

New issue

Jump to bottom

# Stored XSS in slide mode (via reveal-markdown) #1648

Closed TobiasHoll opened this issue on Jan 13, 2021 · 0 comments · Fixed by #1650

TobiasHoll commented on Jan 13, 2021

There is a (quite convoluted) stored XSS vulnerability in the slides feature:

- The presentation's YAML options allow specifying JavaScript files as dependencies. Any JS file specified by the `dependency` option is loaded and executed (when viewed via the presentation mode, `/p/...`), subject to the server's CSP and their Content-Type header
- We can use JSONP endpoints that bypass the CSP (e.g. on Vimeo or Slideshare, or on Disqus with a free API key) to run arbitrary existing JS functions *without arguments*
- We use `Reveal.navigateRight()` followed by `RevealMarkdown.processSlides()` to inject HTML from a `div` element with a `data-markdown` attribute in the speaker notes into the DOM. The slide navigation is necessary to ensure that the speaker notes are outside of the DOM of the original slide before the second request comes in (duplicate calls to `processSlides` break the proof-of-concept).
- Usually, the included markdown would be enclosed in a `<script type="text/template">` tag to ensure that nothing can escape into the DOM. However, `reveal-markdown.js` does not properly escape `</script>` tags (the check is case sensitive, but should at the very least be case insensitive):

```
// prevent script end tags in the content from interfering
// with parsing
content = content.replace(/<\/script>/g, SCRIPT_END_PLACEHOLDER)
```

- Actual JS in this DOM injection will never be loaded, because it is assigned via `innerHTML`, but because the CSP includes so many different embed features, we can load [this jQuery templating code](#) from Disqus (it is used on the Disqus login page, fairly easy to find) that walks the entire DOM and renders jQuery template strings.
- From the template, we can simply grab another script from anywhere (including another note) and `eval` it.

A full proof-of-concept implementation can be found [here](#):

- The main note (`pwn.md`) is responsible for loading the templating JS (and the old jQuery version that it requires) from Disqus, and issuing the two JSONP calls that lead to the DOM injection
- The `payload.md` note contains the content that is injected into the DOM, and loads and executes the final payload
- The `win.js` script is the final XSS payload (here, because this was for [hxp 2020 CTF's hackme challenge](#) it simply grabs the contents of the `/s/the-flag` page and sends it to the attacker, but you can just as easily run any other JavaScript code).
- The `pwn.py` script automates the upload process and automatically reports the XSS page (the slides of `pwn.md` in presentation mode) to the challenge admin, which isn't really relevant here.

The most straightforward way to mitigate this is to fix the DOM injection in `reveal-markdown.js` by modifying the check for `</script>` to cover everything that browsers use to end a `<script>` tag. I also wonder what value the `dependency` YAML option really brings to the slides feature - it is restricted to loading JS from pages listed in the CSP anyways, so it would probably be better to just load scripts related to e.g. video embeds as needed (just as in "normal" markdown mode), and discard the option otherwise.

Note that while the CTF challenge was using a (slightly modified, to fix some - but given some of the solutions, apparently not all - dependencies with known CVEs) CodiMD 2.2.0, this exploit still works against the official 2.3.2 release, and [on hackmd.io \(source\)](#).

2

DerMolly added a commit to `hedgedoc/hedgedoc` that referenced this issue on Jan 13, 2021

added sanitation to the `slideMode` in `frontmatter` ...

6621362

DerMolly mentioned this issue on Jan 14, 2021

added sanitation to the `slideMode` in `frontmatter` `hedgedoc/hedgedoc#727`

Merged

2 tasks

DerMolly added a commit to `hedgedoc/hedgedoc` that referenced this issue on Jan 14, 2021

added sanitation to the `slideMode` in `frontmatter` ...

6f51eae

DerMolly added a commit to `hedgedoc/hedgedoc` that referenced this issue on Jan 14, 2021

added sanitation to the `slideMode` in `frontmatter` ...

35b0d39

DerMolly added a commit to `hedgedoc/hedgedoc` that referenced this issue on Jan 14, 2021

changed the `SCRIPT_END_PLACEHOLDER` regex to case insensitive ...

1546786

jackycute mentioned this issue on Jan 21, 2021

Fix slide mode stored XSS #1650

Merged

Yukaii closed this as completed in [#1650](#) on Jan 25, 2021

Assignees

No one assigned

---

Labels

None yet

---

Projects

None yet

---


Milestone

No milestone

---

Development

Successfully merging a pull request may close this issue.

 **Fix slide mode stored XSS**  
hackmdio/codimd

---

1 participant

