

Share:     

TIMELINE



joaxcar submitted a report to GitLab.

Sep 16th (about 1 y

Summary

Hi GitLab team, I found a stored XSS in merge request creation page caused by a payload in the name of an "approval rule".

Adding approval rules is a feature that is unlocked for premium subscriptions or above. This does not seem to block it from being used against regular users on for example Gitlab.com by inviting them into the "infected project".

This occurs when adding an "Approval rule" to a project and giving it a javascript/html payload as the name and attaching the rule to an approver. When a user tries to create a merge request in the project and opens the "Reviewers" dropdown, information about the user with the attached rule will be shown and the rule name will be injected underneath.

With the payload

Code 115 Bytes

[Wrap lines](#) [Copy](#) [Down](#)

```
1 <iframe/srcdoc='<script/src=/joaxcar_group/first/-/jobs/1415515489/artifacts/raw/data/alert.js></script>'></iframe>
```

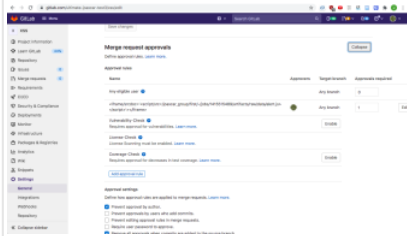
this XSS bypasses the current CSP on Gitlab.com (tried it with an Ultimate trial and inviting a user without a trial to the project)

As I got the impression that all XSS are treated equal when reporting a similar issue, I have not made any deeper analysis of the reason for this firing. Thought I just report it right away. Please reach back to me if you need me to research the impact deeper! As an example, it does not fire when one "edits" a MR which is a bit odd

Steps to reproduce

1. Create two user accounts, `attacker_user` and `victim_user` (`attacker_user` must have at least premium features enabled)
2. Log in as `attacker_user`
3. Create a project `xss_project` by going to https://gitlab.com/projects/new#blank_project
4. Go to projects settings on https://gitlab.com/attacker_user/xss_project/edit and scroll down to and expand "Merge request approvals"

Image F1450906: approvals.png 214.42 KiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)

5. Click "Add approval rule"

6. Put the payload as the name, If on Gitlab.com use

Code 115 Bytes

[Wrap lines](#) [Copy](#) [Down](#)

```
1 <iframe/srcdoc='<script/src=/joaxcar_group/first/-/jobs/1415515489/artifacts/raw/data/alert.js></script>'></iframe>
```

If this is tested on a server without CSP feel free to use the payload

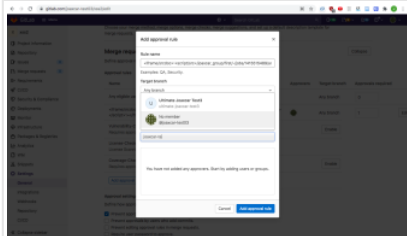
Code 39 Bytes

[Wrap lines](#) [Copy](#) [Down](#)

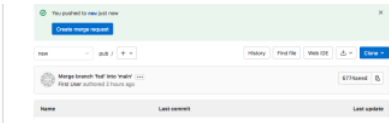
```
1 <script>alert(document.domain)</script>
```

7. Search for and select `attacker_user` as approver and click create rule.

Image F1450905: create.png 212.41 KiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)8. Invite `victim_user` to the project as `Developer` on https://gitlab.com/attacker_user/xss_project/-/project_members9. Log out and log back in as `victim_user`10. Go to [https://\[redacted\]/user_01/pub/-/branches/new](https://[redacted]/user_01/pub/-/branches/new) and create a branch `new`

11. Directly click on "Create merge request" (which will appear on the screen)



- 12. Click on the dropdown at "Reviewers"
- 13. Payload will trigger

Image F1450904: fire.png 167.38 KiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)



Impact

Stored XSS with CSP bypass. Full Javascript functionality without restrictions, so everything from stealing data to generating and exfiltrating access tokens.

Examples

If you access my private project at Gitlab.com (<https://gitlab.com/ultimate-joaxcar-test3/xss>) as an admin, you should be able to create an MR and trigger payload (Just an alert box)

What is the current *bug* behavior?

Approver rule name is injected in the user information without proper sanitization.

What is the expected *correct* behavior?

The name should be sanitized

Output of checks

This bug happens on GitLab.com

Impact

Stored XSS with CSP bypass. Full Javascript functionality without restrictions, so everything from stealing data to generating and exfiltrating access tokens.

- 4 attachments:
- F1450903: [newMR.png](#)
 - F1450904: [fire.png](#)
 - F1450905: [create.png](#)
 - F1450906: [approvals.png](#)

[joaxcar](#) posted a comment. Sep 16th (about 1 y)
Made a mistake on step 10. I added a link to my personal test server. If possible please edit the step to say:
https://gitlab.com/attacker_user/xss_project/-/branches/new

[mhenriksen](#) (GitLab staff) posted a comment. Sep 17th (about 1 y)
Hi [@joaxcar](#), thanks for submitting this finding to us! Unfortunately, it looks like I can't edit the report on my side (I can only change the title). Perhaps [@forest_dw](#) or someone else in the HackerOne triage team can help?

[joaxcar](#) posted a comment. Sep 17th (about 1 y)
Hi [@mhenriksen](#) it is no big problem, If the report is accepted and later made public on your issue tracker, then it might be of interest to redact the link. But there is nothing really to hide on my instance anyway, just for privacy I guess :)

I did some quick digging into the source code and I think the problem is located in the file https://gitlab.com/gitlab-org/gitlab/-/blob/master/app/assets/javascripts/users_select/index.js in the end of the file there is the function

```
Code 654 Bytes Wrap lines Copy Download


1 UsersSelect.prototype.renderApprovalRules = function (elClassName, approvalRules = []) {
2   const count = approvalRules.length;
3
4   if (!elClassName?.includes('reviewer') || !count) {
5     return '';
6   }
7
8   const [rule] = approvalRules;
9   const countText = sprintf('__(%{count}&nbsp;rules)', { count });
10  const renderApprovalRulesCount = count > 1 ? `<span class="ml-1">${countText}</span>` : '';
11  const ruleName = rule.rule_type === 'code_owner' ? __('Code Owner') : rule.name;
12
13  return `<div class="gl-display-flex gl-font-sm">
14    <span class="gl-text-truncate" title="${ruleName}">${ruleName}</span>
```


which uses `rule.name` without sanitation. (line 11 and 14 in my code listing)

It was added 10 months ago in this commit <https://gitlab.com/gitlab-org/gitlab/-/commit/3bf38888882aca098659d71fb6e84ba3daa371f>

Thank you for looking into the report!


/Johan

  changed the status to . Sep 17th (about 1 y)

 OK, I will modify the URL to your suggestion in the imported GitLab issue! We have verified this finding and have escalated to our engineering team. We're tracking progress internally at <https://gitlab.com/gitlab-org/gitlab/-/issues/341140>. This issue will be made public 30 days following the release of a patch.

We will continue to update you via HackerOne as a patch is scheduled for release.

Best regards,
GitLab Security Team


 posted a comment. Sep 23rd (about 1 y)
ETA for fix:


Hi ,

The issue you reported is currently scheduled to be fixed by 2021-10-31.

Thank you again for contacting us!

Best regards,
GitLab Security Team

 posted a comment. Oct 8th (about 1 y)

Hi  I believe that this got patched a week ago in 14.3.1. Is there anything holding the report back? The fix seems to work on Gitlab.com

Regards
Johan

 rewarded  with a \$3,000 bounty. Oct 8th (about 1 y)

Hi ,

I'm sorry for the delay on this, I have been out due to sickness, but you are totally right; it was patched in the latest security release. Thanks for retesting the issue!

We look forward to your next report!

Best regards,
GitLab Security Team

 posted a comment. Nov 1st (about 1 y)

Hi , thank you for the bounty!

I think you forgot to close the report here, :) I thought that I might request a disclosure on this one, but the option is missing when the report is still open

Best regards
Johan

  closed the report and changed the status to . Nov 1st (about 1 y)


 Oops, you're totally right! Closing it now.

 requested to disclose this report. Jan 27th (11 mon)

Could we disclose this one?

Best regards
Johan

 posted a comment. Feb 22nd (10 mon)

 is it possible to disclose this one? Forgot to ping you in my last request

Best regards
Johan

  agreed to disclose this report. Mar 31st (9 mon)

Hello ,

Yes, we can disclose this one. I'll also be making the [GitLab issue public](#), please double check to make sure that we've done the appropriate redactions and let us know if we missed anything.

Have a great day!

Andrew
Security Team | GitLab

 This report has been disclosed. Mar 31st (9 mon)

