Closed | Issue created 7 months ago by GitLab SecurityBot

IDOR in Zentao integration leaks details of issues from a users other "products"

HackerOne report #1542834 by `joaxcar` on 2022-04-16, assigned to @ngeorge1:

Report | How To Reproduce

Report

Summary

Maintainers of projects with a premium subscription have the option to enable "Zentao product integration" (documentation).

This enables users with access to the project to view a list of `product` issues synced from a connected Zentao `product` by visiting https://gitlab.com/GROUPNAME/PROJECTNAME/-/integrations/zentao/issues .

And issue details by visiting https://gitlab.com/GROUPNAME/PROJECTNAME/-/integrations/zentao/issues/ISSUE-KEY

When configuring this feature, the maintainer is made to enter a `product ID`

> ZenTao Product ID: To display issues from a single ZenTao product in a given GitLab project. The Product ID can be found in the ZenTao product page under Settings > Overview.

This is to connect the current `GitLab project` with the target `Zentao product`

The way GitLab handles this functionality is by requesting the maintainer to enter two things other than the project key

1. Zentao instance URL
2. Zentao API key

When a user visits the `/-/integrations/zentao/issues` page the GitLab backend makes a request towards Zentao like so

```
https://username.zentao.net/api.php/v1/products/3/issues
```

Which returns a list of all issues related to `product` with ID 3.

When clicking on one of these issues in the UI the user is redirected to `/-/integrations/zentao/issues/issue-1` and the backend will make a request like so

```
https://jihudemo.zentao.net/api.php/v1/issues/issue-1
```

This request does not contain any check as to which `product` this issue is related. So by enumerating issue titles such as `task-ID`, `bug-ID`, `story-ID` and so on an attacker can enumerate all issues from all `products` the maintainer controls in its Zentao instance.

A small discussion on CVSS

I am not completely sure on how this is intended to work, and Zentao's English documentation is a bit sparse. But from what I have gathered in the GitLab documentation, and issues surrounding this, the intention is to connect ONE GitLab project with ONE Zentao `product` .

So any issues not related to this given `product` should not be accessible to other users.

My understanding of Zentao is that the user can organize issues in `products` , but all issues share names with incrementing values. So two `products` with two `bug` issues will create the four `bug` IDs

```
bug-1
bug-2
bug-3
bug-4
```

This is the same for issue types of `task` and `story` . This makes it extremely easy to enumerate these issues. Hence, the complexity scoring of `Low` .

There is also no restriction for this feature being used as the issue tracker for a `public` project. If this is done, the attacker can perform the attack unauthenticated.

I will report this with a CVSS which is the "highest possible impact" as nothing I have described above is outside "normal workflow". But as usual I understand that the scoring can be subject to change :)

Steps to reproduce

I will use the test account that is used in GitLab issues regarding Zentao. `Zentao demo account` from #338178 (closed)

> Zentao Web URL: https://jihudemo.zentao.net
> Zentao API token: a70fe861f1e60a3bfd857133798f2f1f
> Zentao Product ID: 4

1. Create a user to use as the victim
2. Create a new public project (with minimum premium access to GitLab)
3. In the project page go to https://gitlab.com/GROUPNAME/PROJECTNAME/-/integrations/zentao/edit
4. Fill in the test credentials from above (use product ID 4 as this one only contains one issue)
5. Click save
6. Log out
7. Go to https://gitlab.com/GROUPNAME/PROJECTNAME/-/integrations/zentao/issues
8. The page will list all issues related to the Product. In this case ONE bug issue
9. Click the issue (bug-8) and you will go to https://gitlab.com/GROUPNAME/PROJECTNAME/-/integrations/zentao/issues/bug-8
10. Now change `bug-8` to `bug-1`
11. Details of another `product` is shown
12. Start enumerating

Impact

Zentao integration leaks a user's ALL issues from ALL `products` even if the documentation states that it will show issues from ONE `product`

Examples

Visit my public project at

https://gitlab.com/ultimate_joaxcar_2/kk/-/integrations/zentao/issues

This is accessible as an unauthenticated user.

Try to enumerate issues with names such as `bug-1`, `task-1` and `story-1`

What is the current *bug* behavior?

GitLab uses Zentao issue key directly from attacker supplied URL without matching it to the configured product key.

What is the expected *correct* behavior?

A project should only fetch keys that match the configured key base

Output of checks

This bug happens on GitLab.com

Impact

Zentao integration leaks a user's ALL issues from ALL `products` even if the documentation states that it will show issues from ONE `product`

How To Reproduce

Please add reproducibility information to this section:

1.
2.
3.

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 June 26, 2022 7 months ago

GitLab SecurityBot added Weakness CWE-639 bug vulnerability type bug priority 3 severity 3 scoped labels 7 months ago

GitLab SecurityBot added HackerOne security labels 7 months ago

GitLab SecurityBot @gitlab-securitybot · 7 months ago Author Reporter

HackerOne comment by `golden__retriever` :

Hello [@]joaxcar,

I hope all is well and thanks for leaving a message.

I was not able to get past the full steps to reproduce. I had some issues when it came to getting the second account working. Here are the steps I took to reproduce this report.

1. I as the root user created a public project which had the following URL:

```
https://gitlab.ourdomainname.com/root/publicproject
```

2. I set up the Zentao Integration as prescribed by you

3. I logged into the other user, and tried to go to the URL you suggested and I had no ability to see it:

```
https://gitlab.techkranti.com/root/publicproject/-/integrations/zentao/edit
```

Can you please let me know what I did wrong?

Thanks and regards, [@]golden__retriever r

Attachments

Warning: Attachments received through HackerOne, please exercise caution!

- Screen_Shot_2022-04-16_at_4.02.24_PM.png
- Screen_Shot_2022-04-16_at_4.02.09_PM.png

GitLab SecurityBot @gitlab-securitybot · 7 months ago Author Reporter

HackerOne comment by `joaxcar` :

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

It sounds like you performed the correct steps up until revisiting the project after logout. You wrote that you tried to visit

```
https://gitlab.techkranti.com/root/publicproject/-/integrations/zentao/edit
```

and got the 404 page. It looks like you are visiting the wrong path. The `/-/integrations/zentao/edit` with `edit` in the end of the URL is only available to project maintainers. To view the issue list you need to visit

```
https://gitlab.techkranti.com/root/publicproject/-/integrations/zentao/issues
```

with `issues` in the end of the URL.

To access a single issue visit

```
https://gitlab.techkranti.com/root/publicproject/-/integrations/zentao/issues/bug-1
```

Hope this resolves the problem!

Best regards Johan

GitLab SecurityBot @gitlab-securitybot · 7 months ago Author Reporter

HackerOne comment by `golden__retriever` :

Hello [@]joaxcar,

I hope all is well and thanks for leaving a message.

I have validated the report and have forwarded it to the GitLab team for review. Please remain patient for an update.

Thanks and regards, [@]golden__retriever

Nikhil George added group integrations scoped label 7 months ago

Nikhil George added 1 deleted label 7 months ago

Nikhil George @ngeorge1 · 7 months ago Developer

The issue is reproducible. A product in ZenTao is linked to a Project in Gitlab by providing the product ID when setting up ZenTao integration, however by changing the bug-id(which is sequential) in the URL one can access bugs in another product.

GitLab SecurityBot @gitlab-securitybot · 7 months ago Author Reporter

@g.hickman @arturoherrero @mhenriksen This issue is ready for triage as per HackerOne process.

About this automation: AppSec Escalation Engine

Arturo Herrero added Integration ZenTao scoped label 7 months ago

Arturo Herrero added backend label 7 months ago

GitLab Bot added section dev scoped label 7 months ago

Arturo Herrero @arturoherrero · 7 months ago Maintainer

@meks @kwiebers Do we have documentation about handling security issues related to JiHu contributions/features?

Mek Stittri @meks · 7 months ago Maintainer

Yes we do, this is captured in https://about.gitlab.com/handbook/engineering/security/security-operations/sirt/security-incident-communication-plan.html#communications-channels-and-forms

However this is about general vulnerability. Given this is an area owned by JiHu it would be reasonable to provide faster feedback for them to address.

@jritchey @kbychu thoughts on this?

Arturo Herrero @arturoherrero · 7 months ago Maintainer

We have another ZenTao vulnerability #360540 (closed).

Kevin Chu @kbychu · 7 months ago Maintainer

Agreed with @meks

I think we should do something like the following:

1. Clone the security issues in jh-enablement so JiHu members
2. Keep this issue open because we want to be able to track the issue
3. Ask JiHu to address it and ping us when it's resolved so we can close this issue

We should create a process for addressing regular, non-fire-drill security issues with JiHu. I assume that doesn't exist yet as I have not come across them.

Mek Stittri @meks · 7 months ago Maintainer

Linking slack discussion thread
https://gitlab.slack.com/archives/C01S3DV4MSR/p1651001335373169

Dominic Couture @dcouture · 4 months ago Developer

@meks @kbychu This integration is even more "special" because we're using an API that was developed by Zentao on JiHu's request. It does seem like JiHu would be better suited to address this.

In an issue like XSS in ZenTao integration affecting self hosted... (#360540 - closed) which could be exploited in GitLab.com and affect any users regardless of if they use Zentao or not it's probably preferable that we fix it, but for issues like this one I like @kbychu 's suggestion.

/cc @ankelly

Kevin Chu @kbychu · 4 months ago Maintainer

Thanks @dcouture

Interesting. Is this because the solution will require Zentao to update their APIs?

cc @meks

Dominic Couture @dcouture · 4 months ago Developer

We're likely to have a solution that won't require the update, but long term an update would be good. It's just a weird situation because we're in a position where we need to fix something caused by a design that seems to have been discussed between 2 parties that won't be involved in fixing the bug

Please register or sign in to reply

Mek Stittri mentioned in issue gitlab-com/chief-of-staff-team/cos-team#177 7 months ago

Mek Stittri marked this issue as related to gitlab-com/chief-of-staff-team/cos-team#177 6 months ago

Mek Stittri mentioned in issue gitlab-com/chief-of-staff-team/jihu-billing#3 6 months ago

Grant Hickman @g.hickman · 6 months ago Developer

@gitlab-org/ecosystem-stage/integrations can we estimate this issue? As per https://gitlab.com/gitlab-com/chief-of-staff-team/cos-team/-/issues/177#note_941972567, we're on the hook for the two open security issues.

Grant Hickman @g.hickman · 6 months ago Developer

Based on security prioritization guidelines, it looks like this requires remediation in 90 days, making the due date Jul 24, 2022. Updating the due date. This should also fit in just ahead of the 28th for the security release, if completed by the 26th.

Edited by Grant Hickman 6 months ago

Andy Soiron @Andysoiron · 6 months ago Maintainer

@g.hickman we are refining this in this week's backlog refinement, looks like it will be a weight 2-3.

Grant Hickman @g.hickman · 6 months ago Developer

Thanks @Andysoiron!

Please register or sign in to reply

Grant Hickman changed milestone to %15.1 6 months ago

GitLab Bot added Accepting merge requests label 6 months ago

Grant Hickman added Backlog Refinement Integrations scoped label 6 months ago

Grant Hickman changed milestone to %15.2 6 months ago

GitLab Bot mentioned in issue #362443 (closed) 6 months ago

Luke Duncalfe @.luke · 6 months ago Maintainer

I haven't been able to find any API docs (possibly because I'm limited to searching in English!), but perhaps the API call can be changed to be scoped by a ZenTao project ID, rather than one that can fetch any issue.

Otherwise, somewhere in that method, we could reject the fetch, the earlier the better.

If navigating a lack of English API docs is a problem, we might be able to reach out to `@icbd` the JiHu engineer who contributed the ZenTao integration and ask for their advice or help with locating the docs (without giving out details of the exploit?).

Edited by Luke Duncalfe 4 months ago

Luke Duncalfe @.luke · 4 months ago Maintainer

A great discovery about the ZenTao API is that the endpoints we integrate with were added especially by ZenTao for GitLab, and are currently undocumented #366792 (comment 1015584174).

For reference the *documented* API endpoints (these exclude the ones we use) are https://www.zentao.net/book/apidoc-v1/664.html.

I'm adding this information to the group integrations internal docs https://gitlab.com/gitlab-org/ecosystem-stage/integrations/team/-/merge_requests/29.

Please register or sign in to reply

Andy Soiron @Andysoiron · 6 months ago Maintainer

I wasn't able to set up a ZenTao integration to reproduce this issue. I've created #362602 (closed) to help with the setup.

But this looks similar to #360800 (closed) which describes that we don't validate the issue ID patter when fetching Jira issues.

It looks like ZenTao uses a pattern for IDs too, which is `{{type}}-{{ID-number}}`. If we only allow to fetch resources of type `issue` we can validate that the ID must start with `issue-`.

Grant Hickman changed weight to **3** 6 months ago

Luke Duncalfe added workflow ready for development scoped label and removed Backlog Refinement Integrations label 6 months ago

Grant Hickman added Deliverable label 5 months ago

Grant Hickman mentioned in issue gitlab-org/ecosystem-stage/team-tasks#146 5 months ago

Luke Duncalfe assigned to @.luke 5 months ago

GitLab Bot @gitlab-bot · 5 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 5 months ago

Mek Stittri @meks · 4 months ago Maintainer

Hey folks, are we still on track for this issue, we would like to notify our JiHu counterparts when this is fixed.

cc @kbychu

Luke Duncalfe @.luke · 4 months ago Maintainer

@meks We haven't made any progress on this issue so far. I assigned it to myself a week ago when prompted to self-assign at the beginning of the %15.2 milestone, but I didn't expect a weight **3** issue could make the last security release (cutoff was yesterday). I'm hoping to turn my attention to this now, and will hopefully make the next security release. Our milestones for security release issues are confusing because we work in them in, say, %15.2, but unless it's fixed immediately in the milestone it won't actually get merged until %15.3.

Luke Duncalfe added workflow in dev scoped label and automatically removed workflow ready for development label 4 months ago

Luke Duncalfe @.luke · 4 months ago Maintainer

The required data is not available

The ZenTao `issues/:ID` endpoint we request the issue data from doesn't return back any data that we can use to cross-check the "product ID" of the issue matches the `Integrations::ZenTao#zentao_product_xid`. An example response can be seen here #366792 (comment 1015584174).

Ideal fix

The endpoint was created by ZenTao, especially for the GitLab ZenTao integration #366792 (comment 1015584174). It seems likely we could ask them to make further changes to allow us to fix this security issue.

Ideally, we ask ZenTao to either:

- Supply the product ID in the response from `issues/:ID` so we cross-check the product before returning the data to the frontend.
- Change the endpoint to be `products/:PRODUCT_ID/issues/:ISSUE_ID` and they perform the check.

Problems?

But here is why I think asking ZenTao to make one of the above changes might not be best for this security issue:

- We don't know the timeframe of when ZenTao could deliver the change. Meanwhile, the vulnerability exists.
- Requesting this change might give away a clue to the security issue we're working on.

Proposed workaround

[Update, this workaround doesn't work for "tasks" #360372 (comment 1016963713)].

We can get the product ID of the issue via other endpoints particular to the type of issue (a "story", "bug" or "task" - although tasks might not be one, I'll check) - `stories/:ID` (doc), `bugs/:ID` (doc), `tasks/:ID` (doc).

These endpoints do not return other data that `issues/:ID` does (compare with the example responses here #366792 (comment 1015584174)), so we still need to call `issues/:ID` in order to render the ZenTao page.

So this workaround requires making 2 requests to display a ZenTao issue instead of 1. We can mitigate the effects of this by storing the result of the check in a cache, with a long expiry, say, a week. This would reduce the negative effect of this workaround to a once-a-week-at-most effect.

Once the security release has been released, we could contact ZenTao and ask for the ideal fix.

What are your thoughts @dcouture @g.hickman @Andysoiron?

Edited by Luke Duncalfe 4 months ago

Luke Duncalfe @.luke · 4 months ago — Maintainer

🤦 An issue can be a "task" and the `tasks/:ID` endpoint *does not* return the product ID 😳. So that defeats that idea.

Tasks must be linked to "executions" which must be linked to products, but `executions/:ID` (doc) also doesn't return the product ID either, so even a two-step look up for a task wouldn't give us the associated product ID.

At this point, it's quite hard to see how we could verify a task belongs to a product.

As a final idea, perhaps we could expect that any issue URI would only be hit after the issue had appeared on an index page. We could put the IDs of things that appear on the index page into a long-lasting cache scoped to the product ID using SADD, and only serve an individual issue page if its ID appeared in the cache. This might work. The only workflow that it could break would be where someone tried to visit an issue page directly before anyone had seen it appear in the index, which is probably unlikely.

Luke Duncalfe @.luke · 4 months ago — Maintainer

Otherwise, we could address this using the "ideal fix" of asking ZenTao to add the product ID to data returned in `issues/:ID`, but it does introduce the problems mentioned in #360372 (comment 1016950907).

Andy Soiron @Andysoiron · 4 months ago — Maintainer

@.luke thanks for lining out the possible solutions and problems in such detail.

I agree that the "ideal fix" is the best solution. Workarounds are complex, and I don't think it is worth the extra effort for closing the vulnerability until the ideal fix is available.

Grant Hickman @g.hickman · 4 months ago — Developer

@Andysoiron @.luke - I defer to your engineering expertise on this. It sounds like we should go with the ideal fix approach. It looks like you've already raised a question on getting additional details added here.

@meks would you be the ideal point of contact to coordinate with JiHu on this?

Luke Duncalfe @.luke · 4 months ago — Maintainer

@Andysoiron @g.hickman Another thought is that ZenTao can be self-managed. Our docs say we support:

> - ZenTao 15.4
> - ZenTao Pro 10.2
> - ZenTao Biz 5.2
> - ZenTao Max 2.2

I suspect a fix reliant on an API change to ZenTao wouldn't work for existing self-managed ZenTao instances that have integrated with GitLab. We'd probably need to code it in a way that skipped the check on these older instances leaving them vulnerable. We couldn't expect people to upgrade their self-managed ZenTao instances immediately upon release of the GitLab security release.

It does pose a problem of how we could ever expect self-managed ZenTao users to upgrade their ZenTao instance in order to patch this vulnerability. My original idea #360372 (comment 1016950907) was to implement a workaround and then quickly follow that up with consuming an API change on ZenTao - but I don't think we could do that.

Luke Duncalfe @.luke · 4 months ago — Maintainer

Based on:

- https://www.zentao.pm/download.html
- https://www.zentao.net/download.html

It looks like the ZenTao upgrade process is manual. I was wondering in case there might be some auto-magic where an API change might flow to existing instances automatically, but, it looks like we can't expect an existing ZenTao instance that's integrated with GitLab to have the upgraded API change at any particular point in time.

Luke Duncalfe @.luke · 4 months ago Maintainer

🤔 Based on the above, I feel like our hands are tied in terms of what our approach can be. Current ZenTao API data is inadequate, at the moment I can only think of the approach outlined at the bottom of #360372 (comment 1016963713).

Grant Hickman @g.hickman · 4 months ago Developer

@.luke Is there anyone from the security team we could reach out to for direction on this? I don't believe I'm the right person to make the call.

Luke Duncalfe @.luke · 4 months ago Maintainer

@dcouture was mentioned above 🙂. I think this is probably more of an engineering call currently. I'm just laying out the problem in case others can pick a hole in it.

Dominic Couture @dcouture · 4 months ago Developer

I'm here! Catching up on the TODOs, reading the thread now 👀

Dominic Couture @dcouture · 4 months ago Developer

If there's an API that returns everything we're supposed to have access to then indeed caching that seems to be the best path forward. We (or Jihu) could communicate with Zentao to ensure that they fix their API in future versions and in %16.0 we could make a breaking change in the versions we support and drop the caching patch.

Additional thoughts in #360372 (comment 1018415782)

Please register or sign in to reply

🤖 GitLab Bot 🤖 @gitlab-bot · 4 months ago Maintainer

@(confidential) This issue looks like it may slip this current milestone. Can you leave a 👍 or 👎 to signify if you are on track to deliver this issue? Please also consider updating the issue's Health Status or Milestone to reflect its current state, and communicate with your Product Manager as appropriate.

Bot policy.

Grant Hickman changed health status to **at risk** 4 months ago

Grant Hickman mentioned in issue gitlab-org/ecosystem-stage/team-tasks#150 4 months ago

Arturo Herrero changed milestone to %15.3 4 months ago

Arturo Herrero added missed:15.2 missed-deliverable labels 4 months ago

Luke Duncalfe added workflow in review scoped label and automatically removed workflow in dev label 4 months ago

Luke Duncalfe @.luke · 3 months ago Maintainer

@g.hickman @arturoherrero This security issue will miss the next security release. The MR is undergoing its maintainer review, though, so will become workflow awaiting security release within %15.3 but will close early in %15.4.

Arturo Herrero @arturoherrero · 3 months ago Maintainer

Yes, this was what I suspected given the current timeframe with the security release. @.luke Thanks for the communication.

Please register or sign in to reply

Arturo Herrero mentioned in issue gitlab-org/ecosystem-stage/team-tasks#159 3 months ago

Sean Gregory mentioned in issue gitlab-org/ecosystem-stage/team-tasks#153 3 months ago

Grant Hickman changed milestone to %15.4 3 months ago

Grant Hickman added missed:15.3 label 3 months ago

Grant Hickman removed the health status 3 months ago

Luke Duncalfe @.luke · 3 months ago Maintainer

The MR has had all of its approvals so I'll move this to workflow awaiting security release.

Edited by Luke Duncalfe 3 months ago

Luke Duncalfe added workflow awaiting security release scoped label and automatically removed workflow in review label 3 months ago

Luke Duncalfe @.luke · 3 months ago Maintainer

I'll close this now as the main branch was merged to `master` and backports are awaiting merge.

Luke Duncalfe closed 3 months ago

GitLab SecurityBot @gitlab-securitybot · 2 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.

Mek Stittri @meks · 2 months ago Maintainer

@ankelly can we notify our JH partners as soon as this is public, can this be this coming wednesday?

Edited by Mek Stittri 1 month ago

Andrew Kelly @ankelly · 2 months ago Developer

Yes, I will make a reminder to myself to notify them via the shared Slack once the release goes out. As agreed in the recent handbook changes, I will provide them with a link to the blog post and a link to this issue (which will remain confidential until 30 days after the release).

Dominic Couture @dcouture · 1 month ago Developer

@ankelly This was resolved in the previous release and we're at the 30 day mark now. I'll let you make the issue public if you want to coordinate the messaging.

Mek Stittri @meks · 1 month ago Maintainer

Thanks team!

Please register or sign in to reply

Andrew Kelly @ankelly · 1 month ago Developer

As it has been 30 days since the patch was released, we are making this issue public.

Andrew Kelly made the issue visible to everyone 1 month ago

GitLab Bot added devops manage scoped label 3 days ago

GitLab Bot removed 1 deleted label 3 days ago