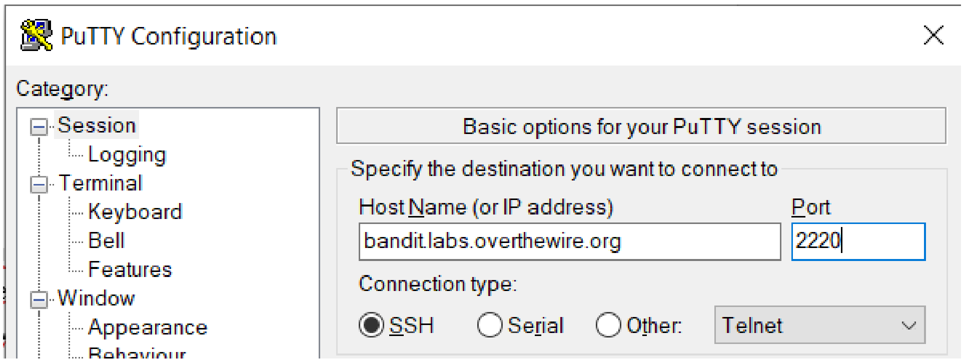
1. Level 0

a) Level Goal

The goal of this level is for you to log into the game using SSH. The host to which you need to connect is bandit.labs.overthewire.org, on port 2220. The username is bandit0 and the password is bandit0. Once logged in, go to the Level 1 page to find out how to beat Level 1.

b) Solution

Wykorzystany zostaje Putty, ustawiony zostaje hostname oraz port zgodnie z zadaniem.



Host Name (or IP address): bandit.labs.overthewire.org

Port: 2220

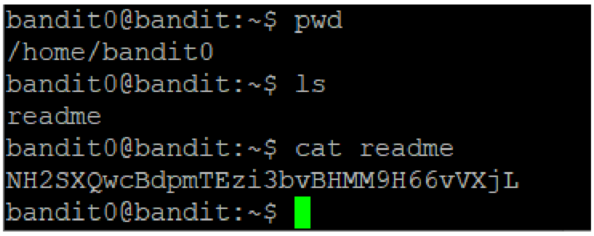
Connection type: ● SSH ○ Serial ○ Other: Telnet

2. Level 0 -> Level 1

a) Level Goal

The password for the next level is stored in a file called **readme** located in the home directory. Use this password to log into bandit1 using SSH. Whenever you find a password for a level, use SSH (on port 2220) to log into that level and continue the game.

b) Solution



```
bandit0@bandit:~$ pwd
/home/bandit0
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL
bandit0@bandit:~$
```

c) Flag

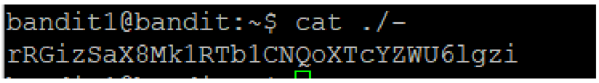
NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL

3. Level 1 -> Level 2

a) Level Goal

The password for the next level is stored in a file called **-** located in the home directory

b) Solution



```
bandit1@bandit:~$ cat ./-
rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi
```

c) Flag

rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi

d) Useful link

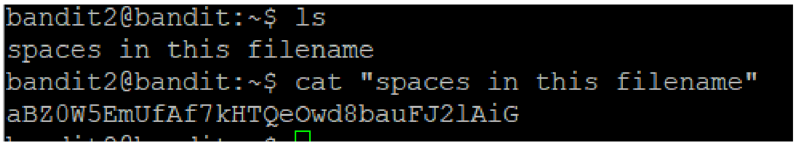
https://www.tecmint.com/special-character-filenames-linux/#Equal-to_in_Filename

4. Level 2 -> Level 3

a) Level Goal

The password for the next level is stored in a file called **spaces in this filename** located in the home directory

b) Solution



```
bandit2@bandit:~$ ls
spaces in this filename
bandit2@bandit:~$ cat "spaces in this filename"
aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG
```

c) Flag

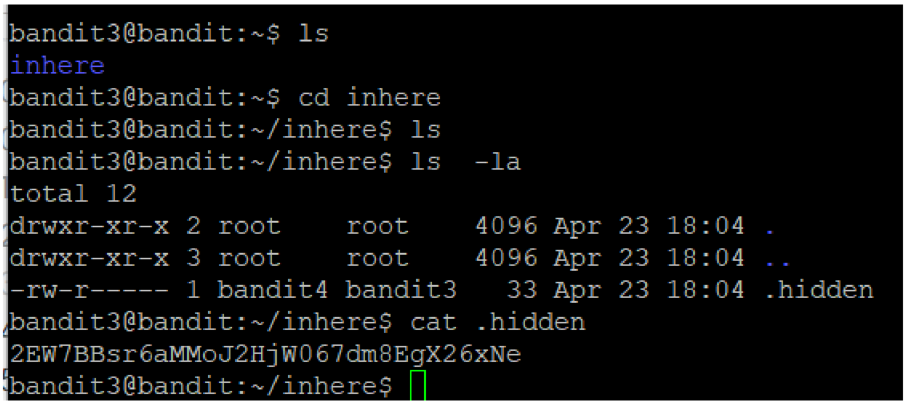
aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG

5. Level 3 -> Level 4

a) Level Goal

The password for the next level is stored in a hidden file in the **inhere** directory.

b) Solution



```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere
bandit3@bandit:~/inhere$ ls
bandit3@bandit:~/inhere$ ls  -la
total 12
drwxr-xr-x 2 root    root    4096 Apr 23 18:04 .
drwxr-xr-x 3 root    root    4096 Apr 23 18:04 ..
-rw-r----- 1 bandit4 bandit3   33 Apr 23 18:04 .hidden
bandit3@bandit:~/inhere$ cat .hidden
2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe
bandit3@bandit:~/inhere$
```

c) Flag

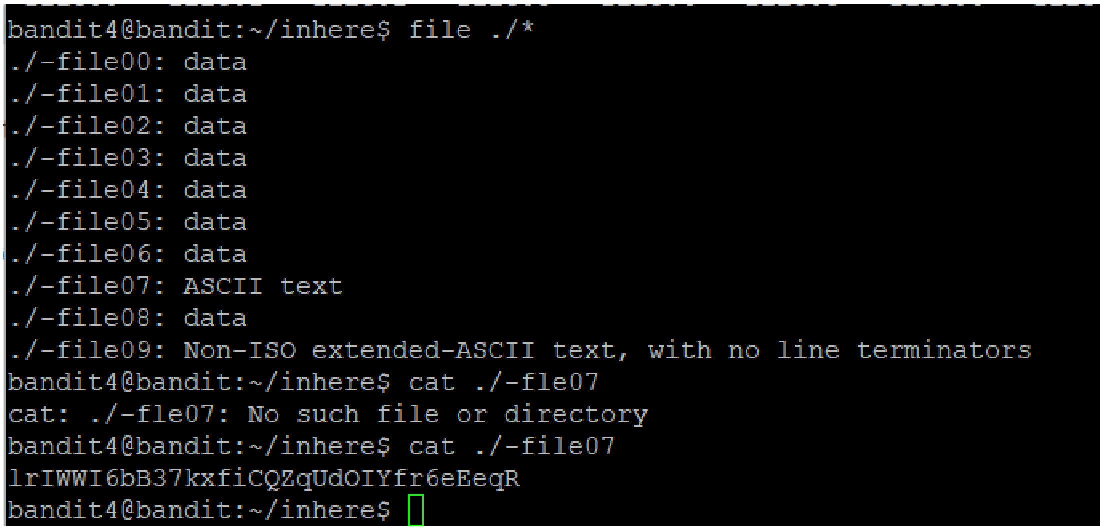
2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe

6. Level 4 -> Level 5

a) Level Goal

The password for the next level is stored in the only human-readable file in the **inhere** directory. Tip: if your terminal is messed up, try the “reset” command.

b) Solution



```
bandit4@bandit:~/inhere$ file ./*
./-file00: data
./-file01: data
./-file02: data
./-file03: data
./-file04: data
./-file05: data
./-file06: data
./-file07: ASCII text
./-file08: data
./-file09: Non-ISO extended-ASCII text, with no line terminators
bandit4@bandit:~/inhere$ cat ./-fle07
cat: ./-fle07: No such file or directory
bandit4@bandit:~/inhere$ cat ./-file07
lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR
bandit4@bandit:~/inhere$ 
```

c) Flag

lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR

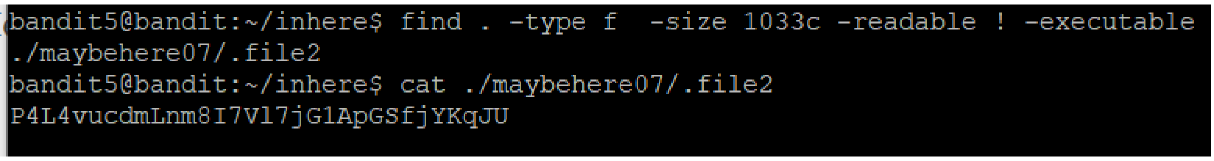
7. Level 5 -> Level 6

a) Level Goal

The password for the next level is stored in a file somewhere under the **inhere** directory and has all of the following properties:

- human-readable
- 1033 bytes in size
- not executable

b) Solution



```
bandit5@bandit:~/inhere$ find . -type f  -size 1033c -readable ! -executable
./maybehere07/.file2
bandit5@bandit:~/inhere$ cat ./maybehere07/.file2
P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU
```

c) Flag

P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU

d) Useful link

https://linuxconfig.org/how-to-use-find-command-to-search-for-files-based-on-file-size

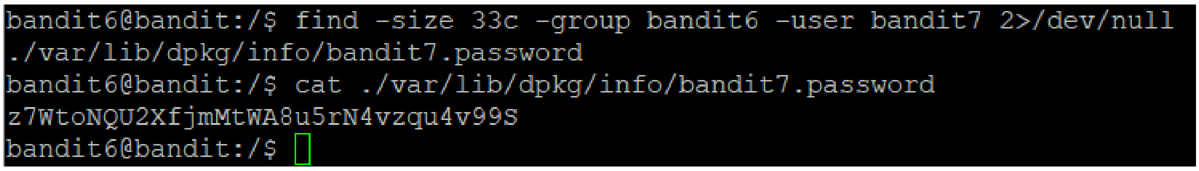
8. Level 6 -> Level 7*

a) Level Goal

The password for the next level is stored **somewhere on the server** and has all of the following properties:

- owned by user bandit7
- owned by group bandit6
- 33 bytes in size

b) Solution



```
bandit6@bandit:/$ find -size 33c -group bandit6 -user bandit7 2>/dev/null
./var/lib/dpkg/info/bandit7.password
bandit6@bandit:/$ cat ./var/lib/dpkg/info/bandit7.password
z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S
bandit6@bandit:/$ 
```

c) Flag

z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S

d) Knowledge

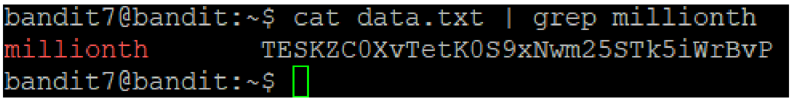
2>/dev/null – wyrzuca errory do katalogu null więc stdout nie jest „zaśmiecony”.

9. Level 7 -> Level 8

a) Level Goal

The password for the next level is stored in the file **data.txt** next to the word **millionth**

b) Solution



```
bandit7@bandit:~$ cat data.txt | grep millionth
millionth       TESKZC0XvTetK0S9xNwm25STk5iWrBvP
bandit7@bandit:~$ 
```

c) Flag

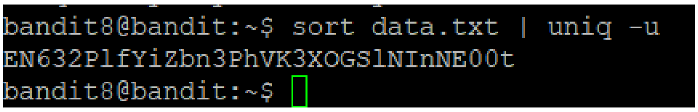
TESKZC0XvTetK0S9xNwm25STk5iWrBvP

10. Level 8 -> Level 9*

a) Level Goal

The password for the next level is stored in the file **data.txt** and is the only line of text that occurs only once

b) Solution



```
bandit8@bandit:~$ sort data.txt | uniq -u
EN632PlfYiZbn3PhVK3XOGSlNInNE00t
bandit8@bandit:~$ 
```

c) Flag

EN632PlfYiZbn3PhVK3XOGSlNInNE00t

d) Knowledge

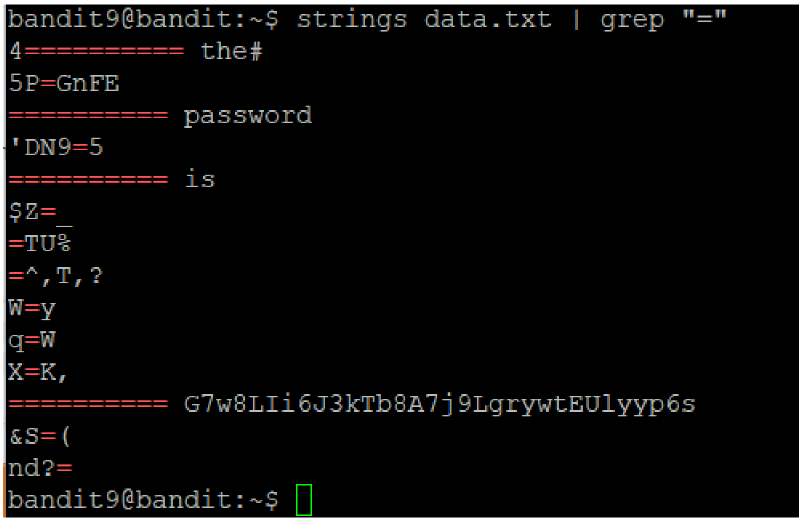
Uniq -u potrzebuje na wejście posortowanego ciągu łańcuchów znakowych.

11. Level 9 -> Level 10

a) Level Goal

The password for the next level is stored in the file **data.txt** in one of the few human-readable strings, preceded by several '=' characters.

b) Solution



```
bandit9@bandit:~$ strings data.txt | grep "="
4========== the#
5P=GnFE
========== password
'DN9=5
========== is
$Z=_
=TU%
=^,T,?
W=y
q=W
X=K,
========== G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s
&S=(
nd?=
bandit9@bandit:~$ 
```

c) Flag

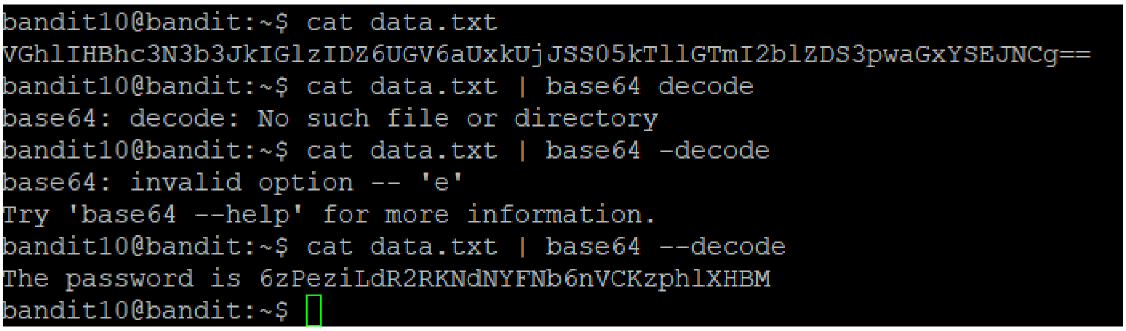
G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s

12. Level 10 -> Level 11

a) Level Goal

The password for the next level is stored in the file **data.txt**, which contains base64 encoded data

b) Solution



```
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b3JkIGlzIDZ6UGV6aUxkUjJSS05kTllGTmI2blZDS3pwaGxYSEJNCg==
bandit10@bandit:~$ cat data.txt | base64 decode
base64: decode: No such file or directory
bandit10@bandit:~$ cat data.txt | base64 -decode
base64: invalid option -- 'e'
Try 'base64 --help' for more information.
bandit10@bandit:~$ cat data.txt | base64 --decode
The password is 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM
bandit10@bandit:~$ 
```

c) Flag

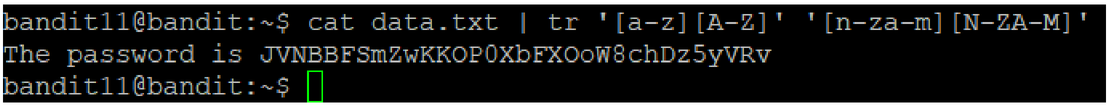
6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM

13. Level 11 -> Level 12

a) Level Goal

The password for the next level is stored in the file **data.txt**, where all lowercase (a-z) and uppercase (A-Z) letters have been rotated by 13 positions

b) Solution



```
bandit11@bandit:~$ cat data.txt | tr '[a-z][A-Z]' '[n-za-m][N-ZA-M]'
The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv
bandit11@bandit:~$ 
```

c) Flag

JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv

d) Useful Link

https://stackoverflow.com/questions/6441260/how-to-shift-each-letter-of-the-string-by-a-given-number-of-letters

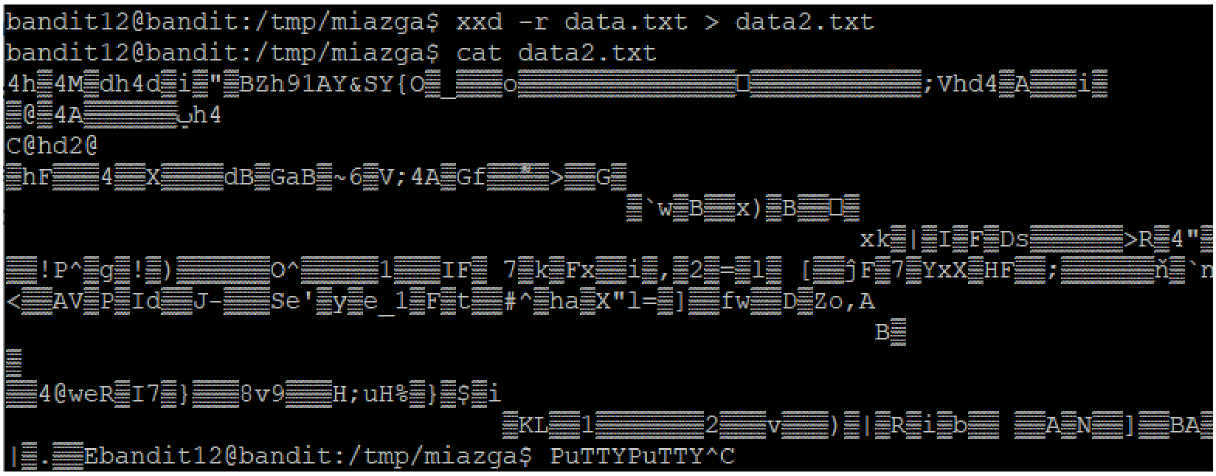
14. Level 12 -> Level13*

a) Level Goal

The password for the next level is stored in the file **data.txt**, which is a hexdump of a file that has been repeatedly compressed. For this level it may be useful to create a directory under /tmp in which you can work using mkdir. For example: mkdir /tmp/myname123. Then copy the datafile using cp, and rename it using mv (read the manpages!)

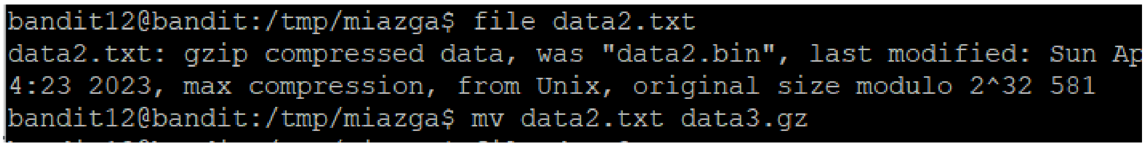
b) Solution

Najpierw cofnięcie operacji hexdump:



```
bandit12@bandit:/tmp/miazga$ xxd -r data.txt > data2.txt
bandit12@bandit:/tmp/miazga$ cat data2.txt
4h▒4M▒dh4d▒i▒"▒BZh91AY&SY{O▒_▒▒▒o▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒;Vhd4▒A▒▒▒i▒
▒@▒4A▒▒▒▒▒▒ٮh4
C@hd2@
▒hF▒▒▒▒4▒▒X▒▒▒▒▒dB▒GaB▒~6▒V;4A▒Gf▒▒▒▒>▒▒▒G▒
                                        ▒`w▒B▒▒x)▒B▒▒▒▒
                                                    xk▒|▒I▒F▒Ds▒▒▒▒▒▒▒>R▒4"▒
▒▒!P^▒g▒!▒)▒▒▒▒▒▒O^▒▒▒▒▒1▒▒▒IF▒ 7▒k▒Fx▒▒i▒,▒2▒=▒1▒ [▒▒ĵF▒7▒YxX▒HF▒▒;▒▒▒▒▒▒ň▒`n
<▒▒AV▒P▒Id▒▒J-▒▒▒Se'▒y▒e_1▒F▒t▒▒#^▒ha▒X"l=▒]▒▒fw▒▒D▒Zo,A
                                                        B▒
▒
▒▒4@weR▒I7▒}▒▒▒8v9▒▒▒H;uH%▒}▒$▒i
                                ▒KL▒▒1▒▒▒▒▒▒▒2▒▒▒v▒▒▒)▒|▒R▒i▒b▒▒ ▒▒A▒N▒▒]▒▒BA▒
|▒.▒▒Ebandit12@bandit:/tmp/miazga$ PuTTYPuTTY^C
```

Następnie za pomocą komendy „file” sprawdzenie informacji o pliku

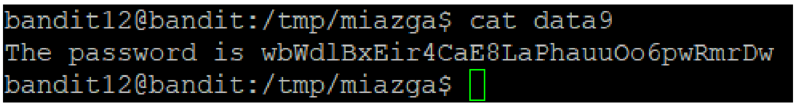


```
bandit12@bandit:/tmp/miazga$ file data2.txt
data2.txt: gzip compressed data, was "data2.bin", last modified: Sun Ap
4:23 2023, max compression, from Unix, original size modulo 2^32 581
bandit12@bandit:/tmp/miazga$ mv data2.txt data3.gz
```

Następnie cofanie operacji kompresji:

- gzip za pomocą gunzip
- bzip2 za pomocą bzip2 -d
- tar za pomocą tar -xf

Powtarzanie operacji aż do momentu uzyskania hasła



```
bandit12@bandit:/tmp/miazga$ cat data9
The password is wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw
bandit12@bandit:/tmp/miazga$ 
```

c) Flag

wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

d) Useful link

https://www.geeksforgeeks.org/hexdump-command-in-linux-with-examples/

https://stackoverflow.com/questions/43724144/hexdump-reverse-command

e) Knowledge

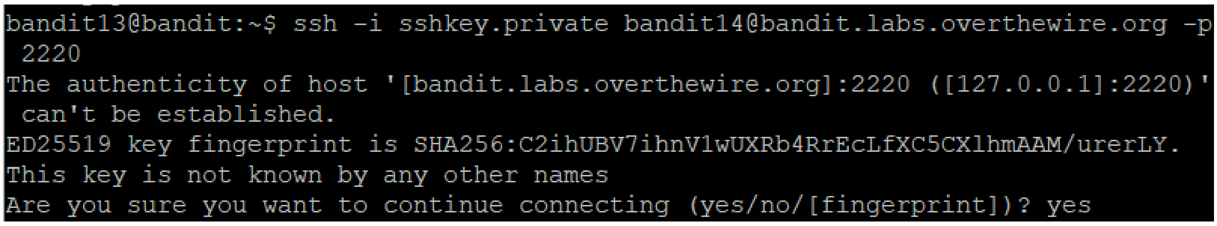
Za pomocą polecenia file można sprawdzić, czy plik uległ kompresji, nawet jeśli nie mamy rozszerzenia.

15. Level13 -> Level 14

a) Level Goal

The password for the next level is stored in **/etc/bandit_pass/bandit14 and can only be read by user bandit14**. For this level, you don't get the next password, but you get a private SSH key that can be used to log into the next level. **Note: localhost** is a hostname that refers to the machine you are working on

b) Solution



```
bandit13@bandit:~$ ssh -i sshkey.private bandit14@bandit.labs.overthewire.org -p
 2220
The authenticity of host '[bandit.labs.overthewire.org]:2220 ([127.0.0.1]:2220)'
 can't be established.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerLY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

c) Flag

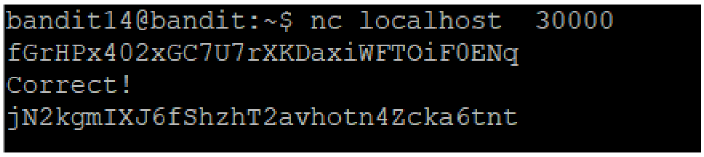
fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq

16. Level14 -> Level15

a) Task Goal

The password for the next level can be retrieved by submitting the password of the current level to **port 30000 on localhost**.

b) Solution



```
bandit14@bandit:~$ nc localhost  30000
fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq
Correct!
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
```

c) Flag

jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt

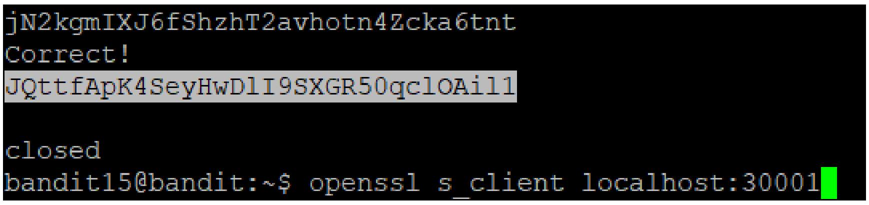
17. Level 15 -> Level 16

a) Level Goal

The password for the next level can be retrieved by submitting the password of the current level to **port 30001 on localhost** using SSL encryption.

Helpful note: Getting "HEARTBEATING" and "Read R BLOCK"? Use -ign_eof and read the "CONNECTED COMMANDS" section in the manpage. Next to 'R' and 'Q', the 'B' command also works in this version of that command…

b) Solution



```
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
Correct!
JQttfApK4SeyHwDlI9SXGR50qclOAil1

closed
bandit15@bandit:~$ openssl s_client localhost:30001
```

c) Flag

JQttfApK4SeyHwDlI9SXGR50qclOAil1

18. Level 16 -> Level 17

a) Level Goal

The credentials for the next level can be retrieved by submitting the password of the current level to **a port on localhost in the range 31000 to 32000**. First find out which of these ports have a server listening on them. Then find out which of those speak SSL and which don't. There is only 1 server that will give the next credentials, the others will simply send back to you whatever you send to it.

b) Solution

Przeskanowałem adresację za pomocą polecenia:

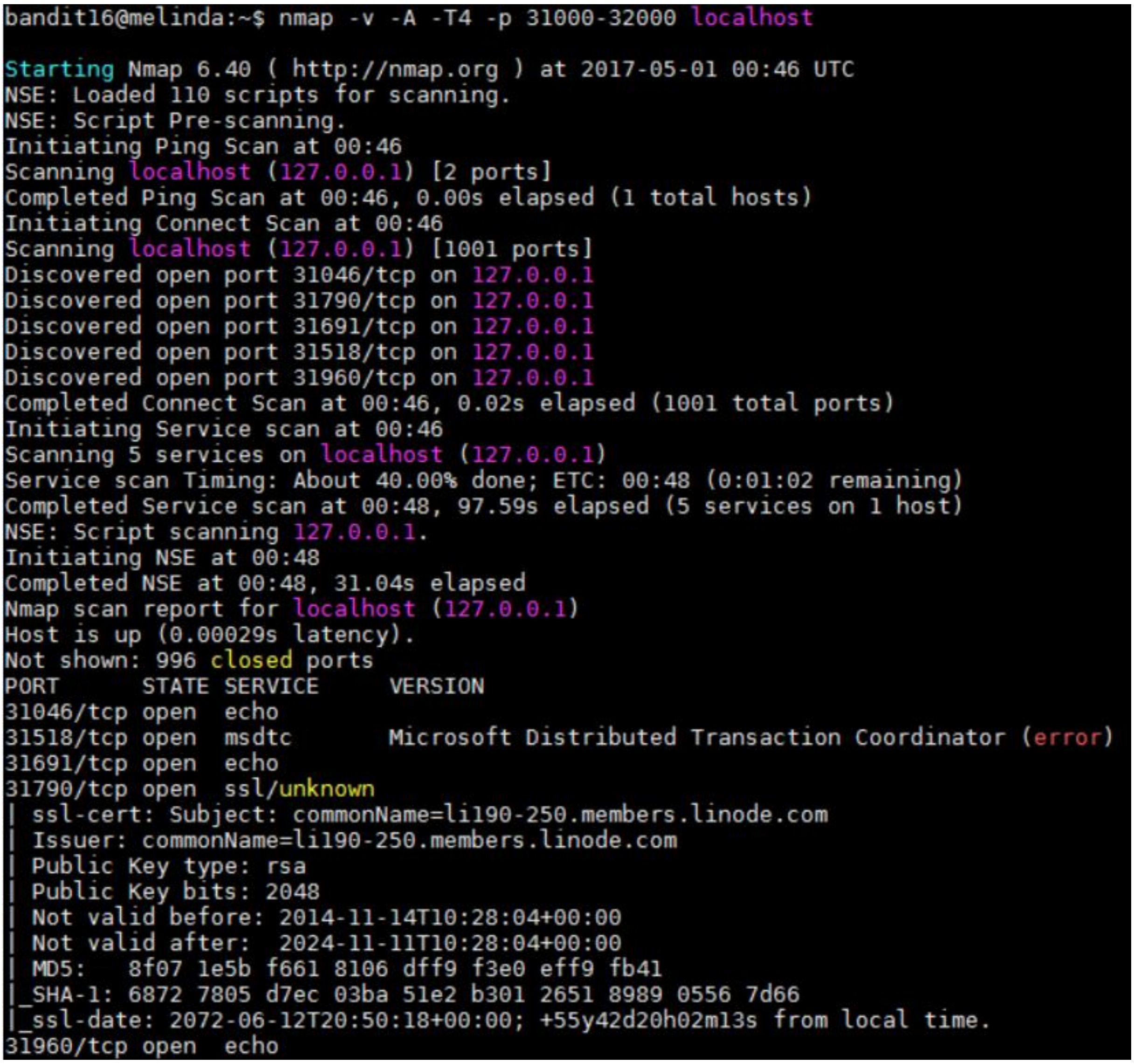
nmap localhost --script ssl-enum-ciphers -p 31000-32000

Dało mi to 2 porty wykorzystujące SSL.

Można było też wykorzystać polecenie:

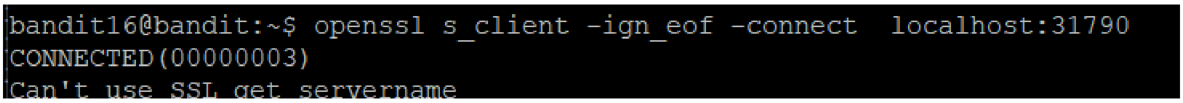
nmap -v -A -T4 -p 31000-32000 localhost

W przypadku tej drugiej od razu bym widział gdzie jest serwis SSL



```
bandit16@melinda:~$ nmap -v -A -T4 -p 31000-32000 localhost

Starting Nmap 6.40 ( http://nmap.org ) at 2017-05-01 00:46 UTC
NSE: Loaded 110 scripts for scanning.
NSE: Script Pre-scanning.
Initiating Ping Scan at 00:46
Scanning localhost (127.0.0.1) [2 ports]
Completed Ping Scan at 00:46, 0.00s elapsed (1 total hosts)
Initiating Connect Scan at 00:46
Scanning localhost (127.0.0.1) [1001 ports]
Discovered open port 31046/tcp on 127.0.0.1
Discovered open port 31790/tcp on 127.0.0.1
Discovered open port 31691/tcp on 127.0.0.1
Discovered open port 31518/tcp on 127.0.0.1
Discovered open port 31960/tcp on 127.0.0.1
Completed Connect Scan at 00:46, 0.02s elapsed (1001 total ports)
Initiating Service scan at 00:46
Scanning 5 services on localhost (127.0.0.1)
Service scan Timing: About 40.00% done; ETC: 00:48 (0:01:02 remaining)
Completed Service scan at 00:48, 97.59s elapsed (5 services on 1 host)
NSE: Script scanning 127.0.0.1.
Initiating NSE at 00:48
Completed NSE at 00:48, 31.04s elapsed
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00029s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE     VERSION
31046/tcp open  echo
31518/tcp open  msdtc       Microsoft Distributed Transaction Coordinator (error)
31691/tcp open  echo
31790/tcp open  ssl/unknown
| ssl-cert: Subject: commonName=li190-250.members.linode.com
| Issuer: commonName=li190-250.members.linode.com
| Public Key type: rsa
| Public Key bits: 2048
| Not valid before: 2014-11-14T10:28:04+00:00
| Not valid after:  2024-11-11T10:28:04+00:00
| MD5:   8f07 1e5b f661 8106 dff9 f3e0 eff9 fb41
|_SHA-1: 6872 7805 d7ec 03ba 51e2 b301 2651 8989 0556 7d66
|_ssl-date: 2072-06-12T20:50:18+00:00; +55y42d20h02m13s from local time.
31960/tcp open  echo
```

Następnie należało połączyć się na znaleziony port/porty za pomocą openssl.



```
bandit16@bandit:~$ openssl s_client -ign_eof -connect  localhost:31790
CONNECTED(00000003)
Can't use SSL_get_servername
```

Po podaniu hasła uzyskiwano klucz prywatny, należało zapisać go do pliku w katalogu tmp.

Następnie należało za pomocą polecenia ssh -i bandit17.key bandit17@localhost -p 2220 połączyć się na kolejne konto, pamiętając o nadaniu odpowiednich uprawnień do pliku zawierającego klucz prywatny.

c) Flag

Brak flagi, logowaliśmy się za pomocą klucza prywatnego z wykorzystaniem ssh.

d) Useful link

https://nmap.org/nsedoc/scripts/ssl-enum-ciphers.html

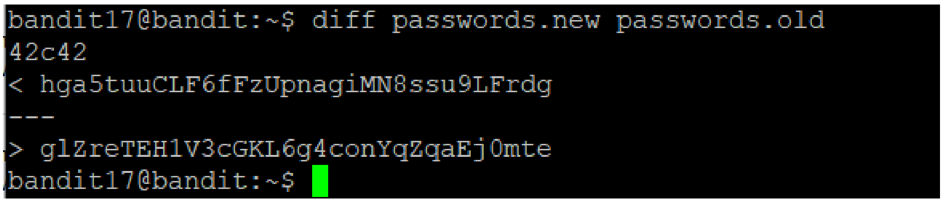
19. Level 17 -> Level 18

a) Level Goal

There are 2 files in the homedirectory: passwords.old and passwords.new. The password for the next level is in passwords.new and is the only line that has been changed between passwords.old and passwords.new

NOTE: if you have solved this level and see 'Byebye!' when trying to log into bandit18, this is related to the next level, bandit19

b) Solution



```
bandit17@bandit:~$ diff passwords.new passwords.old
42c42
< hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg
---
> glZreTEH1V3cGKL6g4conYqZqaEj0mte
bandit17@bandit:~$ 
```

c) Flag

hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg

d) Useful link

https://linuxhint.com/compare-two-files-linux/

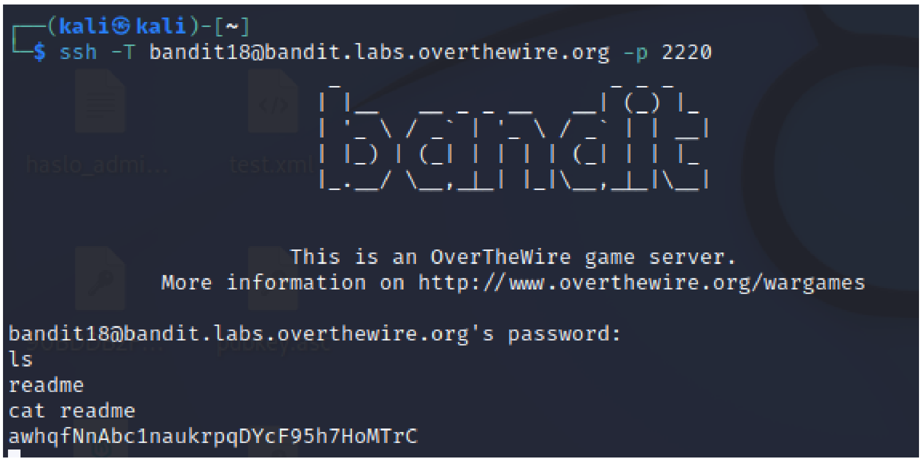
20. Level 18 -> Level 19*

a) Task Goal

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.

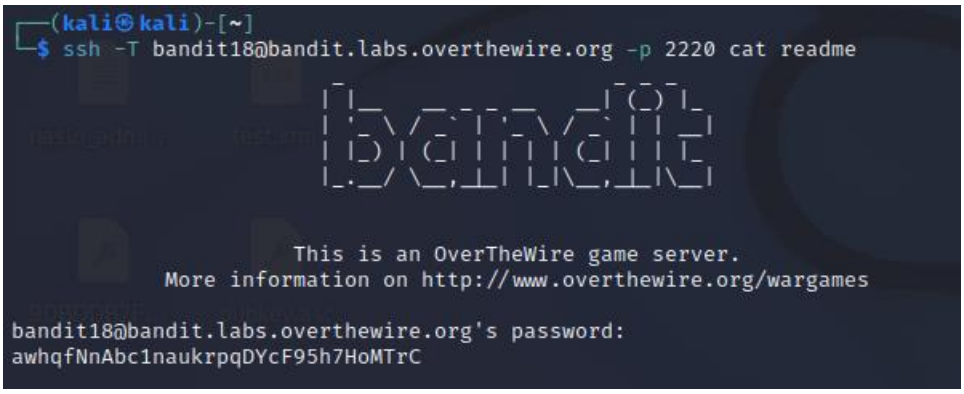
b) Solution

Wykorzystanie opcji „-T" przy łączeniu się poprzez ssh:



```
┌──(kali㉿kali)-[~]
└─$ ssh -T bandit18@bandit.labs.overthewire.org -p 2220
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

bandit18@bandit.labs.overthewire.org's password:
ls
readme
cat readme
awhqfNnAbc1naukrpqDYcF95h7HoMTrC
```

Lub dodanie polecenia na końcu komendy SSH:



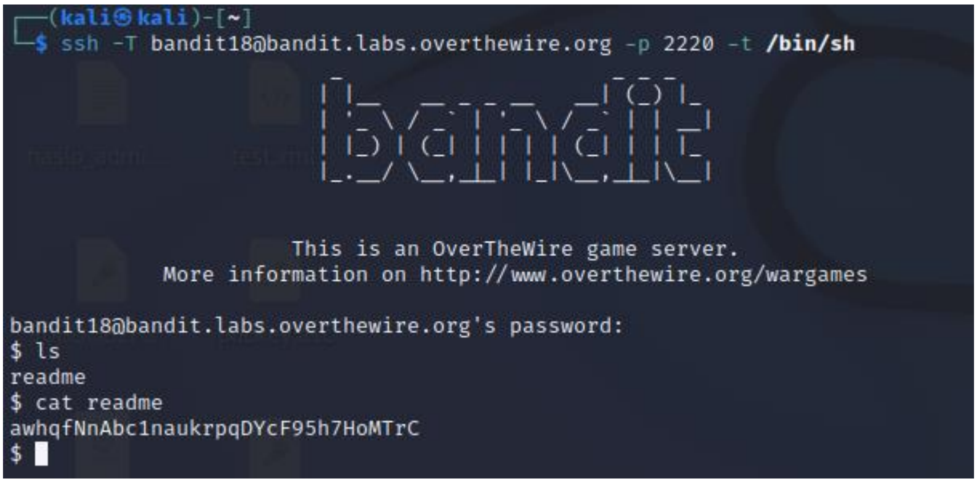
```
┌──(kali㉿kali)-[~]
└─$ ssh -T bandit18@bandit.labs.overthewire.org -p 2220 cat readme
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

bandit18@bandit.labs.overthewire.org's password:
awhqfNnAbc1naukrpqDYcF95h7HoMTrC
```

Wyjaśnienie:

'.bashrc' is a file that is run every time a terminal is loaded. This means it is also run when logging in through SSH because this also loads a terminal.

In the walkthrough to Level 0 I have given a short introduction to SSH. Something I have not mentioned is that SSH does not just allows us to log into a machine remotely, but it also allows remote execution of commands by adding the commands after the common SSH expression.

Alternatively, you could use the same method of executing a command with SSH but use /bin/bash as a command to spawn a bash shell or use the -t flag, which allows a 'pseudo-terminal' to run on the target machine, this way we can run \bin\sh. This is especially useful if we have to do multiple commands because we do not need to repeat the SSH statement and password.



```
┌──(kali㉿kali)-[~]
└─$ ssh -T bandit18@bandit.labs.overthewire.org -p 2220 -t /bin/sh
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_| | | |_
                        |_.__/ \__,_|_| |_|\__,_|_|\__|


                      This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

bandit18@bandit.labs.overthewire.org's password:
$ ls
readme
$ cat readme
awhqfNnAbc1naukrpqDYcF95h7HoMTrC
$ █
```

c) Flag

awhqfNnAbc1naukrpqDYcF95h7HoMTrC

d) Useful link:

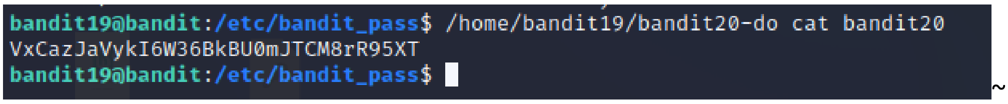
https://mayadevbe.me/posts/overthewire/bandit/level19/

21. Level 19 -> Level 20

a) Task Goal

To gain access to the next level, you should use the setuid binary in the homedirectory. Execute it without arguments to find out how to use it. The password for this level can be found in the usual place (/etc/bandit_pass), after you have used the setuid binary.

b) Solution



```
bandit19@bandit:/etc/bandit_pass$ /home/bandit19/bandit20-do cat bandit20
VxCazJaVykI6W36BkBU0mJTCM8rR95XT
bandit19@bandit:/etc/bandit_pass$ 
```

~

c) Flag

VxCazJaVykI6W36BkBU0mJTCM8rR95XT

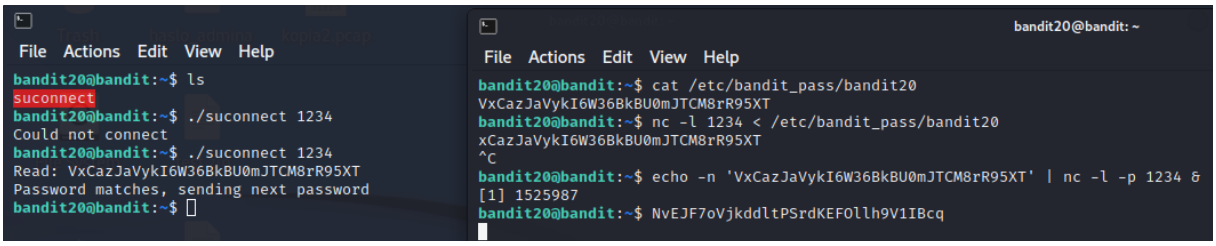
22. Level 20 -> Level 21*

a) Level Goal

There is a setuid binary in the homedirectory that does the following: it makes a connection to localhost on the port you specify as a commandline argument. It then reads a line of text from the connection and compares it to the password in the previous level (bandit20). If the password is correct, it will transmit the password for the next level (bandit21).

NOTE: Try connecting to your own network daemon to see if it works as you think

b) Solution



```
bandit20@bandit:~$ ls
suconnect
bandit20@bandit:~$ ./suconnect 1234
Could not connect
bandit20@bandit:~$ ./suconnect 1234
Read: VxCazJaVykI6W36BkBU0mJTCM8rR95XT
Password matches, sending next password
bandit20@bandit:~$ 
```

```
bandit20@bandit:~$ cat /etc/bandit_pass/bandit20
VxCazJaVykI6W36BkBU0mJTCM8rR95XT
bandit20@bandit:~$ nc -l 1234 < /etc/bandit_pass/bandit20
xCazJaVykI6W36BkBU0mJTCM8rR95XT
^C
bandit20@bandit:~$ echo -n 'VxCazJaVykI6W36BkBU0mJTCM8rR95XT' | nc -l -p 1234 &
[1] 1525987
bandit20@bandit:~$ NvEJF7oVjkddltPSrdKEFOllh9V1IBcq
```

c) Flag

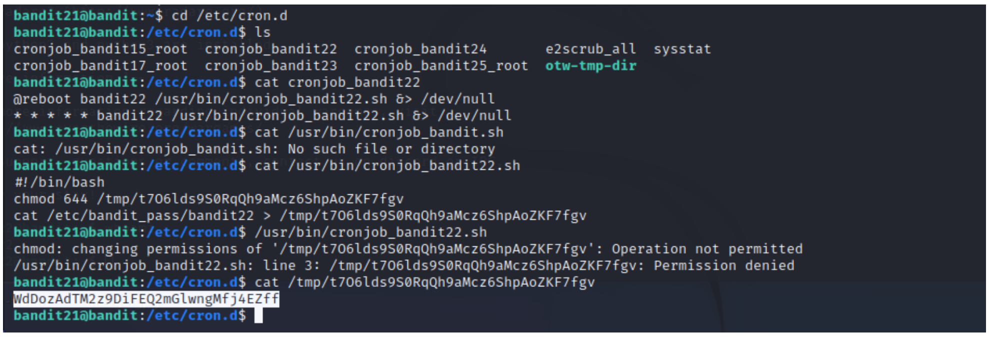
NvEJF7oVjkddltPSrdKEFOllh9V1IBcq

23. Level 21 -> Level 22

a) Level Goal

A program is running automatically at regular intervals from cron, the time-based job scheduler. Look in /etc/cron.d/ for the configuration and see what command is being executed.

b) Solution



```
bandit21@bandit:~$ cd /etc/cron.d
bandit21@bandit:/etc/cron.d$ ls
cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       e2scrub_all  sysstat
cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root  otw-tmp-dir
bandit21@bandit:/etc/cron.d$ cat cronjob_bandit22
@reboot bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
* * * * * bandit22 /usr/bin/cronjob_bandit22.sh &> /dev/null
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit.sh
cat: /usr/bin/cronjob_bandit.sh: No such file or directory
bandit21@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit22.sh
#!/bin/bash
chmod 644 /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
cat /etc/bandit_pass/bandit22 > /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
bandit21@bandit:/etc/cron.d$ /usr/bin/cronjob_bandit22.sh
chmod: changing permissions of '/tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv': Operation not permitted
/usr/bin/cronjob_bandit22.sh: line 3: /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv: Permission denied
bandit21@bandit:/etc/cron.d$ cat /tmp/t7O6lds9S0RqQh9aMcz6ShpAoZKF7fgv
WdDozAdTM2z9DiFEQ2mGlwngMfj4EZff
bandit21@bandit:/etc/cron.d$ 
```

c) Flag

WdDozAdTM2z9DiFEQ2mGlwngMfj4EZff

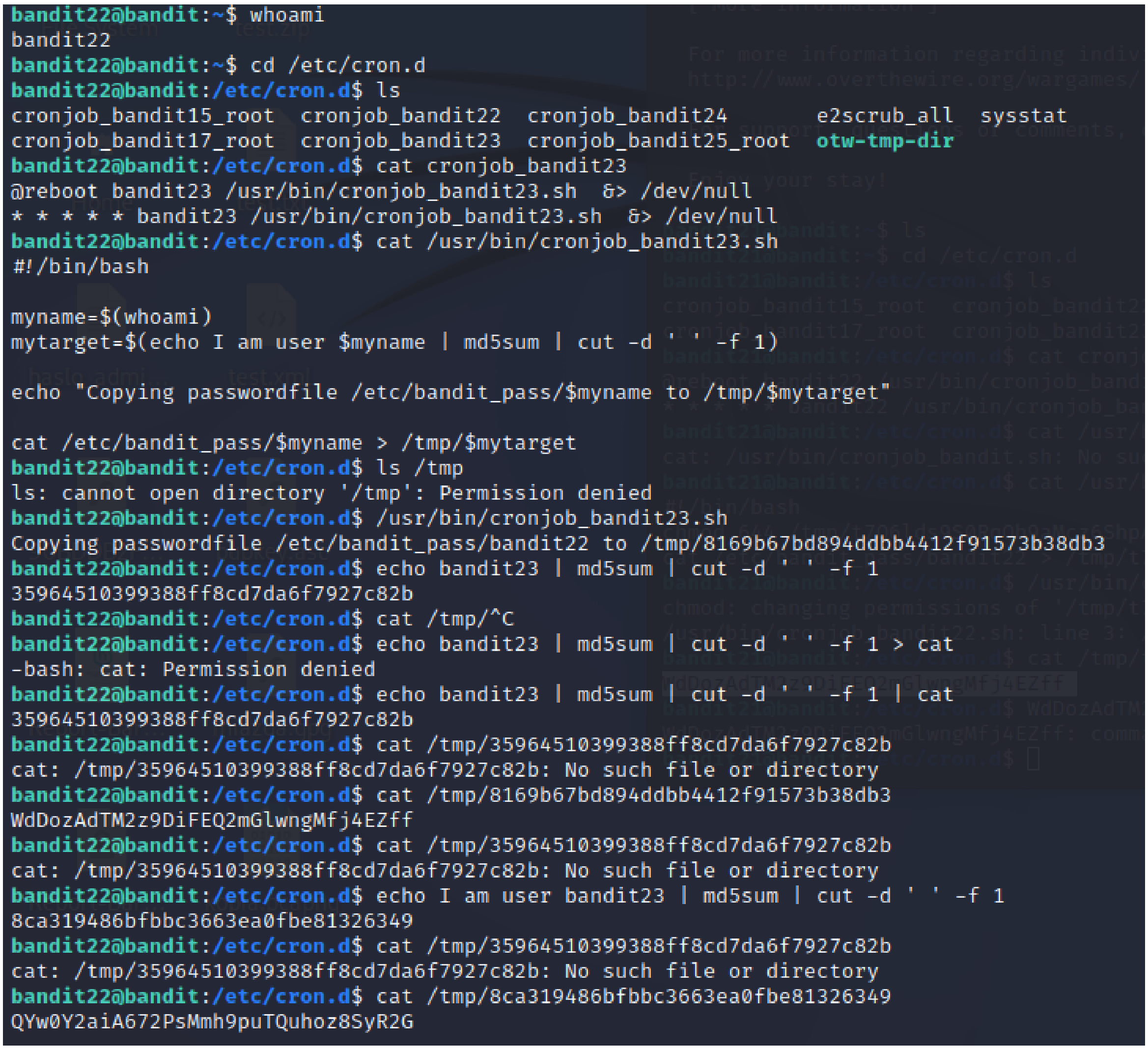
24. Level 22 -> Level 23

a) Level Goal

A program is running automatically at regular intervals from cron, the time-based job scheduler. Look in /etc/cron.d/ for the configuration and see what command is being executed.

NOTE: Looking at shell scripts written by other people is a very useful skill. The script for this level is intentionally made easy to read. If you are having problems understanding what it does, try executing it to see the debug information it prints.

b) Solution



```
bandit22@bandit:~$ whoami
bandit22
bandit22@bandit:~$ cd /etc/cron.d
bandit22@bandit:/etc/cron.d$ ls
cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       e2scrub_all  sysstat
cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root  otw-tmp-dir
bandit22@bandit:/etc/cron.d$ cat cronjob_bandit23
@reboot bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
* * * * * bandit23 /usr/bin/cronjob_bandit23.sh  &> /dev/null
bandit22@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit23.sh
#!/bin/bash

myname=$(whoami)
mytarget=$(echo I am user $myname | md5sum | cut -d ' ' -f 1)

echo "Copying passwordfile /etc/bandit_pass/$myname to /tmp/$mytarget"

cat /etc/bandit_pass/$myname > /tmp/$mytarget
bandit22@bandit:/etc/cron.d$ ls /tmp
ls: cannot open directory '/tmp': Permission denied
bandit22@bandit:/etc/cron.d$ /usr/bin/cronjob_bandit23.sh
Copying passwordfile /etc/bandit_pass/bandit22 to /tmp/8169b67bd894ddbb4412f91573b38db3
bandit22@bandit:/etc/cron.d$ echo bandit23 | md5sum | cut -d ' ' -f 1
35964510399388ff8cd7da6f7927c82b
bandit22@bandit:/etc/cron.d$ cat /tmp/^C
bandit22@bandit:/etc/cron.d$ echo bandit23 | md5sum | cut -d ' ' -f 1 > cat
-bash: cat: Permission denied
bandit22@bandit:/etc/cron.d$ echo bandit23 | md5sum | cut -d ' ' -f 1 | cat
35964510399388ff8cd7da6f7927c82b
bandit22@bandit:/etc/cron.d$ cat /tmp/35964510399388ff8cd7da6f7927c82b
cat: /tmp/35964510399388ff8cd7da6f7927c82b: No such file or directory
bandit22@bandit:/etc/cron.d$ cat /tmp/8169b67bd894ddbb4412f91573b38db3
WdDozAdTM2z9DiFEQ2mGlwngMfj4EZff
bandit22@bandit:/etc/cron.d$ cat /tmp/35964510399388ff8cd7da6f7927c82b
cat: /tmp/35964510399388ff8cd7da6f7927c82b: No such file or directory
bandit22@bandit:/etc/cron.d$ echo I am user bandit23 | md5sum | cut -d ' ' -f 1
8ca319486bfbbc3663ea0fbe81326349
bandit22@bandit:/etc/cron.d$ cat /tmp/35964510399388ff8cd7da6f7927c82b
cat: /tmp/35964510399388ff8cd7da6f7927c82b: No such file or directory
bandit22@bandit:/etc/cron.d$ cat /tmp/8ca319486bfbbc3663ea0fbe81326349
QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G
```

c) Flag

QYw0Y2aiA672PsMmh9puTQuhoz8SyR2G

25. Level 23 -> Level 24

a) Level Goal

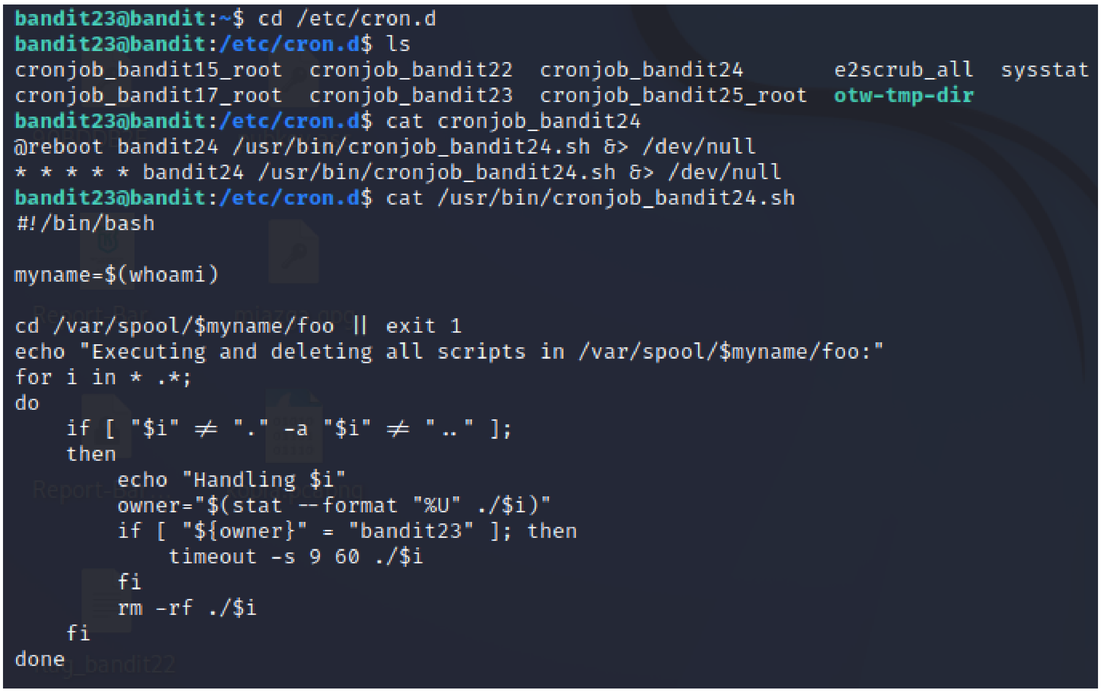
A program is running automatically at regular intervals from cron, the time-based job scheduler. Look in /etc/cron.d/ for the configuration and see what command is being executed.

NOTE: This level requires you to create your own first shell-script. This is a very big step and you should be proud of yourself when you beat this level!

NOTE 2: Keep in mind that your shell script is removed once executed, so you may want to keep a copy around…

b) Solution

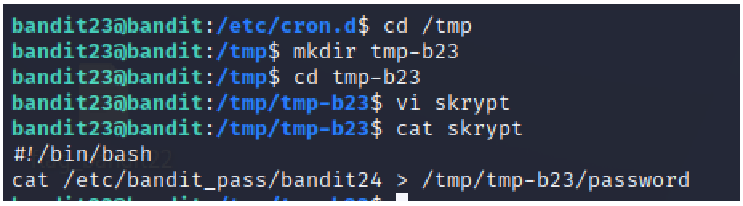
Znajduję skrypt, który będzie się wykonywać cyklicznie:



```
bandit23@bandit:~$ cd /etc/cron.d
bandit23@bandit:/etc/cron.d$ ls
cronjob_bandit15_root  cronjob_bandit22  cronjob_bandit24       e2scrub_all  sysstat
cronjob_bandit17_root  cronjob_bandit23  cronjob_bandit25_root  otw-tmp-dir
bandit23@bandit:/etc/cron.d$ cat cronjob_bandit24
@reboot bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
* * * * * bandit24 /usr/bin/cronjob_bandit24.sh &> /dev/null
bandit23@bandit:/etc/cron.d$ cat /usr/bin/cronjob_bandit24.sh
#!/bin/bash

myname=$(whoami)

cd /var/spool/$myname/foo || exit 1
echo "Executing and deleting all scripts in /var/spool/$myname/foo:"
for i in * .*;
do
    if [ "$i" ≠ "." -a "$i" ≠ ".." ];
    then
        echo "Handling $i"
        owner="$(stat --format "%U" ./$i)"
        if [ "${owner}" = "bandit23" ]; then
            timeout -s 9 60 ./$i
        fi
        rm -rf ./$i
    fi
done
```

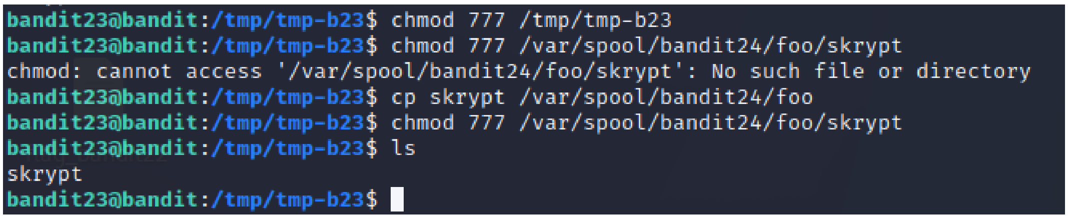
Skrypt ten wykonuje, a następnie usuwa programy znajdujące się w katalogu /car/spool/bandit24/foo

Tworzę swój własny skrypt, który następnie zostanie podstawiony do wykonania:



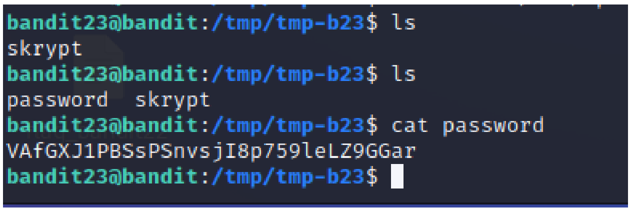
```
bandit23@bandit:/etc/cron.d$ cd /tmp
bandit23@bandit:/tmp$ mkdir tmp-b23
bandit23@bandit:/tmp$ cd tmp-b23
bandit23@bandit:/tmp/tmp-b23$ vi skrypt
bandit23@bandit:/tmp/tmp-b23$ cat skrypt
#!/bin/bash
cat /etc/bandit_pass/bandit24 > /tmp/tmp-b23/password
```

Nadaję odpowiednie uprawnienia do skryptu i katalogu w tmp oraz kopiuję plik.



```
bandit23@bandit:/tmp/tmp-b23$ chmod 777 /tmp/tmp-b23
bandit23@bandit:/tmp/tmp-b23$ chmod 777 /var/spool/bandit24/foo/skrypt
chmod: cannot access '/var/spool/bandit24/foo/skrypt': No such file or directory
bandit23@bandit:/tmp/tmp-b23$ cp skrypt /var/spool/bandit24/foo
bandit23@bandit:/tmp/tmp-b23$ chmod 777 /var/spool/bandit24/foo/skrypt
bandit23@bandit:/tmp/tmp-b23$ ls
skrypt
bandit23@bandit:/tmp/tmp-b23$
```

Czekam na wykonanie się podstawionego skryptu zgodnie z czynnościami zaplanowanymi w cron.d



```
bandit23@bandit:/tmp/tmp-b23$ ls
skrypt
bandit23@bandit:/tmp/tmp-b23$ ls
password  skrypt
bandit23@bandit:/tmp/tmp-b23$ cat password
VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar
bandit23@bandit:/tmp/tmp-b23$ 
```

c) Flag

VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar

26. Level 24-> Level 25

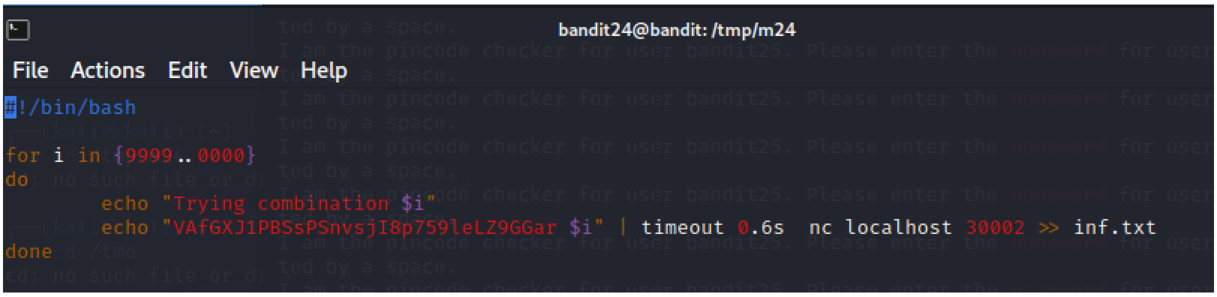
a) Level Goal

A daemon is listening on port 30002 and will give you the password for bandit25 if given the password for bandit24 and a secret numeric 4-digit pincode. There is no way to retrieve the pincode except by going through all of the 10000 combinations, called brute-forcing.

You do not need to create new connections each time.

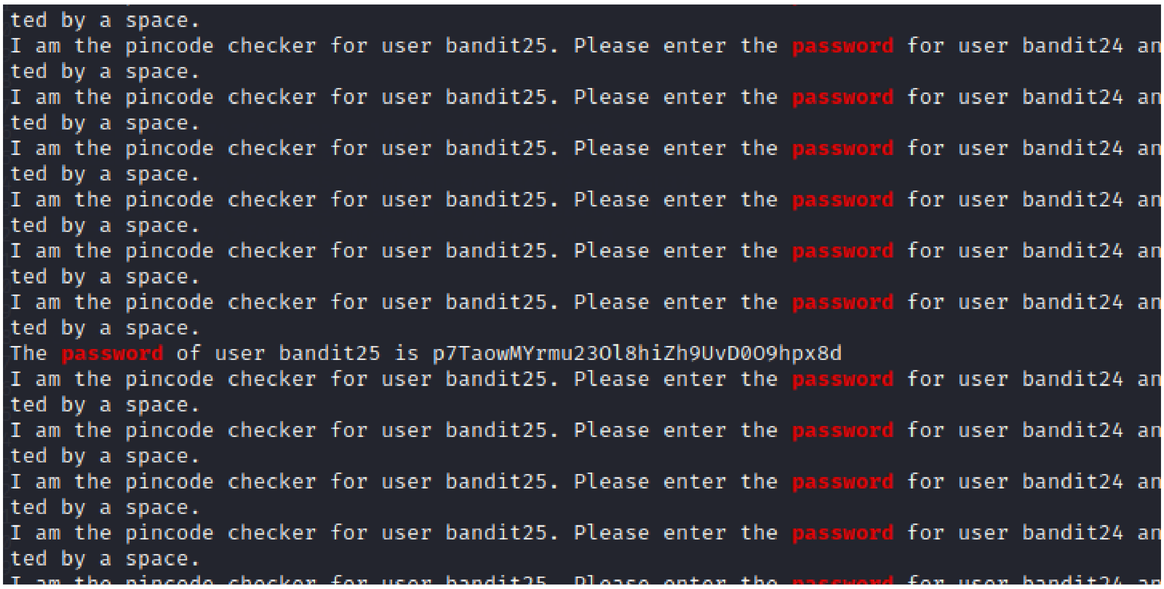
b) Solution

Tworzę skrypt do łączenia się poprzeć netcat i podejmowania próby brute-force:



```
#!/bin/bash

for i in {9999..0000}
do
        echo "Trying combination $i"
        echo "VAfGXJ1PBSsPSnvsjI8p759leLZ9GGar $i" | timeout 0.6s  nc localhost 30002 >> inf.txt
done
```

Wykonuję skrypt, następnie przeszukuję plik za pomocą grep:



```
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
The password of user bandit25 is p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
ted by a space.
I am the pincode checker for user bandit25. Please enter the password for user bandit24 an
```

c) Flag

p7TaowMYrmu23Ol8hiZh9UvD0O9hpx8d

27. Level 25 -> Level 26*

a) Level Goal

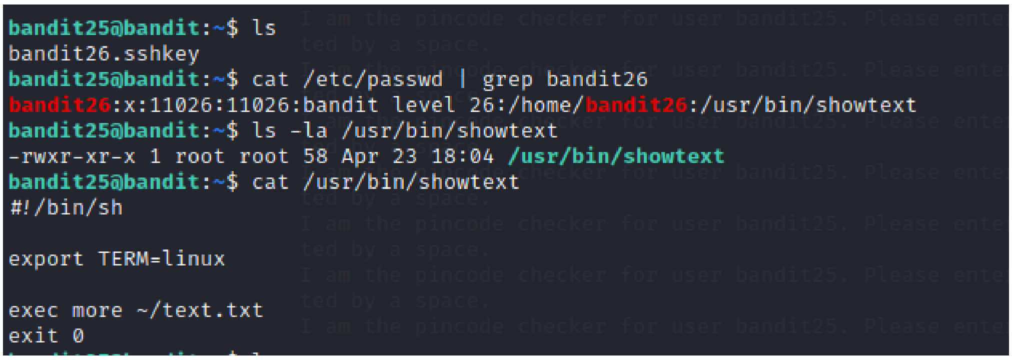
Logging in to bandit26 from bandit25 should be fairly easy… The shell for user bandit26 is not /bin/bash, but something else. Find out what it is, how it works and how to break out of it.

b) Solution

Każdy użytkownik ma swój shell, który się uruchamia podczas logowania się poprzez ssh. Informacja o domyślnym shellu dla użytkownika znajduje się na końcu linii w pliku /etc/passwd.

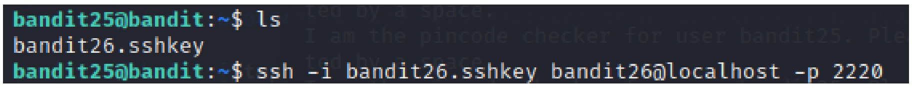
More to komenda umożliwiająca wyświetlanie plików w interaktywnym trybie. Tryb ten włącza się, gdy plik jest zbyt duży, aby wyświetlić go w całości. Komenda v w trybie interaktywnym umożliwia edycję pliku za pomocą VIM.

Najpierw sprawdzam wykorzystywany przez użytkownika bandit26 shell:



```
bandit25@bandit:~$ ls
bandit26.sshkey
bandit25@bandit:~$ cat /etc/passwd | grep bandit26
bandit26:x:11026:11026:bandit level 26:/home/bandit26:/usr/bin/showtext
bandit25@bandit:~$ ls -la /usr/bin/showtext
-rwxr-xr-x 1 root root 58 Apr 23 18:04 /usr/bin/showtext
bandit25@bandit:~$ cat /usr/bin/showtext
#!/bin/sh

export TERM=linux

exec more ~/text.txt
exit 0
```

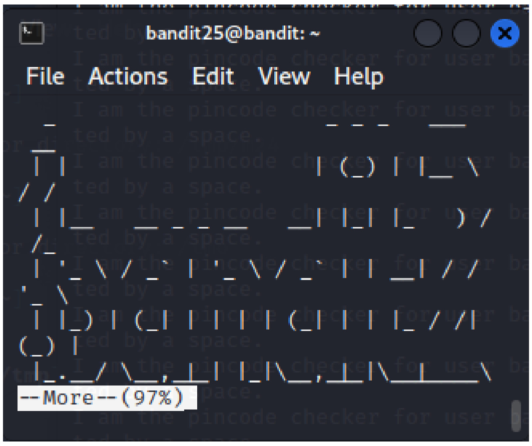
Próbując zalogować się z wykorzystaniem znalezionego klucza ssh zostajemy od razu wylogowani:

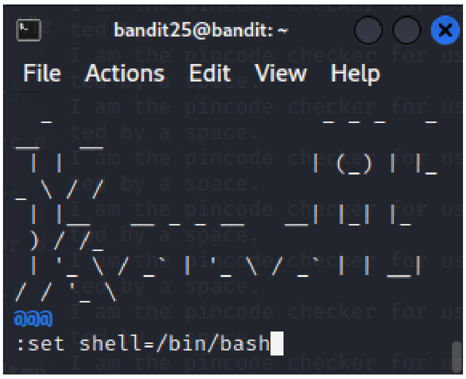


```
bandit25@bandit:~$ ls
bandit26.sshkey
bandit25@bandit:~$ ssh -i bandit26.sshkey bandit26@localhost -p 2220
```

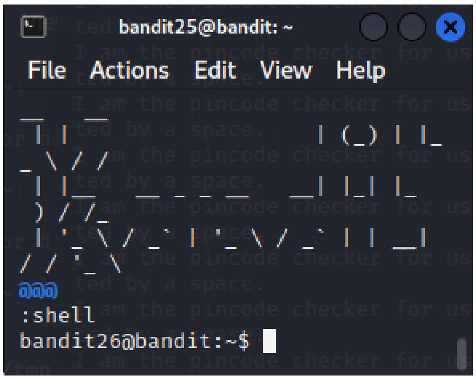
W tym przypadku komenda more nie uruchomiła trybu interaktywnego, ponieważ okno było wystarczająco duże, aby wyświetlić cały tekst, należy zmniejszyć okno.

Po zmniejszeniu okna udaje się uzyskać tryb interaktywny:

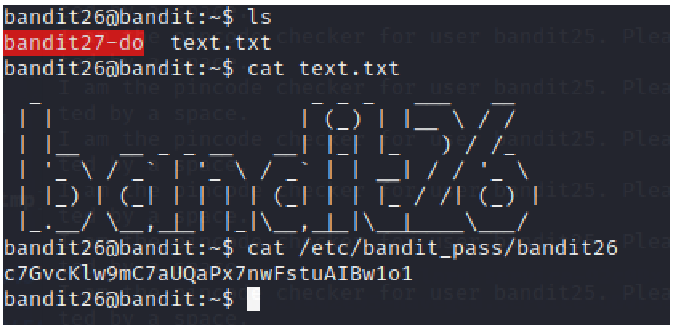




```
  _                     _ _ _   _
__   __
 | |                   | (_) | |_
_ \ / /
 | |__   __ _ _ __   __| |_| |_
 ) / /_
 | '_ \ / _` | '_ \ / _` | | __|
/ / '_ \
@@@
:set shell=/bin/bash
```



```
__   __
 | |                   | (_) | |_
_ \ / /
 | |__   __ _ _ __   __| |_| |_
 ) / /_
 | '_ \ / _` | '_ \ / _` | | __|
/ / '_ \
@@@
:shell
bandit26@bandit:~$
```



```
bandit26@bandit:~$ ls
bandit27-do  text.txt
bandit26@bandit:~$ cat text.txt
  _                     _ _ _   ___   __
 | |                   | (_) | |__ \ / /
 | |__   __ _ _ __   __| |_| |_   ) / /_
 | '_ \ / _` | '_ \ / _` | | __| / / '_ \
 | |_) | (_| | | | | (_| | | |_ / /| (_) |
 |_.__/ \__,_|_| |_|\__,_|_|\__|____\___/
bandit26@bandit:~$ cat /etc/bandit_pass/bandit26
c7GvcKlw9mC7aUQaPx7nwFstuAIBw1o1
bandit26@bandit:~$
```

c) Flag

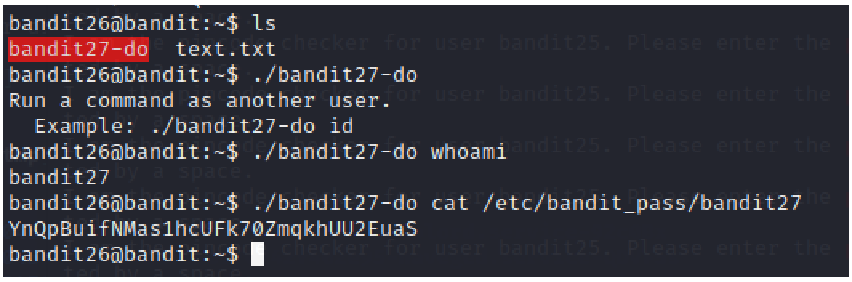
c7GvcKlw9mC7aUQaPx7nwFstuAIBw1o1

28. Level 26 -> Level 27

a) Level Goal

Good job getting a shell! Now hurry and grab the password for bandit27!

b) Solution



```
bandit26@bandit:~$ ls
bandit27-do  text.txt
bandit26@bandit:~$ ./bandit27-do
Run a command as another user.
  Example: ./bandit27-do id
bandit26@bandit:~$ ./bandit27-do whoami
bandit27
bandit26@bandit:~$ ./bandit27-do cat /etc/bandit_pass/bandit27
YnQpBuifNMas1hcUFk70ZmqkhUU2EuaS
bandit26@bandit:~$
```

c) Flag

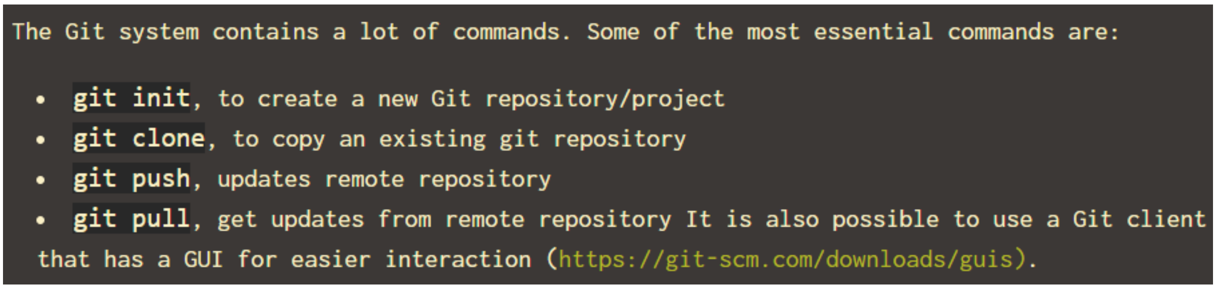
YnQpBuifNMas1hcUFk70ZmqkhUU2EuaS

29. Level 27 -> Level 28

a) Level Goal

There is a git repository at ssh://bandit27-git@localhost/home/bandit27-git/repo via the port 2220. The password for the user bandit27-git is the same as for the user bandit27.

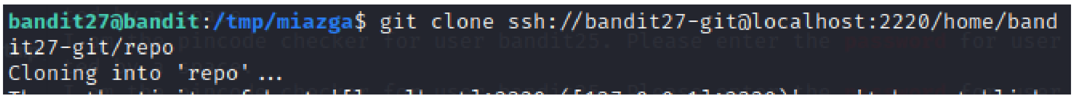
b) Solution



The Git system contains a lot of commands. Some of the most essential commands are:

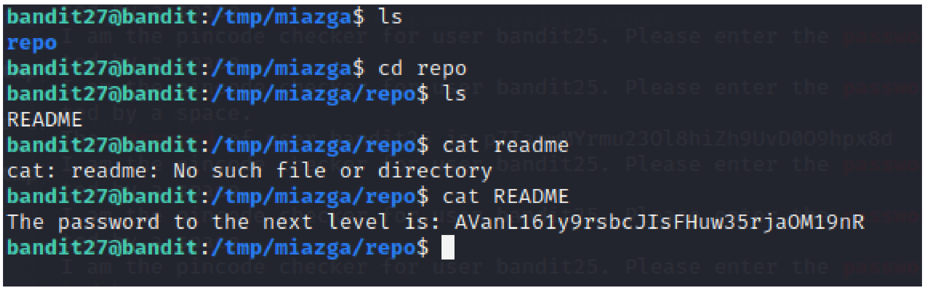
- `git init`, to create a new Git repository/project
- `git clone`, to copy an existing git repository
- `git push`, updates remote repository
- `git pull`, get updates from remote repository It is also possible to use a Git client that has a GUI for easier interaction (https://git-scm.com/downloads/guis).

Klonuję repo za pomocą polecenia:



```
bandit27@bandit:/tmp/miazga$ git clone ssh://bandit27-git@localhost:2220/home/band
it27-git/repo
Cloning into 'repo'...
```

Znajduję hasło w REAMDE



```
bandit27@bandit:/tmp/miazga$ ls
repo
bandit27@bandit:/tmp/miazga$ cd repo
bandit27@bandit:/tmp/miazga/repo$ ls
README
bandit27@bandit:/tmp/miazga/repo$ cat readme
cat: readme: No such file or directory
bandit27@bandit:/tmp/miazga/repo$ cat README
The password to the next level is: AVanL161y9rsbcJIsFHuw35rjaOM19nR
bandit27@bandit:/tmp/miazga/repo$ 
```

c) Flag

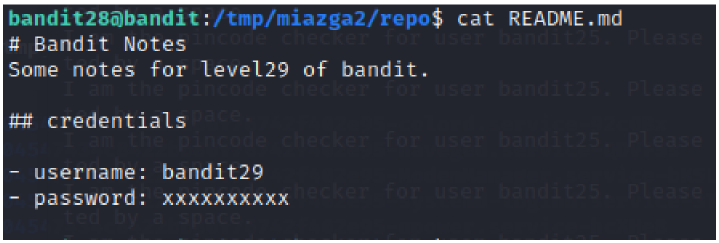
AVanL161y9rsbcJIsFHuw35rjaOM19nR

30. Level 28 -> Level 29

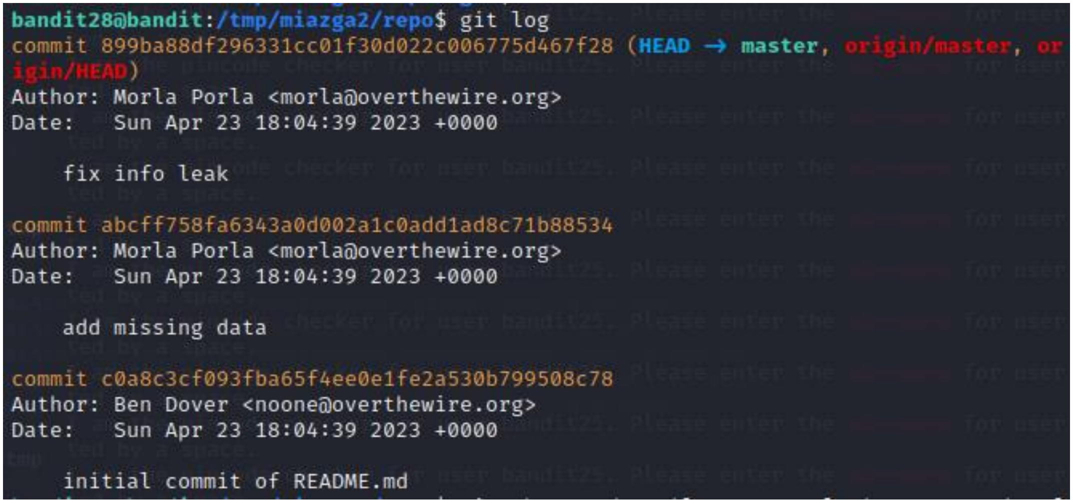
a) Level Goal

There is a git repository at ssh://bandit28-git@localhost/home/bandit28-git/repo via the port 2220. The password for the user bandit28-git is the same as for the user bandit28.

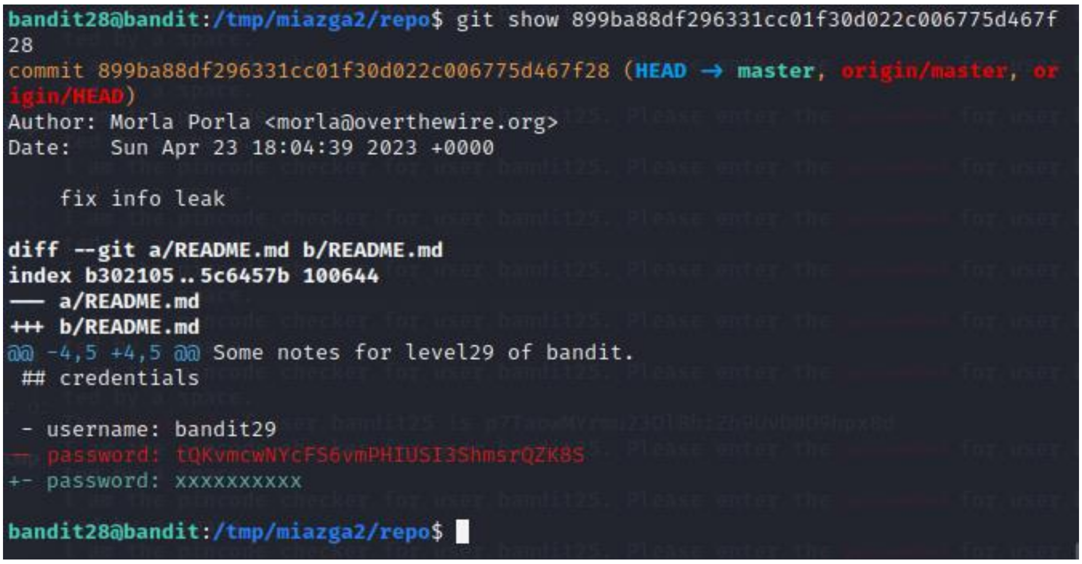
b) Solution



```
bandit28@bandit:/tmp/miazga2/repo$ cat README.md
# Bandit Notes
Some notes for level29 of bandit.

## credentials

- username: bandit29
- password: xxxxxxxxxx
```



```
bandit28@bandit:/tmp/miazga2/repo$ git log
commit 899ba88df296331cc01f30d022c006775d467f28 (HEAD → master, origin/master, or
igin/HEAD)
Author: Morla Porla <morla@overthewire.org>
Date:   Sun Apr 23 18:04:39 2023 +0000

    fix info leak

commit abcff758fa6343a0d002a1c0add1ad8c71b88534
Author: Morla Porla <morla@overthewire.org>
Date:   Sun Apr 23 18:04:39 2023 +0000

    add missing data

commit c0a8c3cf093fba65f4ee0e1fe2a530b799508c78
Author: Ben Dover <noone@overthewire.org>
Date:   Sun Apr 23 18:04:39 2023 +0000

    initial commit of README.md
```



```
bandit28@bandit:/tmp/miazga2/repo$ git show 899ba88df296331cc01f30d022c006775d467f
28
commit 899ba88df296331cc01f30d022c006775d467f28 (HEAD → master, origin/master, or
igin/HEAD)
Author: Morla Porla <morla@overthewire.org>
Date:   Sun Apr 23 18:04:39 2023 +0000

    fix info leak

diff --git a/README.md b/README.md
index b302105..5c6457b 100644
--- a/README.md
+++ b/README.md
@@ -4,5 +4,5 @@ Some notes for level29 of bandit.
 ## credentials

 - username: bandit29
-- password: tQKvmcwNYcFS6vmPHIUSI3ShmsrQZK8S
+- password: xxxxxxxxxx

bandit28@bandit:/tmp/miazga2/repo$ █
```

c) Flag

tQKvmcwNYcFS6vmPHIUSI3ShmsrQZK8S

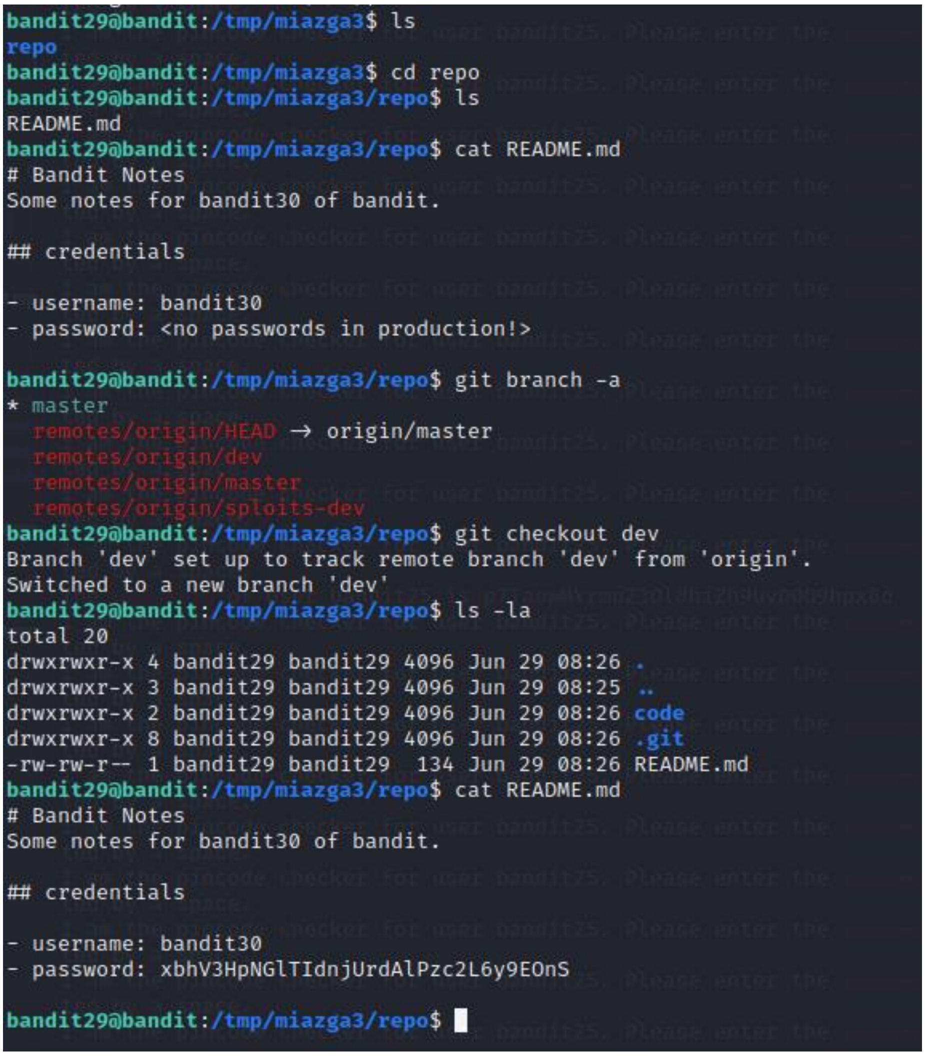
31. Level 29 -> Level 30

a) Level Goal

There is a git repository at ssh://bandit29-git@localhost/home/bandit29-git/repo via the port 2220. The password for the user bandit29-git is the same as for the user bandit29.

Clone the repository and find the password for the next level.

b) Solution



```
bandit29@bandit:/tmp/miazga3$ ls
repo
bandit29@bandit:/tmp/miazga3$ cd repo
bandit29@bandit:/tmp/miazga3/repo$ ls
README.md
bandit29@bandit:/tmp/miazga3/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: <no passwords in production!>

bandit29@bandit:/tmp/miazga3/repo$ git branch -a
* master
  remotes/origin/HEAD → origin/master
  remotes/origin/dev
  remotes/origin/master
  remotes/origin/sploits-dev
bandit29@bandit:/tmp/miazga3/repo$ git checkout dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
Switched to a new branch 'dev'
bandit29@bandit:/tmp/miazga3/repo$ ls -la
total 20
drwxrwxr-x 4 bandit29 bandit29 4096 Jun 29 08:26 .
drwxrwxr-x 3 bandit29 bandit29 4096 Jun 29 08:25 ..
drwxrwxr-x 2 bandit29 bandit29 4096 Jun 29 08:26 code
drwxrwxr-x 8 bandit29 bandit29 4096 Jun 29 08:26 .git
-rw-rw-r-- 1 bandit29 bandit29  134 Jun 29 08:26 README.md
bandit29@bandit:/tmp/miazga3/repo$ cat README.md
# Bandit Notes
Some notes for bandit30 of bandit.

## credentials

- username: bandit30
- password: xbhV3HpNGlTIdnjUrdAlPzc2L6y9EOnS

bandit29@bandit:/tmp/miazga3/repo$ █
```

c) Flag

xbhV3HpNGlTIdnjUrdAlPzc2L6y9EOnS

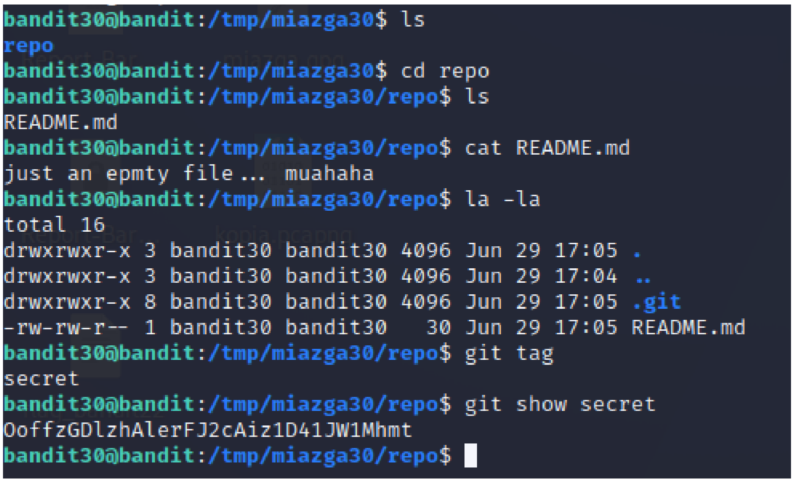
32. Level 30 -> Level 31

a) Level Goal

There is a git repository at ssh://bandit30-git@localhost/home/bandit30-git/repo via the port 2220. The password for the user bandit30-git is the same as for the user bandit30.

Clone the repository and find the password for the next level.

b) Solution



```
bandit30@bandit:/tmp/miazga30$ ls
repo
bandit30@bandit:/tmp/miazga30$ cd repo
bandit30@bandit:/tmp/miazga30/repo$ ls
README.md
bandit30@bandit:/tmp/miazga30/repo$ cat README.md
just an epmty file... muahaha
bandit30@bandit:/tmp/miazga30/repo$ la -la
total 16
drwxrwxr-x 3 bandit30 bandit30 4096 Jun 29 17:05 .
drwxrwxr-x 3 bandit30 bandit30 4096 Jun 29 17:04 ..
drwxrwxr-x 8 bandit30 bandit30 4096 Jun 29 17:05 .git
-rw-rw-r-- 1 bandit30 bandit30   30 Jun 29 17:05 README.md
bandit30@bandit:/tmp/miazga30/repo$ git tag
secret
bandit30@bandit:/tmp/miazga30/repo$ git show secret
OoffzGDlzhAlerFJ2cAiz1D41JW1Mhmt
bandit30@bandit:/tmp/miazga30/repo$ 
```

c) Flag

OoffzGDlzhAlerFJ2cAiz1D41JW1Mhmt

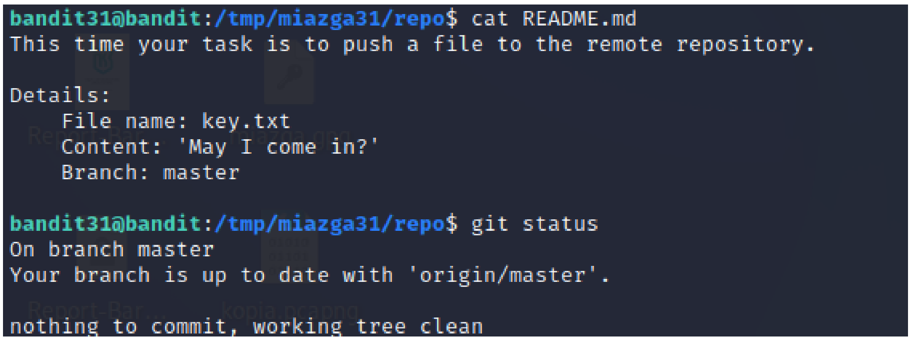
33. Level 31 -> Level 32

a) Level Goal

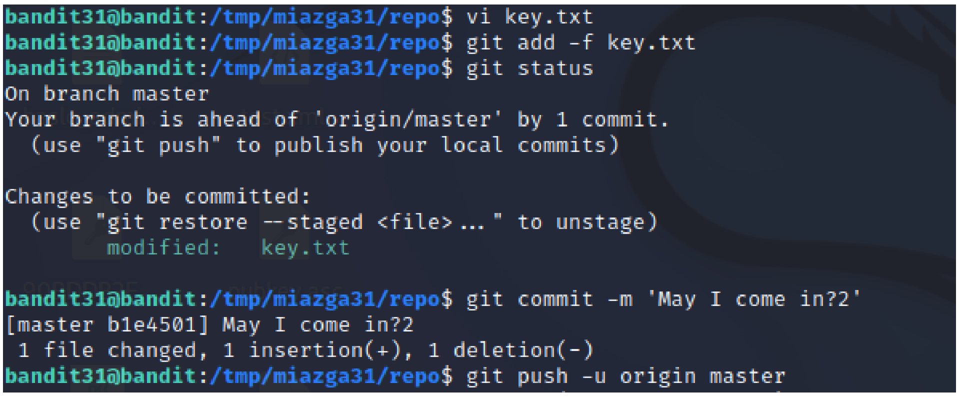
There is a git repository at ssh://bandit31-git@localhost/home/bandit31-git/repo via the port 2220. The password for the user bandit31-git is the same as for the user bandit31.

Clone the repository and find the password for the next level.

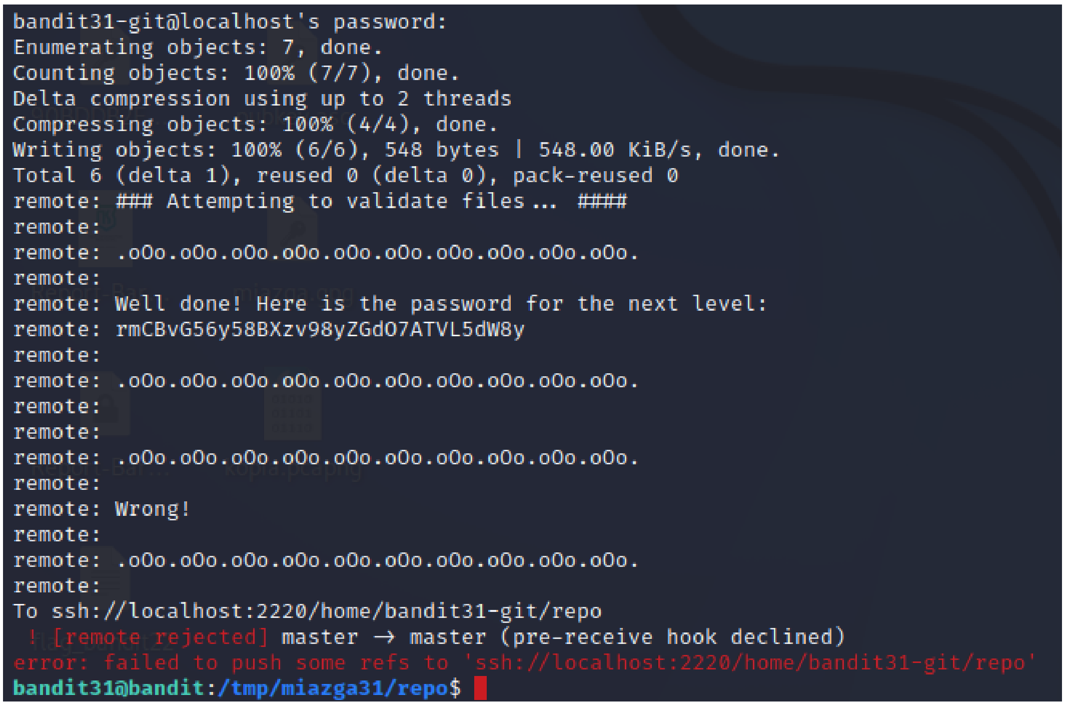
b) Solution



```
bandit31@bandit:/tmp/miazga31/repo$ cat README.md
This time your task is to push a file to the remote repository.

Details:
    File name: key.txt
    Content: 'May I come in?'
    Branch: master

bandit31@bandit:/tmp/miazga31/repo$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```



```
bandit31@bandit:/tmp/miazga31/repo$ vi key.txt
bandit31@bandit:/tmp/miazga31/repo$ git add -f key.txt
bandit31@bandit:/tmp/miazga31/repo$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   key.txt

bandit31@bandit:/tmp/miazga31/repo$ git commit -m 'May I come in?2'
[master b1e4501] May I come in?2
 1 file changed, 1 insertion(+), 1 deletion(-)
bandit31@bandit:/tmp/miazga31/repo$ git push -u origin master
```



```
bandit31-git@localhost's password:
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 548 bytes | 548.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: ### Attempting to validate files ... ####
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
remote: Well done! Here is the password for the next level:
remote: rmCBvG56y58BXzv98yZGdO7ATVL5dW8y
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
remote: Wrong!
remote:
remote: .oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.oOo.
remote:
To ssh://localhost:2220/home/bandit31-git/repo
 ! [remote rejected] master → master (pre-receive hook declined)
error: failed to push some refs to 'ssh://localhost:2220/home/bandit31-git/repo'
bandit31@bandit:/tmp/miazga31/repo$
```

c) Flag

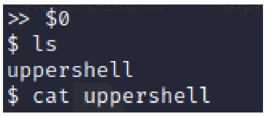
rmCBvG56y58BXzv98yZGdO7ATVL5dW8y

34. Level 32 -> Level 33

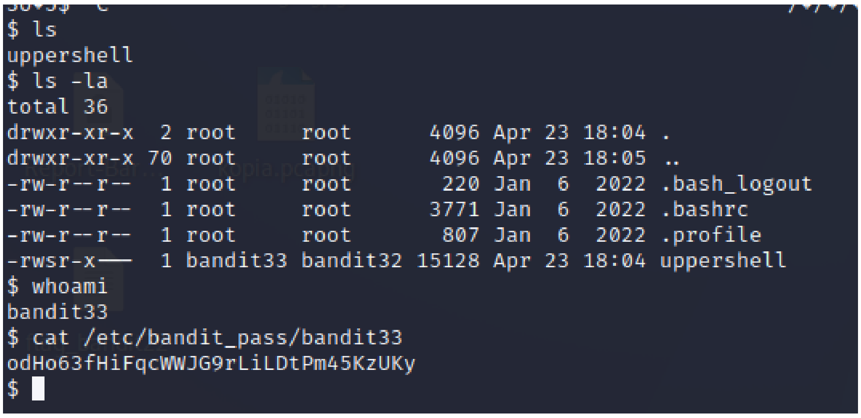
a) Level Goal

After all this git stuff its time for another escape. Good luck!

b) Solution



```
>> $0
$ ls
uppershell
$ cat uppershell
```



```
$ ls
uppershell
$ ls -la
total 36
drwxr-xr-x  2 root     root      4096 Apr 23 18:04 .
drwxr-xr-x 70 root     root      4096 Apr 23 18:05 ..
-rw-r--r--  1 root     root       220 Jan  6  2022 .bash_logout
-rw-r--r--  1 root     root      3771 Jan  6  2022 .bashrc
-rw-r--r--  1 root     root       807 Jan  6  2022 .profile
-rwsr-x---  1 bandit33 bandit32 15128 Apr 23 18:04 uppershell
$ whoami
bandit33
$ cat /etc/bandit_pass/bandit33
odHo63fHiFqcWWJG9rLiLDtPm45KzUKy
$
```

c) Flag

odHo63fHiFqcWWJG9rLiLDtPm45KzUKy