitz.com/edit/node-fhizma?file=index.js

@sindresorhus I think you can close this issue now.

SamVerschueren commented 22 hours ago Contributor

Regarding the Snyk "vulnerability". I'm in contact with them to see what needs to happen but that is unrelated to this issue.

sindresorhus closed this as completed 22 hours ago

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

8 participants