

## **BANDIT GAME LAB 01 – IT22883902 (MARIO)**

Github – (<https://github.com/Shenal01/BanditGame.git>)

### PASSWORDS

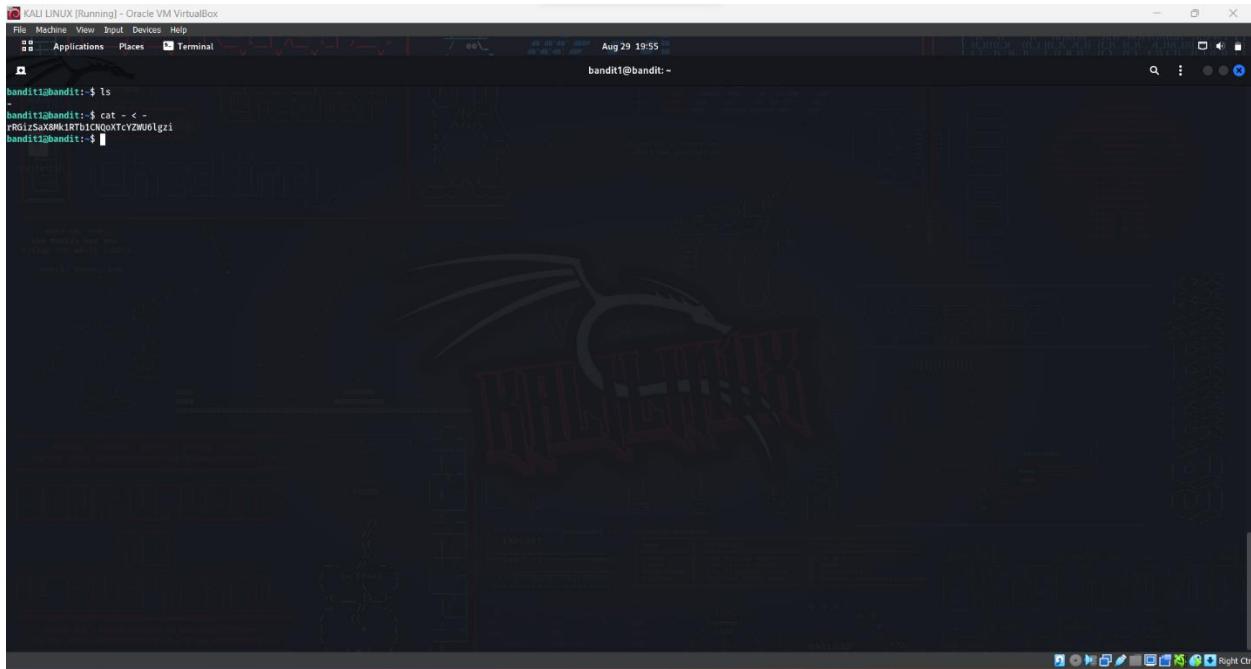
LVL 0 - 1

NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL

LVL 1 - 2

rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi

cat < .



KALI LINUX [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
Applications Places Terminal Aug 29 19:55  
bandit1@bandit: ~  
bandit1@bandit: \$ ls  
bandit1@bandit: \$ cat < -  
rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi  
bandit1@bandit: \$

LVL 2 - 3

aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG

we have to use the escape character “\”

(YOU CAN USE TAB KEY)

A screenshot of a Kali Linux desktop environment. The terminal window displays a command-line session where the user is generating a backdoor payload using msfvenom. The session includes commands like 'cat < .', 'cat . bash\_logout', and '# ./bash\_logout' which is described as being executed by bash(1) when login shell exits. The terminal also shows a script or configuration file with comments about leaving the console clear to increase privacy. In the background, there's a large watermark that reads 'DON'T UPLOAD TO VIRUSTOTAL' and 'YOU CAN UPLOAD OUTPUT/BACKDOOR FILE TO WWW.NOHOSTBRO.COM'. To the right of the terminal, there's a visualization of a backdoor, consisting of a grid of red and black dots forming a circular pattern. The bottom of the screen features a dock with various icons for tools like Metasploit, Nmap, and Wireshark.

LVL 3 - 4

2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe

The key to the next level is stored in a hidden file in the **inhere** directory according the website so in order to check the file you have to use the command “ls -al” because using only the command “ls” won’t show hidden files.

A screenshot of a Kali Linux desktop environment. The terminal window shows a user named 'bandit3' logged in as 'bandit'. The user runs several commands to list files and change directories, eventually revealing a hidden file containing a long string of characters. The desktop background features a complex, abstract geometric pattern. On the left, there's a 'firstshellsh' icon and a note that says 'wake up, Neo... the matrix has you follow the white rabbit.' Below the terminal, a banner reads 'WARNING ! WARNING ! WARNING ! WARNING ! YOU CAN UPLOAD OUTPUT/BACKDOOR FILE TO WWW.VIRUSTOTAL.COM'. At the bottom, there's a 'DON'T UPLOAD' message and a 'CREATE TOTAL' button. A 'RECC' icon is also visible. The right side of the screen shows a large, circular binary data visualization.

## LVL 4 - 5

lrIWWI6bB37kxfiCQZqUdOIJYfr6eEeqR

(find . -type f | xargs file)

find: This command is used to search for files. The -type f option tells the find command to only search for files.

.: This is the current directory.

|: This symbol is called a pipe. It is used to connect the output of one command to the input of another command.

xargs: This command takes the output of another command and passes it as arguments to another command.

file: This command prints out the type of a file.



## LVL 5 – 6

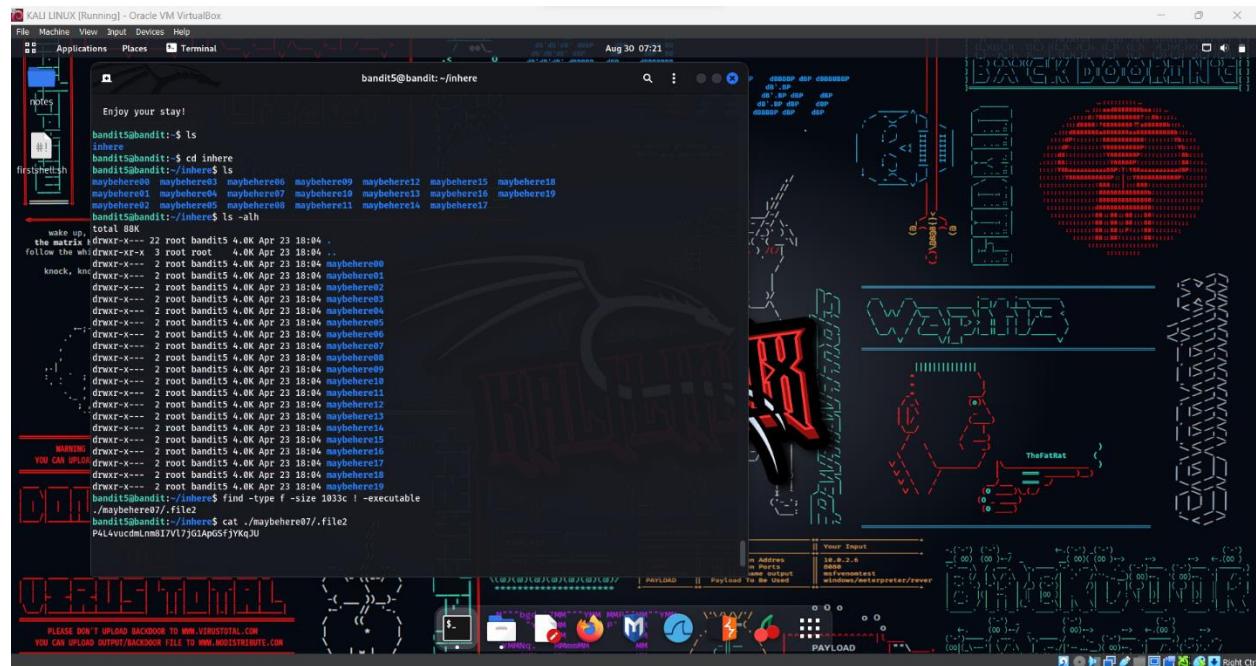
P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU

(find -type f -size 1033c ! -executable)

human-readable

1033 bytes in size

not executable



## LVL 6 – 7

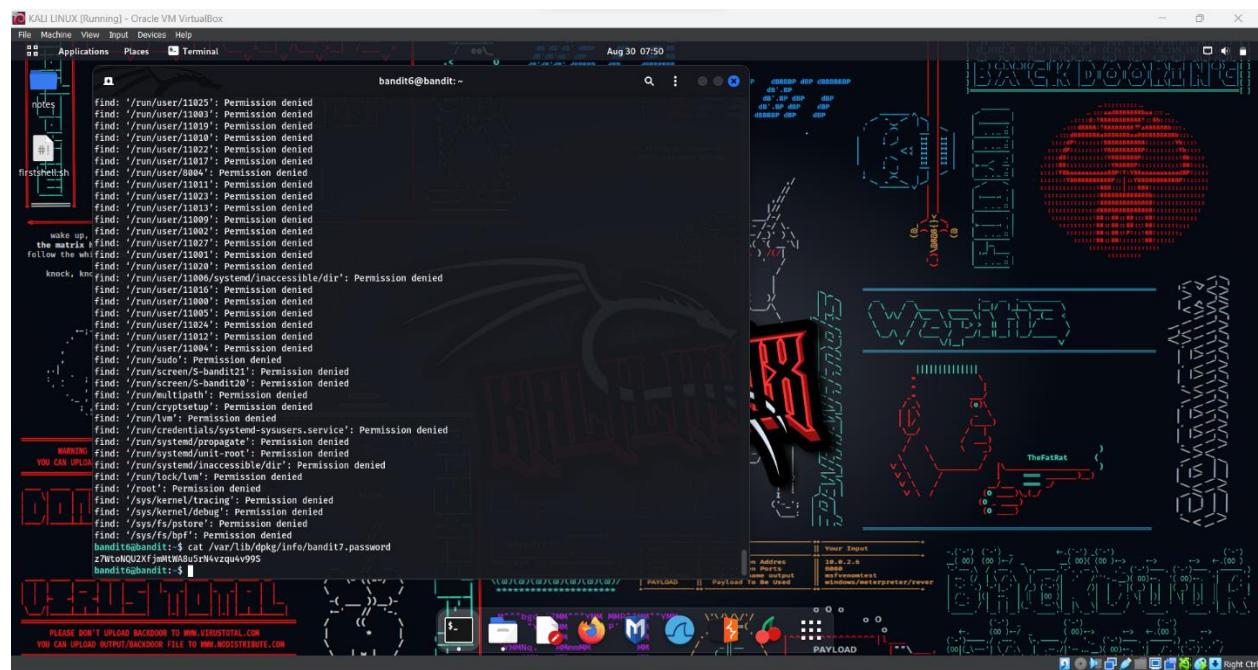
owned by user bandit7

owned by group bandit6

33 bytes in size

z7Wt0NQU2XfjmMtWA8u5rN4vzqu4v99S

(/ -type f -user bandit7 -group bandit6 -size 33c)



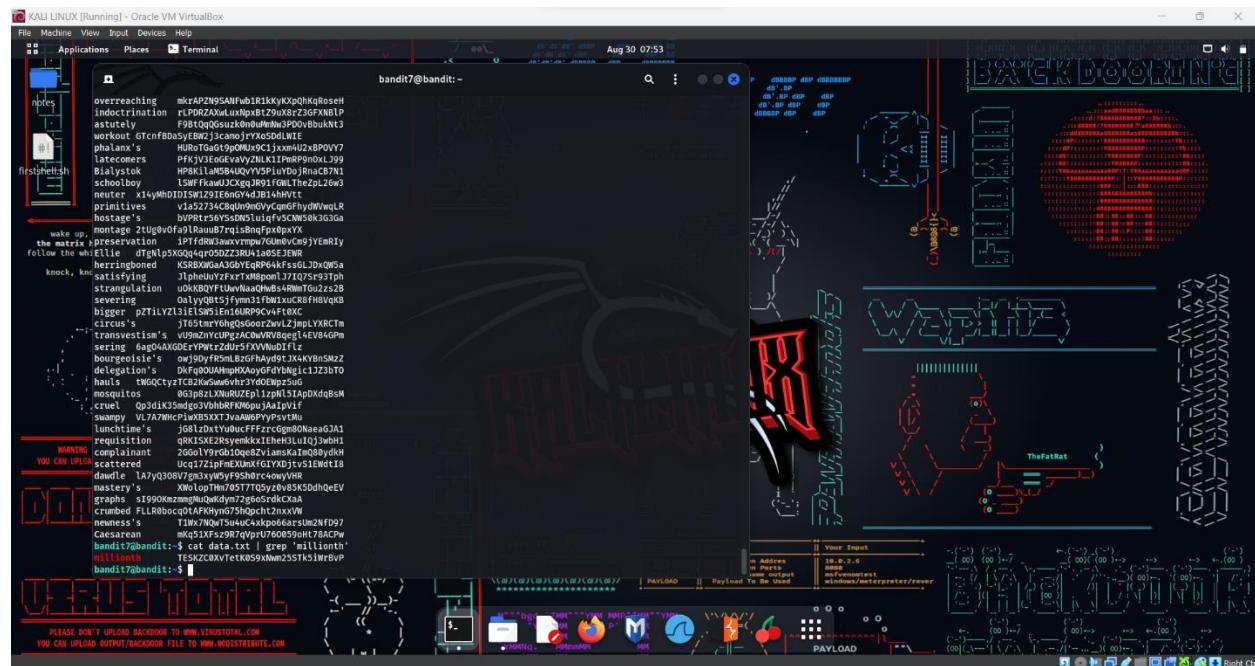
## LVL 7 - 8

TESKZC0XvTetK0S9xNwm25STk5iWrBvP

(strings data.txt | grep "millionth")

(cat data.txt | grep "millionth")

YOU CAN USE ANY TWO OF THIS

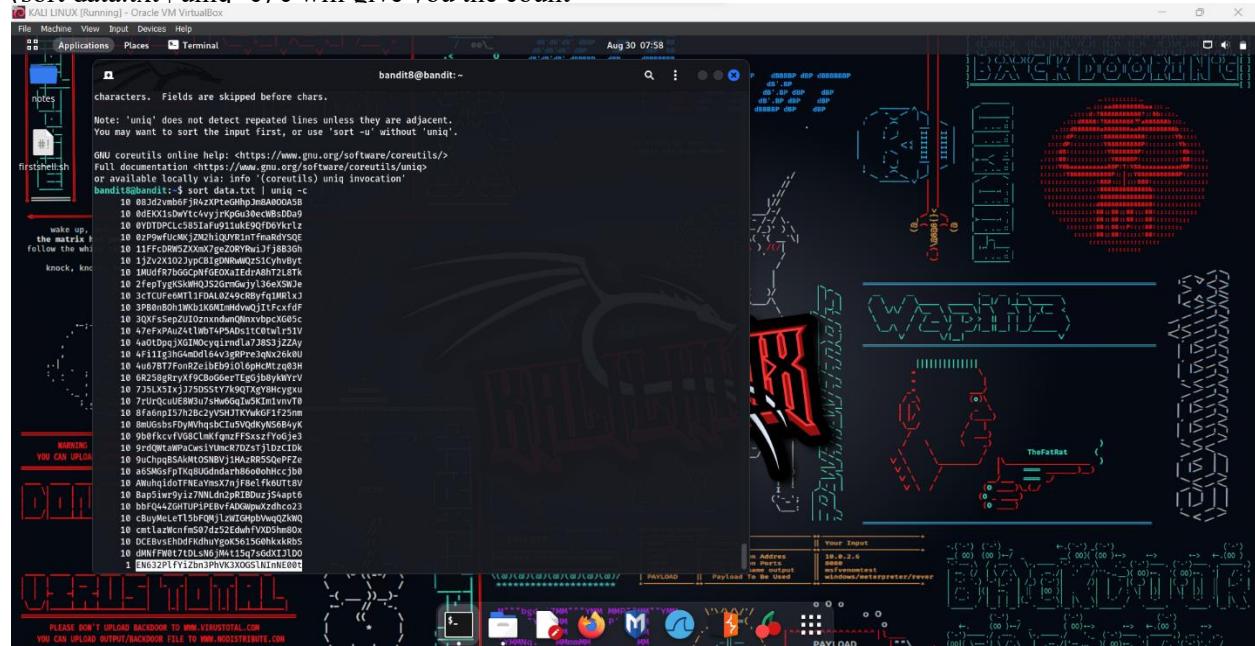


## LVL 8 - 9

IN THIS LEVEL THERE ARE DUPLICATE PASSWORD WE HAVE TO SORT IT AND FIND THE UNIQUE PASSWORD IN ORDER TO GET INTO THE NEXT LEVEL

EN632PlfYiZbn3PhVK3XOGSiNInNE00t

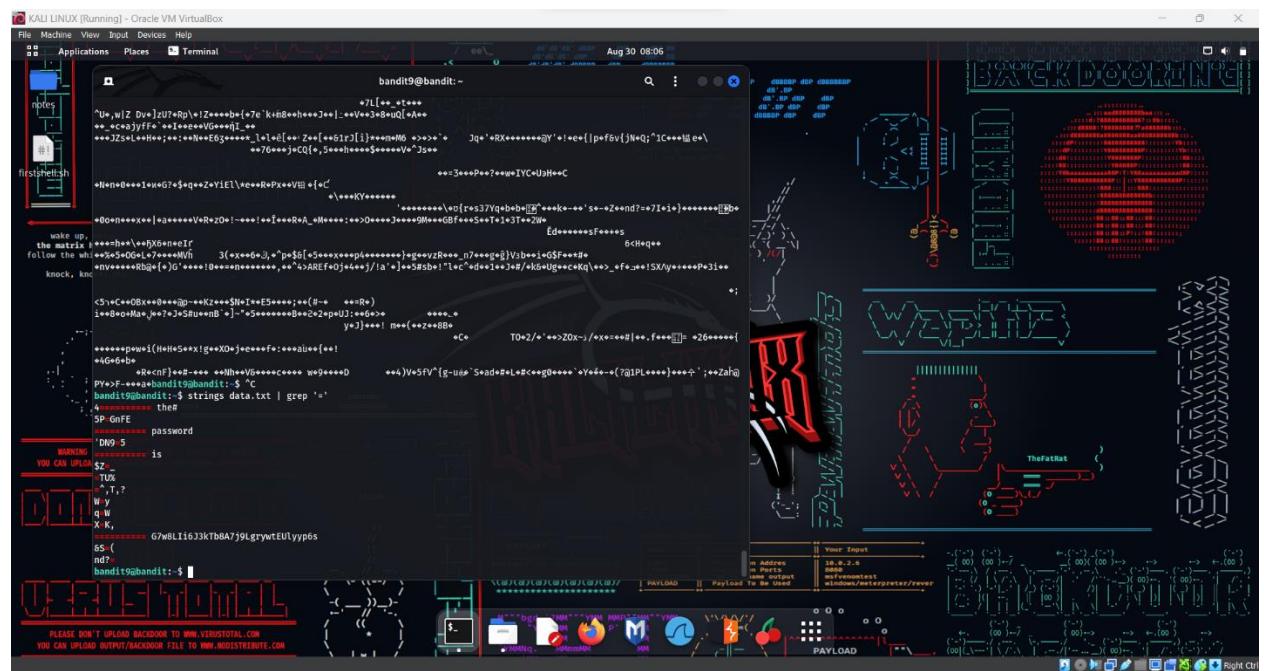
(sort data.txt | uniq -c) c will give you the count



LVL 9 – 10

G7w8LIi6J3kTb8A7j9LgrywtEULyyyp6s

```
(strings data.txt | grep "=")
```



## LVL 10 – 11

IN THIS LEVEL WE HAVE TO DECODE DATA IN ORDER TO GET THE PASSWORD

6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM

(base64 -d data.txt)

-d, --decode            decode data  
-i, --ignore-garbage    when decoding, ignore non-alphabet characters  
-w, --wrap=COLS        wrap encoded lines after COLS character (default 76).

Use 0 to disable line wrapping



## LVL 11 – 12

IN THIS LEVEL WE HAVE TO USE CYBERCHEF WEBSITE AND COPY PASTE THE TEXT WE GET AND GET THE PASSWORD IN ORDER TO REACH NEXT LEVEL.

Use ROT13

JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv

The screenshot shows the CyberChef interface with the following configuration:

- Operations:** ROT13 is selected.
- Recipe:** ROT13 is set with the following options:
  - Rotate lower case chars
  - Rotate upper case chars
  - Rotate numbers
  - Amount: 13
- Input:** The input text is "Gur cnffjbeq vf WIAOSF2MjXXBC0KoSKBbJ8puQm51IEi".
- Output:** The output text is "The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv".



LVL 12 - 13

`xxd` - make a hex dump or do the reverse.

IN THIS LEVEL IDENTIFY THE FILE TYPES (GZIP, BZIP2, TAR, .BIN) COPY, MOVE FILES AND CONVERT EXTRACT FILES TO FIND PASSWORD

Create a directory and copy that data.txt, after that check its file type it's gzip. Now move it to a .gz (create one) and decompress it. Now view the file type It says now bzip2. Again, move it to a bzip2 file (create one) and now check the file type it says now again gzip. Again, do the same thing. Now check the file type it says tar. Again move the file and extract the tar file. Now you got data5.bin it's tar file type.

Again extract the file, now you'll get data6.bin. it's a bzip2 file. Do this over and over again until you get the password this will take quite a while.

wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

```
[bandit12@bandit: /tmp/shenal]$ ./myelf /tmp/shenal
sh: ./myelf: cannot execute file: No such file or directory
[bandit12@bandit: /tmp/shenal]$ rm dir1 /tmp/shenal
[bandit12@bandit: /tmp/shenal]$ cp data.txt /tmp/shenal
[bandit12@bandit: /tmp/shenal]$ ls
data.txt
[bandit12@bandit: /tmp/shenal]$ xxd -r data.txt > data
[bandit12@bandit: /tmp/shenal]$ ls
data.txt
[bandit12@bandit: /tmp/shenal]$ file data
data: gzip compressed data, ASCII text, last modified: Sun Apr 23 18:04:23 2023, max compression, from Unix, original size modulo 2^32 581
[bandit12@bandit: /tmp/shenal]$ gzip -d data.txt.gz
[bandit12@bandit: /tmp/shenal]$ file file.gz
file.gz: gzip compressed data, ASCII text
[bandit12@bandit: /tmp/shenal]$ file file
file: file: file
[bandit12@bandit: /tmp/shenal]$ file file
file: file: file
[bandit12@bandit: /tmp/shenal]$ file file.bin
file: file: file
[bandit12@bandit: /tmp/shenal]$ file formal.bin
formal.bin: formal binary, last modified: Sun Apr 23 18:04:23 2023, max compression, from Unix, original size modulo 2^32 26480
[bandit12@bandit: /tmp/shenal]$ gzip -d file.gz
[bandit12@bandit: /tmp/shenal]$ file file
file: file: file
[bandit12@bandit: /tmp/shenal]$ file data.txt
data.txt: file
[bandit12@bandit: /tmp/shenal]$ tar xf file.tar
[bandit12@bandit: /tmp/shenal]$ mv file.tar
[bandit12@bandit: /tmp/shenal]$ rm file
[bandit12@bandit: /tmp/shenal]$ ls
data.txt
[bandit12@bandit: /tmp/shenal]$ file data5.bin
data5.bin: file data5.bin
[bandit12@bandit: /tmp/shenal]$ file file5.txt
file5.txt: file
[bandit12@bandit: /tmp/shenal]$ rm file5.txt
rm: cannot remove 'file': No such file or directory
[bandit12@bandit: /tmp/shenal]$ ls
data.txt
[bandit12@bandit: /tmp/shenal]$ tar xf data.tar
[bandit12@bandit: /tmp/shenal]$ mv data5.bin data.tar
[bandit12@bandit: /tmp/shenal]$ ls
data.txt
```

```
[4] kali@kali: ~ [Oracle VM VirtualBox] [Terminal] [File] [Machine] [View] [Input] [Devices] [Help] [Applications] [Places] [Terminal] [Aug 30 08:58] bandit12@bandit: /tmp/shenal  
bandit12@bandit: ~ $ mkdir /tmp/shenal  
bandit12@bandit: ~ $ cd /tmp/shenal  
bandit12@bandit: /tmp/shenal $ ls  
data.txt  
bandit12@bandit: /tmp/shenal $ xxd -r data.txt > data  
data  
data.txt  
bandit12@bandit: /tmp/shenal $ ls  
data  
data.txt  
bandit12@bandit: /tmp/shenal $ file data  
data: compressed data, was "data2.hz", last modified: Sun Apr 23 18:04:23 2023, max compression, from Unix, original size modulo 2^32 581  
bandit12@bandit: /tmp/shenal $ mv data file.gz  
bandit12@bandit: /tmp/shenal $ gzip -d file.gz  
bandit12@bandit: /tmp/shenal $ ls  
data.txt  
file  
bandit12@bandit: /tmp/shenal $ file file  
file: POSIX tar archive (GNU)  
bandit12@bandit: /tmp/shenal $ mv file file.tar  
bandit12@bandit: /tmp/shenal $ tar xf file.tar  
bandit12@bandit: /tmp/shenal $ ls  
data.txt  
file  
bandit12@bandit: /tmp/shenal $ file data0.bin  
data0.bin: POSIX tar archive (GNU)  
bandit12@bandit: /tmp/shenal $ rm data.txt  
rm: cannot remove 'file': No such file or directory  
bandit12@bandit: /tmp/shenal $ ls  
data0.bin  
bandit12@bandit: /tmp/shenal $ mv data0.bin data.tar  
bandit12@bandit: /tmp/shenal $ ls  
data.tar  
bandit12@bandit: /tmp/shenal $ tar xf data.tar
```

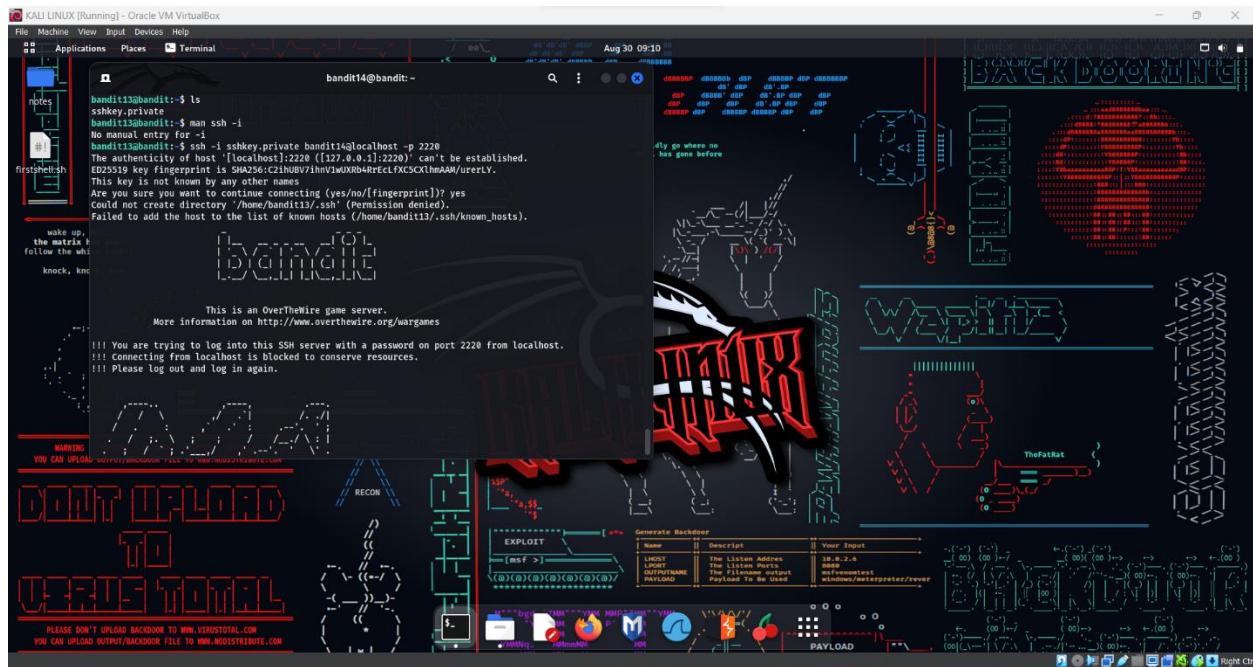
## LVL 13 – 14

```
ssh bandit13@bandit.labs.overthewire.org -p 2220
```

```
ssh -i sshkey.private bandit14@localhost -p 2220
```

```
-i identity_file
```

Selects a file from which the identity (private key) for public key authentication is read.



## LVL 13 – 14

(cat /etc/bandit\_pass/bandit14)

fGrHPx402xGC7U7rXKDaxiWFT0iF0ENq

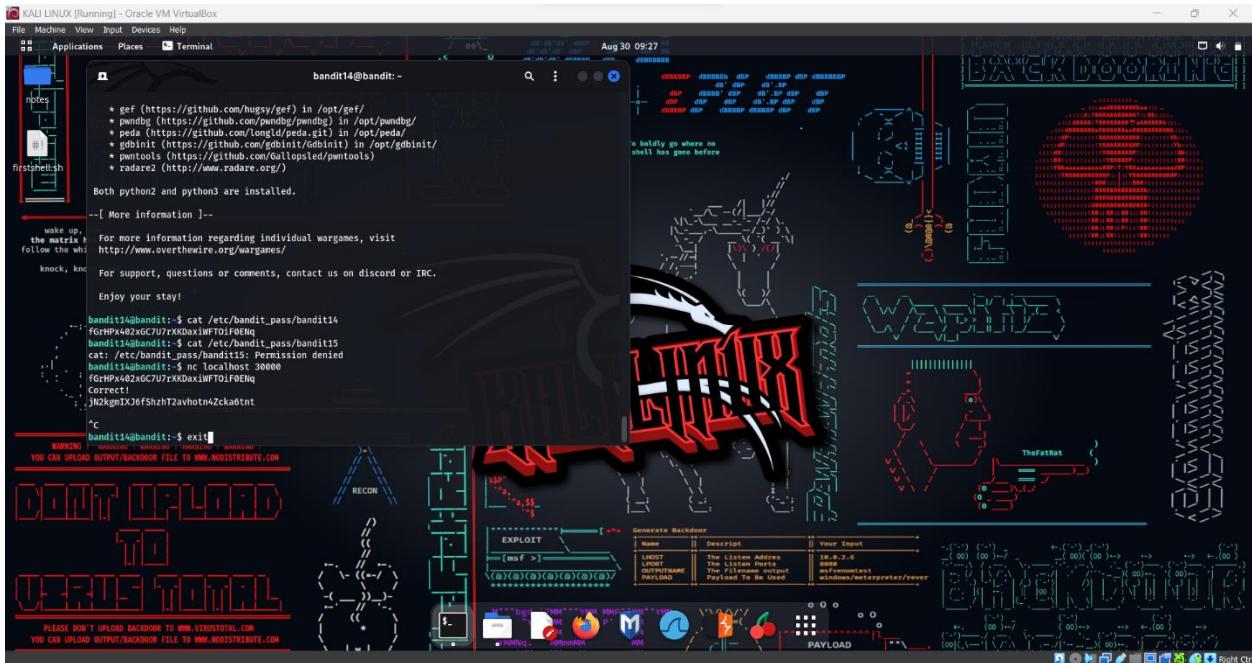


## LVL 14 – 15

nc localhost 30000 and give the previous password then you'll get the new password.

```
cat /etc/bandit_pass/bandit14
```

```
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
```



## LVL 15 -16

ncat - Concatenate and redirect sockets

grep searches for PATTERNs in each FILE. PATTERNs is one or more patterns separated by newline characters, and grep prints each line that matches a pattern. Typically, PATTERNs should be quoted when grep is used in a shell command.

--ssl Connect or listen with SSL

- ssl-cert Specify SSL certificate file (PEM) for listening
- ssl-key Specify SSL private key (PEM) for listening
- ssl-verify Verify trust and domain name of certificates
- ssl-trustfile PEM file containing trusted SSL certificates
- ssl-ciphers Cipherlist containing SSL ciphers to use
- ssl-alpn ALPN protocol list to use.

--ssl (Use SSL)

--ssl-verify (Verify server certificates)

In client mode, --ssl-verify is like --ssl except that it also requires verification of certificates; these will also be used if available. Use --ssl-trustfile to give a custom

--ssl-cert certfile.pem (Specify SSL certificate)

--ssl-key.

--ssl-key keyfile.pem (Specify SSL private key)

certificate named with --ssl-cert.

--ssl-trustfile cert.pem (List trusted certificates)

verification. It has no effect unless combined with --ssl-verify. The argument to this

--ssl-ciphers cipherlist (Specify SSL ciphersuites)

--ssl-alpn ALPN list (Specify ALPN protocol list)

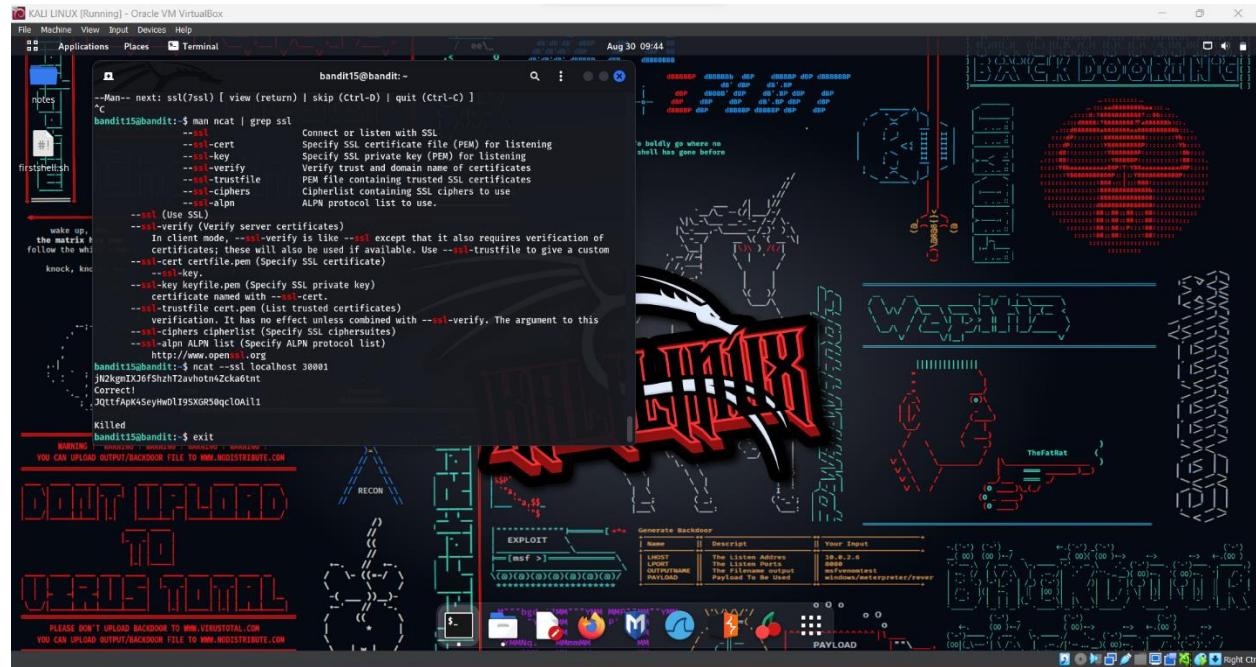
<http://www.openssl.org>

```
cat /etc/bandit_pass/bandit15
```

```
man ncat | grep ssl
```

```
ncat --ssl localhost 30001 and paste old password (previous level) that you have got.
```

```
JQttfApK4SeyHwDlI9SXGR50qclOAII1
```



LVL 16 – 17

```
cat /etc/bandit_pass/bandit16
nmap localhost -p 31000-32000 (-P PORT RANGES)
type this port
ncat --ssl localhost 31790
and you'll get this.

-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMIOf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl87ORiO+rW4LCDNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbK2MBGySxDLrjf0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAOIBABagpxpM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RILwD1NhPx3iBl
J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjF4uNtJom+asvlpms8A
vLY9r60wYSvmZhNqBURj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEAE8JtPxP0GRJ+IQkX262jM3dElkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsgdifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAYpHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFau0ECgYAbjo46T4hyP5tJi93V5HDi
TtieK7xRVxUI+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu
L8ktHMPvodBwNsSBULpG0QKBgBApITfC1HOnWiMGOU3KPwYWt0O6CdTkmJ0mL8Ni
bh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU
YDjHdSOoKvDQNWy6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyZRqaM
77pBAoGAMmjmlJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b
```

vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=

-----END RSA PRIVATE KEY-----

```
ssh -i key bandit17@bandit.labs.overthewire.org -p 2220
```



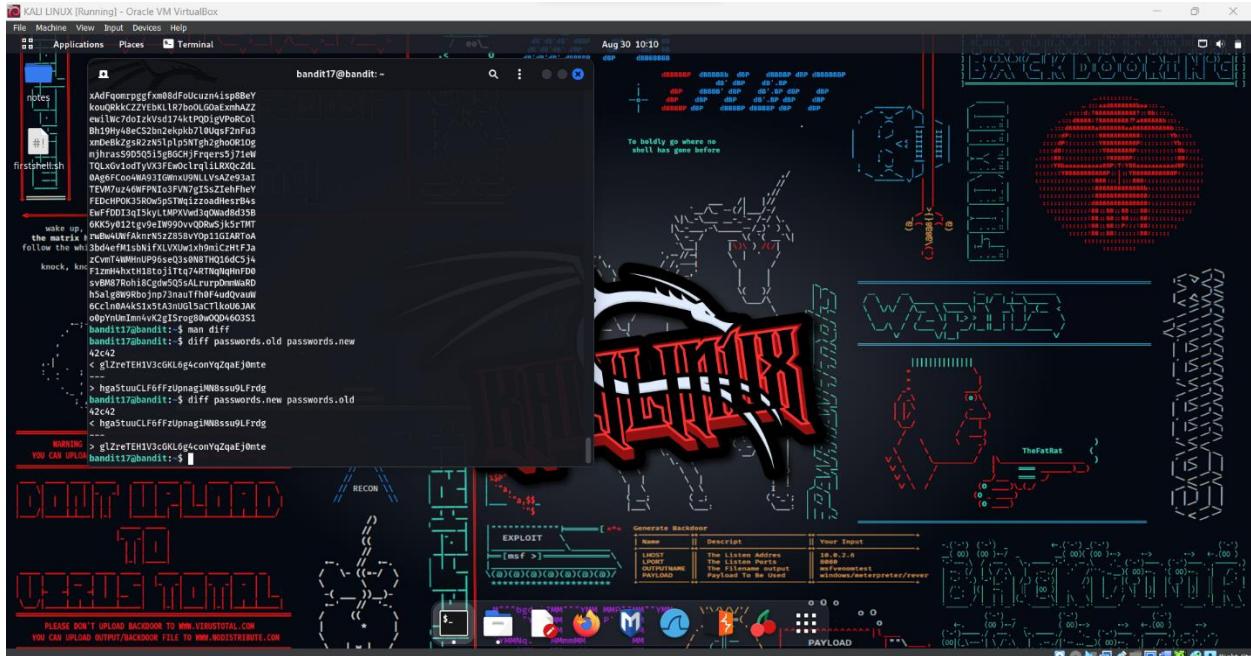
LVL 17-18

There are two file with passwords we need to find password by comparison.

## DIFF - COMPARE FILES LINE BY LINE

diff passwords.new passwords.old

hga5tuuCLF6fFzUpnagiMN8ssu9LFrfg (glZreTEH1V3cGKL6g4conYqZqaEj0mte)



## LVL 18 -19

The password for the next level is stored in a file `readme` in the homedirectory. Unfortunately, someone has modified `.bashrc` to log you out when you log in with SSH.

`man ssh | grep terminal`

`-T` Disable pseudo-terminal allocation.

`-t` Force pseudo-terminal allocation. This can be used to execute arbitrary screen-based programs on a remote machine, which can be very useful, e.g. when implementing

If an interactive session is requested, ssh by default will only request a pseudo-terminal (pty) for interactive sessions when the client has one. The flags `-T` and `-t` can

If a pseudo-terminal has been allocated, the user may use the escape characters noted below.

If no pseudo-terminal has been allocated, the session is transparent and can be used to reliably transfer binary data. On most systems, setting the escape character to

When a pseudo-terminal has been requested, ssh supports a number of functions through the use of an escape character.

`SSHASKPASS` If ssh needs a passphrase, it will read the passphrase from the current terminal if it was run from a terminal. If ssh does not have a terminal asso-

```
ssh -T bandit18@bandit.labs.overthewire.org -p 2220
```

```
awhqdNnAbc1naukrpqDYcF95h7HoMTrC
```

OR

```
ssh -t bandit18@bandit.labs.overthewire.org -p 2220 /bin/sh
```

```
awhqdNnAbc1naukrpqDYcF95h7HoMTrC
```



LVL 19 – 20

```
./bandit20-do cat /etc/bandit_pass/bandit20
```

VxCazJaVykI6W36BkBU0mJTCM8rR95XT

We have to run commands as another user hence we can simply run cat command by using it.



## LVL 20 - 21

NEED 2 TERMINALS TO CREATE A CONNECTION, LOGIN FROM BOTH TERMINALS(for me to understand easily).

```
cat /etc/bandit_pass/bandit20 | nc -l localhost -p 7894
```

```
./suconnect 7894 (this not works some times)
```

Or

```
nc -l -p 21000 OR nc -lvp 9999 (both ways are correct)
```

```
and in the other terminal ./suconnect 21000
```

then paste the previous level password in first terminal then it is correct password for level 21 will paste on the terminal.

it sends the password by the connection and it shows in next terminal

```
NvEJF7oVjkddltPSrdKEF0llh9V1IBcq
```



## LVL 21 - 22

```
ls /etc/cron.d/
```

```
cat /etc/cron.d/cronjob_bandit22
```

```
cat /usr/bin/cronjob_bandit22.sh
```

```
cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
```

```
WdDozAdTM2z9DiFEQ2mGlwngMfj4EZffcar
```

```
File Machine View Input Devices Help
Applications Places Terminal Aug 30 15:44
notes
firstshellsh
wake up, Neo
the matrix has
follow the white
bandit21@bandit:~$ ls -lh
total 24K
knock, knock
drwxr-xr-x  2 root  root   4.0K Apr 23 18:04 .
drwxr-xr-x 70 root  root   2.0K Jan  6 2022 .bash_logout
-rw-r--r--  1 root  root   3.7K Jan  6 2022 .bashrc
-rw-r--r--  1 bandit21 bandit21 33 Apr 23 18:04 .prevpass
-rw-r--r--  1 root  root   807 Jan  6 2022 .profile
bandit21@bandit:~$ ls /etc/cron.d/
cronjob_bandit22 cronjob_bandit22 cronjob_bandit24
cronjob_bandit27 cronjob_bandit27 cronjob_bandit29_root
bandit21@bandit:~$ ls /etc/cron.d/cronjob_bandit22
/etc/cron.d/cronjob_bandit22
bandit21@bandit:~$ cat /etc/cron.d/cronjob_bandit22
#!/bin/bash
@reboot bandit22 /usr/bin/cronjob_bandit22.sh > /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh > /dev/null
bandit21@bandit:~$ cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
WdDozAdTM2z9DiFEQ2mGlwngMfj4EZffcar
bandit21@bandit:~$ exit
#
```

WARNING! YOU CAN UPLOAD BACKDOOR TO WWW.VIRUSTOTAL.COM  
YOU CAN UPLOAD OUTPUT/BACKDOOR FILE TO WWW.VIRUDISTRIBUTE.COM

EXPLOIT Generate Backdoor

|         |                     |                             |
|---------|---------------------|-----------------------------|
| [msf >] | Name:               | Your Input:                 |
| (*)     | Description:        | 192.168.2.5                 |
| (*)     | LISTEN:             | 8080                        |
| (*)     | LISTENPORTNAME:     | http                        |
| (*)     | Payload:            | Windows/Meterpreter/reverse |
| (*)     | Payload To Be Used: |                             |

PAYLOAD

LVL 22 - 23

```
/usr/bin/cronjob_bandit24.sh  
myname=bandit23  
$myname  
echo I am user $myname | md5sum | cut -d '' -f 1  
cat /tmp/8ca319486bfbbc3663ea0fbe81326349  
QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G
```

LVL 23 - 24

get 2 terminals  
(1 terminal)  
cd /etc/cron.d/  
cat /etc/cron.d/cronjob\_bandit24  
cat /usr/bin/cronjob\_bandit24.sh  
ls \*(means all the files) ls c\*(it shows all the files that begin with letter c)  
. is current directory and .. is parent directory  
in here we belong to other category  
ls -l /var/spool/bandit24/foo/file1.sh

(2 terminal)  
mkdir /tmp/shenal111  
vim file.sh

```
#!/bin/bash  
  
cat /etc/bandit_pass/bandit24 > /tmp/shenal111/destination.txt  
  
chmod o+x file1.sh
```

```
chmod o+w .
chmod o+w /tmp/shenal1^C
touch destination0.txt^C (do not execute just ctrl+C)
chmod o+w destination0.txt^C
cp file.sh /var/spool/bandit24/foo
after that quickly ( in first terminal you can view ls -l /var/spool/bandit24/foo/file1.sh)
in 2 terminal cat destination0.txt (then you will get the password)
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar
```

LVL 24 - 25

create a directory mkdir tmp/ex1

nano file.sh

```
#!/bin/bash
for i in {0000..9999}
do
echo "VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar $i"
done
```

./test.sh | nc localhost 30002 | grep -v "Wrong"

p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d

LVL 25 -26

```
cat bandit26.sshkey
ssh -i bandit26.sshkey bandit26@localhost -p2220
file /bin/bash
file /usr/bin/showtext
cat /usr/bin/showtext
log in again with more function
```

v to editor

:e /etc/bandit\_pass/bandit26

7GvcKlw9mC7aUQaPx7nwFstuAIBw1o1

c7GvcKlw9mC7aUQaPx7nwFstuAIBw1o1

to exit q

2 way to get the password

create a shell set shell=/bin/bash and cat the /etc/bandit\_pass/bandit26

LVL 26 -27

first resize(minimize) and loginn to the bandit 26

change - :set shell=/bin/bash

ls - text.text is the bandit 26 logo

./bandit27-do id - executing the file

./bandit27-do id - then we have bandit 26 and 27 permissions.

./bandit27-do cat /etc/bandit\_pass/bandit27

YnQpBuifNMas1hcUFk70ZmqkhUU2EuaS

you cannot logout from the shell because you didn't actually log in to the shell use exit or ctrl + D

LVL 27 -28

here we are connecting to a git hub repository

hit with it the port number 2220

ssh://bandit27-git@localhost:2220/home/bandit27-git/repo

ls

ls

cd repo

cat README

```
ssh://bandit28-git@localhost/home/bandit28-git/repo
```

```
AVanL161y9rsbcJIsFHuw35rjaOM19nR
```

LVL 28 - 29

here we are AGAIN connecting to a git hub repository

hit with it the port number 2220

```
git clone ssh://bandit28-git@localhost/home/bandit28-git/repo
```

```
ls
```

```
cd repo
```

you cant cat the file

```
git log --help
```

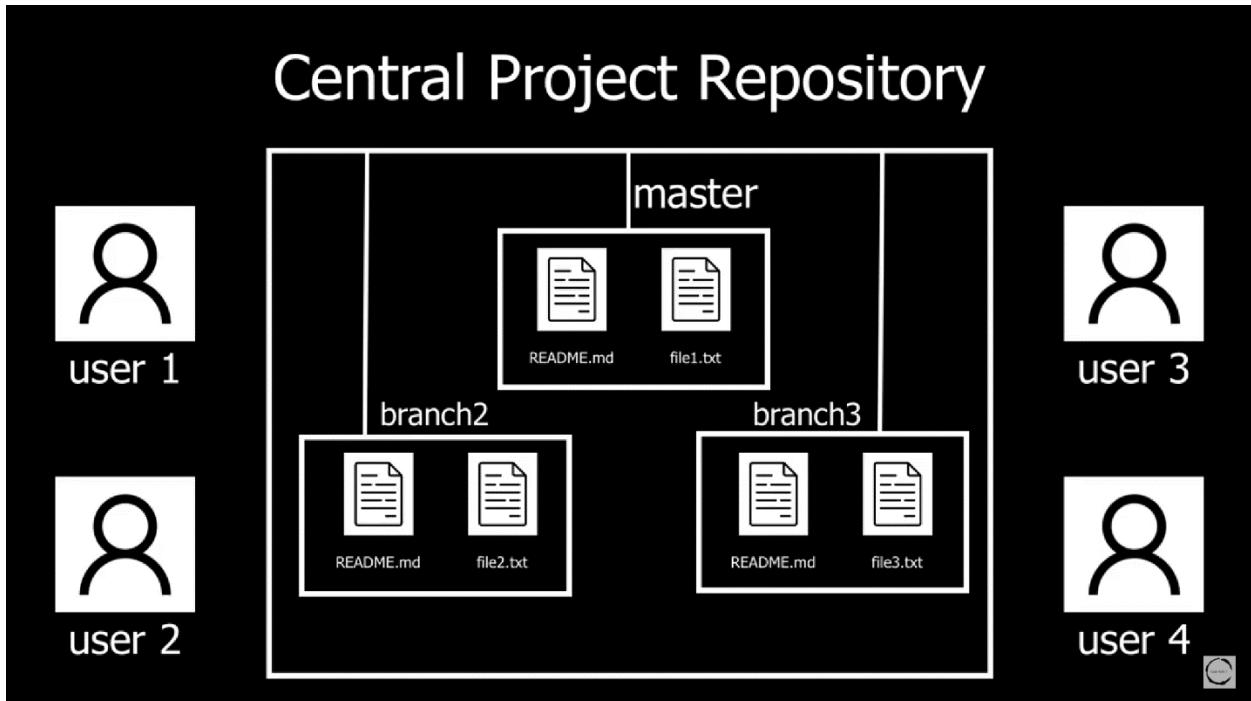
```
git log
```

```
git show 899ba88df296331cc01f30d022c006775d467f28
```

```
tQKvmcwNYcFS6vmPHIUSI3ShmsrQZK8S
```

```
WECHALLUSER="it22883902" WECHALLTOKEN="i have removed my token please enter here"  
wechall
```

LVL 29 - 30



```
mkdir /tmp/mariolvl29  
cd /tmp/mariolvl29  
git clone ssh://bandit29-git@localhost:2220/home/bandit29-git/repo  
git branch  
git branch -a (shows all branches)  
git checkout or git switch (checkout is more than switch)
```

```
git checkout dev  
switching branches an take the password
```

ls , cat

xbhV3HpNGlTIdnjUrdAlPzc2L6y9EOnS

LVL 30 – 31

in this level same ssh as git repo

but there are no passwords in README.md in latest version also there are same branch

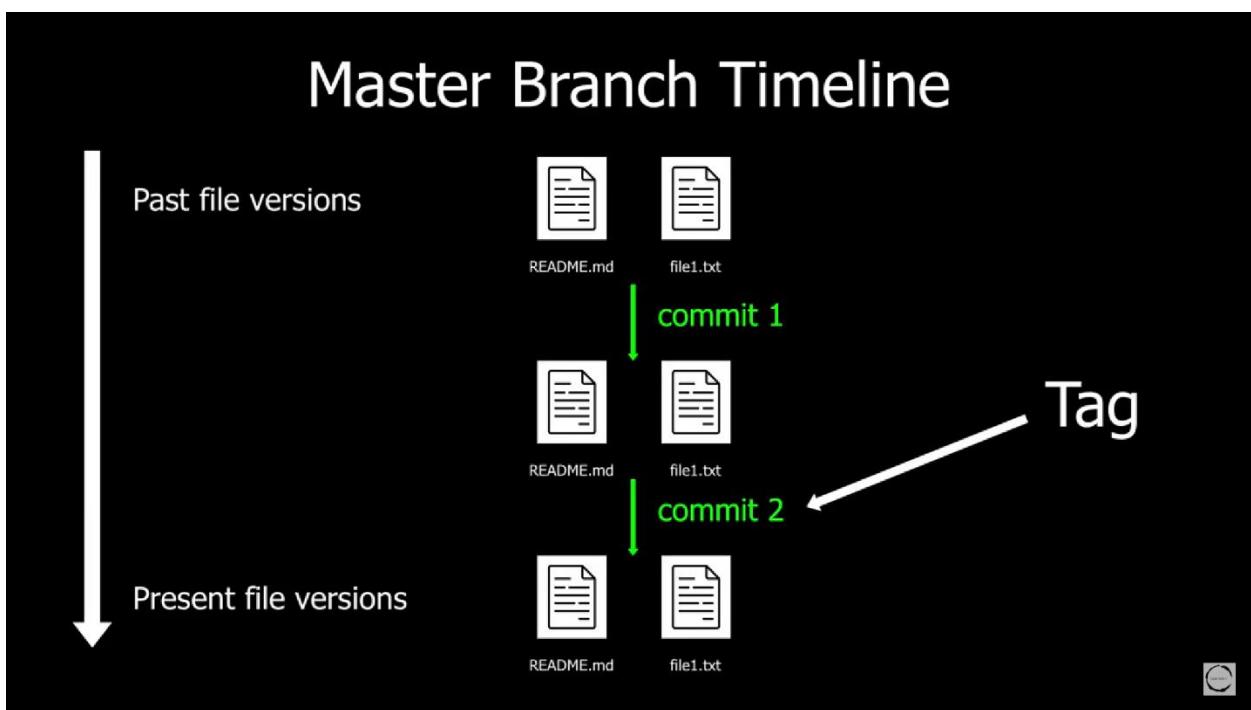
in here we use git tag ( file that have been through several changes {past file ---> present file})

git tag will list all the tags

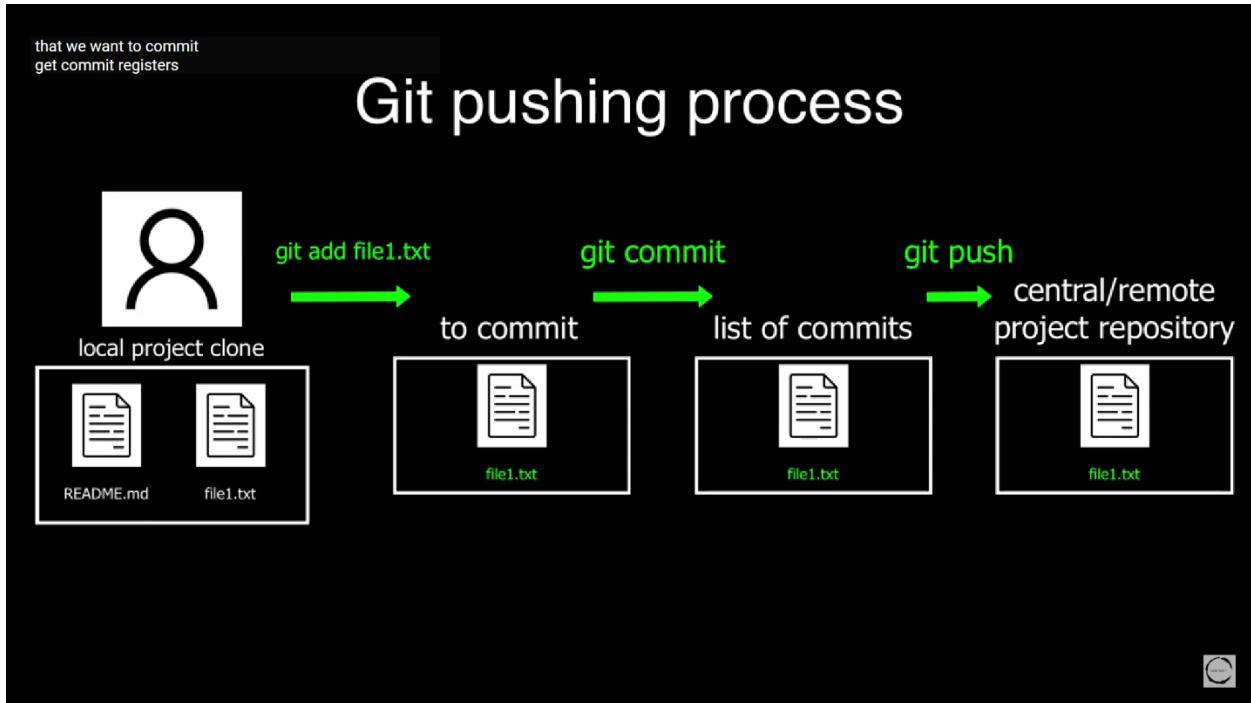
git tag

git show secret

OoffzGDIzhAlerFJ2cAiz1D41JW1Mhmt



LVL 31 - 32



there are no branches

when you cat the README.md it shows some details then you need to create key.txt

```
echo "May I come in?" > key.txt
```

```
git status
```

```
git add key.txt
```

```
cat .gitignore
```

```
git add -f key.txt
```

```
git status
```

```
git commit -m "added key.txt file"
```

```
git status
```

after that you have to use git push

```
git push (then you will get the password)
```

rmCBvG56y58BXzv98yZGdO7ATVL5dW8y