

Unrestricted file upload leads to Stored XSS

[HackerOne report #880099](#) by [semsem123](#) on 2020-05-21, assigned to [@vdesousa](#):

Summary

I found that I can upload png file with JavaScript code and execute it in wiki page.

Steps to reproduce

(Step-by-step guide to reproduce the issue, including)

- 1-login to gitlab account
- 2-open your project
- 3-open Wiki page.
- 4-Click "New page" button.
- 5-attach png file which contain below code

```
<?xml version="1.0" standalone="no"?><DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"><svg onload="alert(1)" xmlns="http://www.w3.org/2000/svg"> <polygon
id="triangle" points="0,0 0,50 50,0" fill="#000900" stroke="#004400"/> </svg>
```
- 6-Click "Create page" button.
- 7-Click on green triangle
- 8-if The alert dialog not appears from first time just click on it one more time

Impact

If wiki pages created by using this vulnerability are visible to everyone (Wiki Visibility setting is set to "Everyone With Access") in "Public" project, there is a possibility that a considerable number of GitLab users and visitors click a malicious link.

Examples

gitlab.com

tested on Google Chrome

<https://gitlab.com/semsemhacker123/semsestest/-/wikis/aaaa-home>
6: <https://gitlab.com/semsemhacker123/semsestest/-/wikis/uploads/1308853a75502f77b3e22a2f9b0cc88a/11111111.png>

What is the current *bug* behavior?

The alert dialog appears after clicking "green triangle" in created page.

What is the expected *correct* behavior?

the png file it must be not executed as `!image/svg+xml`

Impact


An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page

Attachments


Warning: Attachments received through HackerOne, please exercise caution!

- [11111111.png](#)
- [gitlab_poc.png](#)

📁 Drag your designs here or [click to upload](#).


Tasks 


No tasks are currently assigned. Use tasks to break down this issue into smaller parts.


Linked items 


Link issues together to show that they're related or that one is blocking others. [Learn more](#).


Activity


 **GitLab SecurityBot** added [security](#) [severity](#) [2](#) scoped labels [2 years ago](#)

 **GitLab SecurityBot** added [HackerOne](#) [security](#) labels [2 years ago](#)

 **GitLab SecurityBot** added [security-group-missing](#) [security-triage-assess](#) labels [2 years ago](#)


 **Vitor Meireles De Sousa** removed [priority](#) [2](#) [severity](#) [2](#) labels [2 years ago](#)


 **Vitor Meireles De Sousa** added [priority](#) [3](#) [severity](#) [3](#) scoped labels [2 years ago](#)


 **Vitor Meireles De Sousa** [@vdesousa](#) · [2 years ago](#)


Requires the user to click on the SVG file to trigger.

Developer

 **Vitor Meireles De Sousa** changed due date to August 21, 2020 [2 years ago](#)


 **Vitor Meireles De Sousa** added 1 deleted label [2 years ago](#)


 **Vitor Meireles De Sousa** added [design](#) [create](#) scoped label [2 years ago](#)

 **Vitor Meireles De Sousa** [@vdesousa](#) · [2 years ago](#)

[@vdesousa](#) [@dtschier](#) SVG XSS on Wiki attachments using the following endpoint `/api/v4/projects/<id>/wikis/attachments` . Requires the user to click on the SVG.

Developer

 **GitLab SecurityBot** removed [security-group-missing](#) [security-triage-assess](#) labels [2 years ago](#)

 **Markus Koller** [@touceira](#) · [2 years ago](#)


Looks like the problem is that we're sending wiki attachments with `Content-Disposition: inline`, but should use `attachment` instead to force a download in the browser. We can fix this by passing `inline: false` in https://gitlab.com/gitlab-org/gitlab/blob/5a66b453f3b3ff6ccb52ead5186551bd7b6280d/app/controllers/projects/wikis_controller.rb#L49

I'm curious why the same exploit doesn't work in other cases, e.g. `Projects::RawController#show` also uses `send_blob` with `inline: true` by default, and it looks like we have some special processing in `Workhorse` to force `attachment` for unsafe content types like SVG: https://gitlab.com/gitlab-org/gitlab-workhorse/blob/c4f1edeb92ee7203569f774d2ad5b7ea0a0b518/internal/header/content_headers.rb#L34

For some reason this doesn't apply for wiki attachments, not sure if this requires more investigation [@vdesousa](#).


Edited by [Markus Koller](#) [2 years ago](#)

Contributor

 **Luke Duncalf** [@lukes](#) · [2 years ago](#)

[@touceira](#) [@dtschier](#) This is extremely related to <https://gitlab.com/gitlab-org/gitlab/-/issues/213310> which has a good overview of the current methods of handling SVG XSS vulnerabilities (and the pitfalls).

Maintainer

 **Markus Koller** [@touceira](#) · [2 years ago](#)

[@lukes](#) thanks for the link!

Workhorse will ensure a `Content-Disposition` of `attachment` on a 200 but not a 304.

I checked the wiki attachments again and when the SVG is not cached it's indeed sent with `attachment`, so this explains the Workhorse behaviour from my comment above.

It also looks like we're already doing everything correctly from the Rails side and sending `max-age=60`, `public`, `must-revalidate`, `no-store` (or `private` for private projects) but Cloudflare is overriding this on .com with `public`, `max-age=14400`: https://gitlab.com/gitlab-org/gitlab/-/issues/213310#note_316944578

On staging we have `Respect Existing Headers` enabled, and I do get the correct cache headers there and can't reproduce this vulnerability.

Contributor

But I think we'd also still need to disable caching altogether for wiki attachments, because unauthenticated requests won't have `no-store` at all and will still cache: https://github.com/gitlab-org/gitlab/-/issues/213310#note_318182944. This mirrors what we did for similar vulnerabilities, e.g. <https://github.com/gitlab-org/security/gitlab/issues/99>

So to sum up, I think:

- We should pass `allow_caching: false` in https://github.com/gitlab-org/gitlab/-/blob/16d-8369ead9a310b6f2e0465820def9aa7a19ea/app/controllers/projects/wikis_controller.rb#L49 (or remove the argument, since that's the default).
- For gitlab.com, we'd also need the `Respect Existing Headers` setting in Cloudflare.
 - [@ahmadsherif](#) do you know if this is something we're planning to do in the near future?

/cc [@dsatchar](#)



[Vitor Meireles De Sousa](#) [@vdesousa](#) · 2 years ago

Developer

Are we sure disabling the cache will prevent it from displayed `inline` ? If Cloudflare is messing with the headers, do we have a way to enforce our headers, or disabling caching is the solution as done on other issues?

Forcing the download will highly help in solving this issue as you suggested.



[Ahmad Sherif](#) [@ahmadsherif](#) · 2 years ago

Developer

For gitlab.com, we'd also need the `Respect Existing Headers` setting in Cloudflare. [...] do you know if this is something we're planning to do in the near future?

[@toupleira](#) It's not on our radar to my knowledge, but it should be trivial to enable it.



[Markus Koller](#) [@toupleira](#) · 2 years ago

Contributor

Are we sure disabling the cache will prevent it from displayed `inline` ?

[@vdesousa](#) I think so, or at least that's what I got out of the discussion in <https://github.com/gitlab-org/gitlab/-/issues/213310>:

Best I can tell, the SVG XSS issues all stem from the same root cause. Workhorse will ensure a `Content-Disposition` of 'attachment' on a 200 but not a 304. The vulnerability occurs when the SVG comes through as 'Content-Disposition: inline' and the script is executed. There is a detailed technical description of how this happens [here](#).

Workhorse will only modify reverse proxied responses from Rails when the HTTP status is HTTP 200 OK: <https://github.com/gitlab-org/gitlab-workhorse/-/blob/master/internal/senddata/senddata.go#L75>

So we can still use `inline`, and as long as we disable caching (which means no 304 responses) Workhorse will rewrite it to `attachment` for us.

If Cloudflare is messing with the headers, do we have a way to enforce our headers, or disabling caching is the solution as done on other issues?

From my testing it seems Cloudflare is currently always overriding these headers on .com, so we need to enable that `Respect Existing Headers` setting to avoid this.

It's not on our radar to my knowledge, but it should be trivial to enable it.

[@ahmadsherif](#) thanks, should I create an issue for this in <https://github.com/gitlab-com/gl-infra/production/>?



[Ahmad Sherif](#) [@ahmadsherif](#) · 2 years ago

Developer

should I create an issue for this in <https://github.com/gitlab-com/gl-infra/production/>?

[@toupleira](#) That's right.



[Markus Koller](#) [@toupleira](#) · 2 years ago

Contributor

[@ahmadsherif](#) ok great, I added an issue now at <https://github.com/gitlab-com/gl-infra/production/-/issues/2220>.

[@dsatchar](#) we can move ahead with my proposed fix above independently, though for gitlab.com it won't be fully fixed until this infrastructure change is in place as well.



[Darva Satchar](#) [@dsatchar](#) · 2 years ago

Maintainer

[@luke](#) .

👍 Can you add a weight for this one?

Edited by [Darva Satchar](#) 2 years ago



[Markus Koller](#) [@toupleira](#) · 2 years ago

Contributor

[@dsatchar](#) I'll add a weight of 1 as it's just a one-line change in Rails.



[Darva Satchar](#) [@dsatchar](#) · 2 years ago

Maintainer

[@toupleira](#) .

Do you think you have the time during this release to fix this without negatively impacting this release? If, so please feel free to work on this. 🙏



[Markus Koller](#) [@toupleira](#) · 2 years ago

Contributor

[@dsatchar](#) no guarantees, but I'll see that I can fit it in! 🙏

Please [register](#) or [sign in](#) to reply



[Darva Satchar](#) added [security](#) planning breakdown scoped label 2 years ago.



[Darva Satchar](#) [@dsatchar](#) · 2 years ago

Maintainer

[@brouillon](#) .

Do you have any thoughts on this one?



[Markus Koller](#) changed weight to 1 2 years ago.



[Markus Koller](#) changed milestone to %13.1 2 years ago.



[Markus Koller](#) assigned to [@toupleira](#) 2 years ago.



[Markus Koller](#) changed health status to needs attention 2 years ago.



[Markus Koller](#) [@toupleira](#) · 2 years ago

Contributor

[@ahmadsherif](#) thanks for applying <https://github.com/gitlab-com/gl-infra/production/-/issues/2220>!

[@vdesousa](#) [@dsatchar](#) with this change in place I can't actually reproduce this anymore on .com. I thought the exploit would still work with unauthenticated users and public projects where we send `Cache-Control: max-age=60, public`, but because we also send `Content-Disposition: attachment` the browser doesn't cache the response at all. I can't find any mention of this in RFCs, but at least Firefox and Chrome seem to have this behaviour. I also can't reproduce it locally in the GDK.

We could still explicitly disable caching, but it looks like there's no need anymore for that. WDYT?



[Vitor Meireles De Sousa](#) [@vdesousa](#) · 2 years ago

Developer

[@toupleira](#)

I can perform retesting next week (not sure I got some time today) on .com just to cross-check too.

I would prefer explicitly disable caching just in case someone finds another issue that exploits this vector.

What we can also do, if you're short on time, is to go as it is (since it looks like it's fixed - but I will re-test first), and we create an issue to disabled caching next.



[Markus Koller](#) [@toupleira](#) · 2 years ago

Contributor

[@vdesousa](#) that's okay, I was planning to submit the security fix today as it should be just a small change 😊



[Vitor Meireles De Sousa](#) [@vdesousa](#) · 2 years ago

Developer

[@toupleira](#) I just retested on .com on a private project, here is the headers:

```
Cache-Control: max-age=60, private, must-revalidate, no-store
Content-Disposition: attachment
```

for a public project (and wiki set to everyone):

Cache-Control: max-age=60, public, must-revalidate, no-store

Content-Disposition: attachment

Both have the same headers, so as far as it is right now this issues is fixed with the current headers.

Markus Koller @touveira · 2 years ago

Contributor

@vdesousa thanks for confirming!

Please [register](#) or [sign in](#) to reply

Darva Satcher @dsatcher · 2 years ago

Maintainer

@touveira ·

Can you give me a little more information regarding this so that I can have a better understanding of the consequence of making or not making this change?

1. Are there other benefits to disabling caching?

2. Are there disadvantages to disabling caching?

Markus Koller @touveira · 2 years ago

Contributor

@dsatcher sure!

Are there other benefits to disabling caching?

Basically that would give us certainty that caching is really disabled. Using Content-Disposition: attachment seems to disable caching as well, but I'm not sure how reliable this is in other browsers (I only tested Firefox and Chrome).

Are there disadvantages to disabling caching?

I don't think so, as it looks like we currently end up not caching anyway.

That's two arguments in favour of doing the no-caching fix, so I'll go ahead and do that 🙌

Please [register](#) or [sign in](#) to reply

GitLab SecurityBot mentioned in issue adietrich/sandbox#931 · 2 years ago

Darva Satcher added workflow ready for development scoped label and automatically removed workflow planning breakdown label · 2 years ago

Darva Satcher added workflow scheduling scoped label and automatically removed workflow ready for development label · 2 years ago

Darva Satcher changed milestone to %13.2 · 2 years ago

Christen Dybenko added workflow ready for development scoped label and automatically removed workflow scheduling label · 2 years ago

Darva Satcher added Deliverable label · 2 years ago

Markus Koller added workflow in dev scoped label and automatically removed workflow ready for development label · 2 years ago

Markus Koller added workflow in review scoped label and automatically removed workflow in dev label · 2 years ago

Markus Koller added workflow verification scoped label and automatically removed workflow in review label · 2 years ago

Markus Koller @touveira · 2 years ago

Contributor

This is now fixed with https://gitlab.com/gitlab-org/security/gitlab/-/merge_requests/632.

@vdesousa I'll reassign the issue back to you for making it public etc.

Markus Koller assigned to @vdesousa and unassigned @touveira · 2 years ago

Markus Koller removed workflow verification label · 2 years ago

Costel Maxim @cmaksim · 2 years ago

Developer

Issue fixed in 13.1.2


Costel Maxim closed · 2 years ago

Costel Maxim assigned to @cmaksim and unassigned @vdesousa · 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago

AuthorReporter

This [HackerOne security](#) issue was closed 30 days ago and may become public.

Please ensure the following items are true and add a  reaction:

- Issue description and comments do not contain sensitive data belonging to GitLab.
- Issue does not reveal private information of the reporter (i.e. session IDs, passwords).

If the issue needs to stay confidential, please add the [how confidential](#) label.

If you removed confidential data from the issue description before making it public, make sure that the description history entry is deleted.

Vitor Meireles De Sousa made the issue visible to everyone · 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago

AuthorReporter

[HackerOne report #890099](#) was disclosed on 2020-08-03 @ 12:26.

- Bounty awarded: \$1500

Jeremy Mastos mentioned in epic 83979 (closed) · 2 years ago

Matthias Klappler mentioned in issue #237848 (closed) · 2 years ago

Please [register](#) or [sign in](#) to reply