Sri Lanka Institute of Information Technology

IE2012 - Systems and Network Programming

Year 2, Semester 1

Overthewire Bandit

Reg No : IT22276414

Name : Diwanka.K.A.W

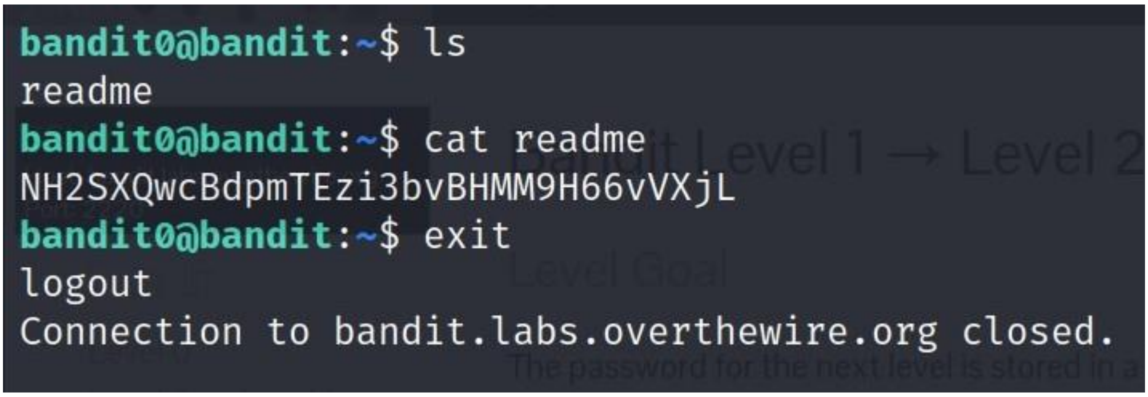
LEVEL 0

Log into using: ssh bandit0@bandit.labs.overthewire.org -p 2220

They have given the password for this one as bandit0

LEVEL 0 -1

I logged in as the bandit0 user profile and used the "ls" command to see if I could find any interesting files. A file with the name "readme" is present, according to the report.
Using "**cat readme**" to access the file, I discovered a single line that read.

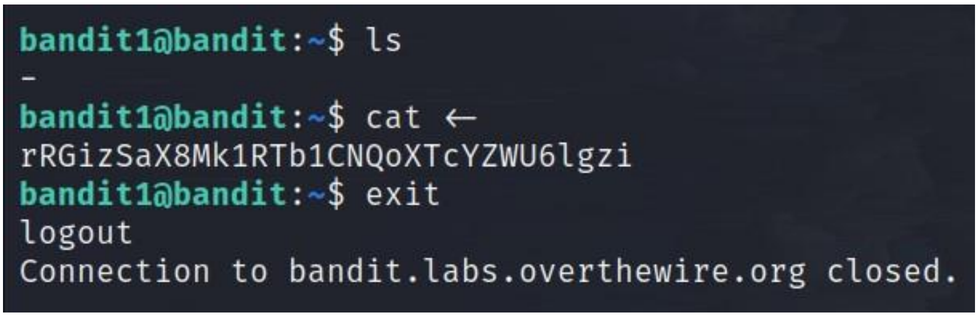


```
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL
bandit0@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - NH2SXQwcBdpmTEzi3bvBHMM9H66vVXjL

LEVEL 1-2

The password is stored in a file called **-** according to the Bandit website. After running "ls" and logging into the account, I found the file. I then discovered what the password was. Files that start with a dash can't be read unless you use the operator to reroute them to stdin.

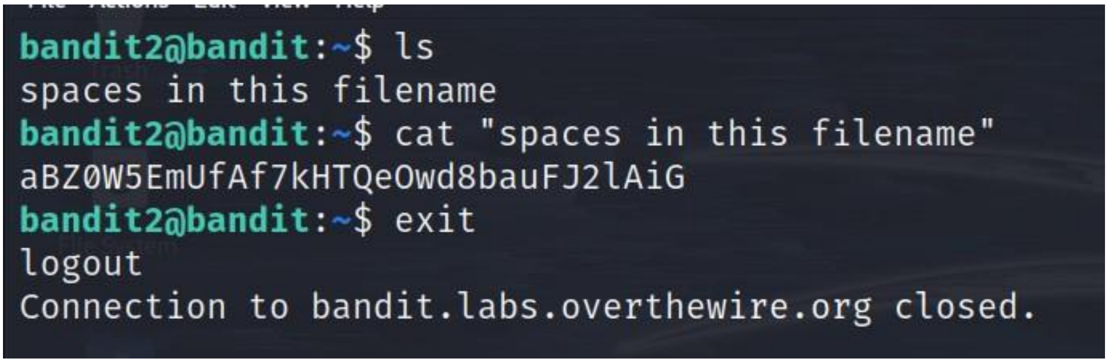


```
bandit1@bandit:~$ ls
-
bandit1@bandit:~$ cat ←
rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi
bandit1@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - rRGizSaX8Mk1RTb1CNQoXTcYZWU6lgzi

LEVEL 2-3

A file named spaces in this filename contains the password for the subsequent user. Running **cat "spaces in this filename"** allowed me to get the password. To read the file name, if it contains spaces, you must enclose it in quotation marks.

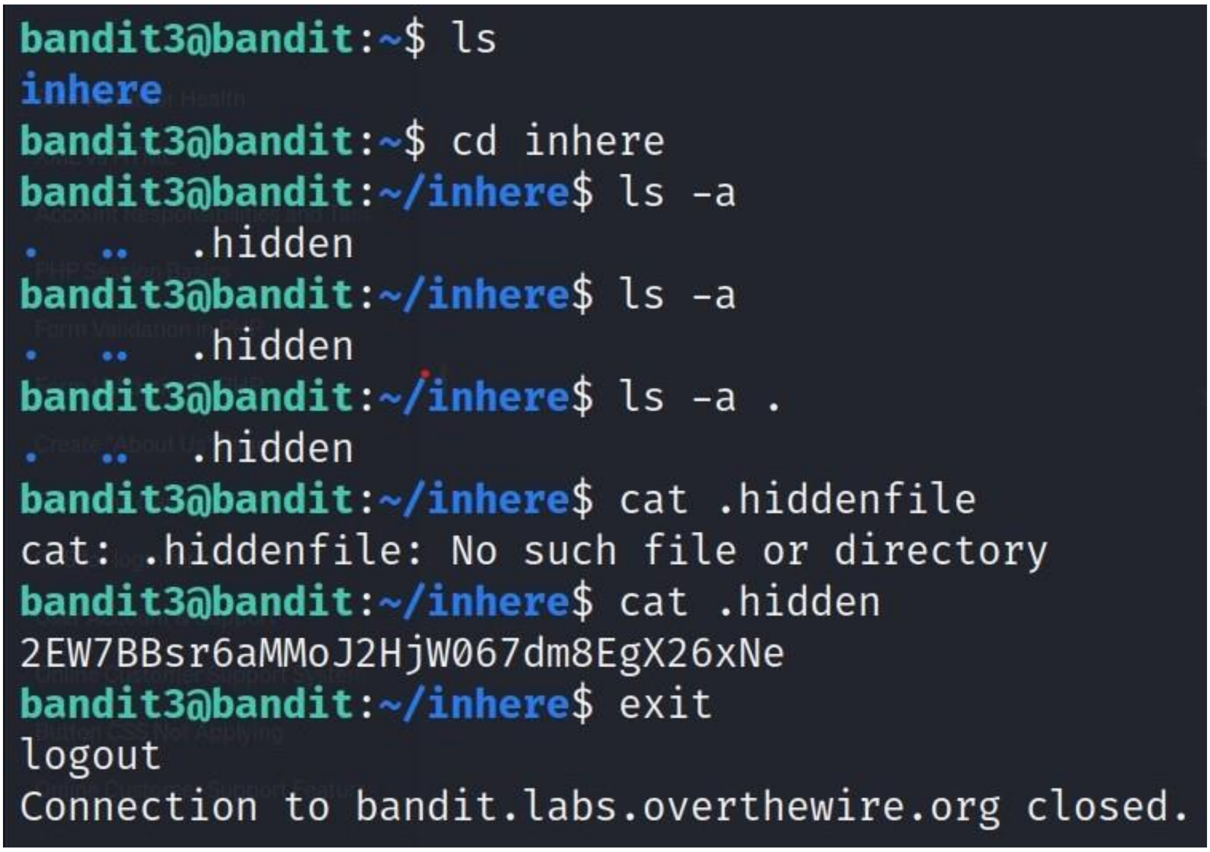


```
bandit2@bandit:~$ ls
spaces in this filename
bandit2@bandit:~$ cat "spaces in this filename"
aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG
bandit2@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - aBZ0W5EmUfAf7kHTQeOwd8bauFJ2lAiG

LEVEL 3-4

The password is contained in a hidden file located in the "inhere" directory. I used "cd" to access the inhere folder after using "ls" to list the files and directories. After that, I ran "ls -a" to see every file, hidden and otherwise. Linux assigns a dot to the names of all hidden files and directories. I discovered the next password by running **"cat .hidden"** on the hidden file. .hidden is the name of the secret file.

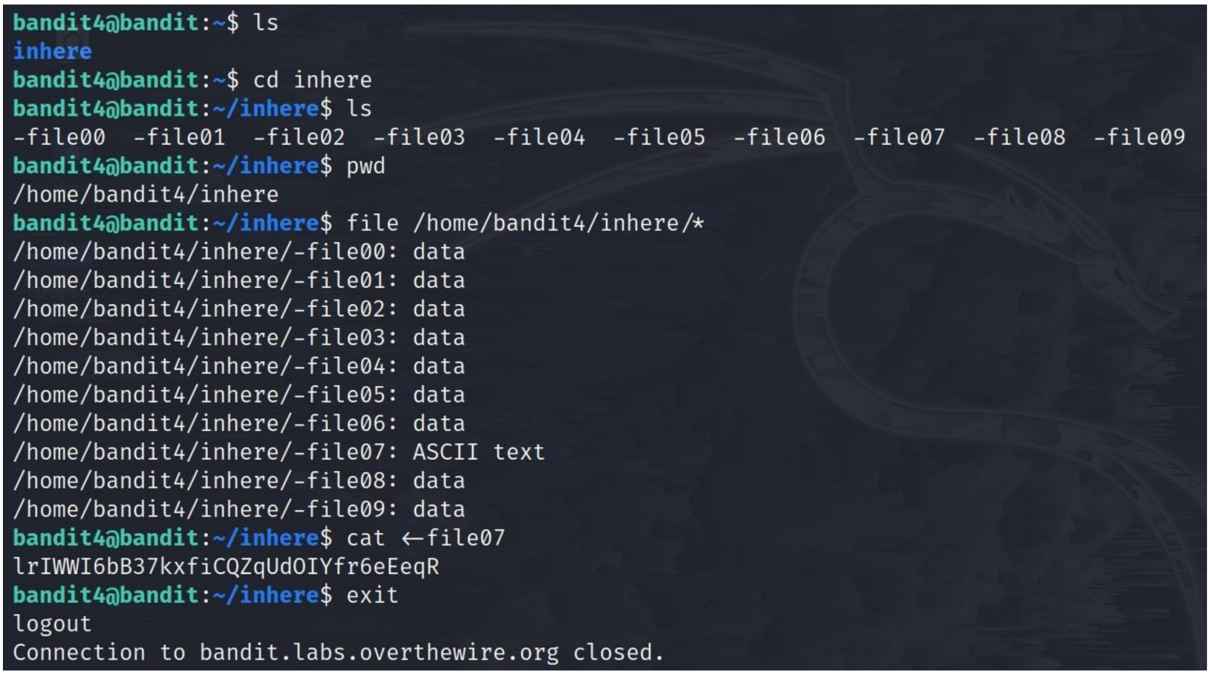


```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere
bandit3@bandit:~/inhere$ ls -a
.   ..   .hidden
bandit3@bandit:~/inhere$ ls -a
.   ..   .hidden
bandit3@bandit:~/inhere$ ls -a .
.   ..   .hidden
bandit3@bandit:~/inhere$ cat .hiddenfile
cat: .hiddenfile: No such file or directory
bandit3@bandit:~/inhere$ cat .hidden
2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe
bandit3@bandit:~/inhere$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - 2EW7BBsr6aMMoJ2HjW067dm8EgX26xNe

LEVEL 4-5

In our task to find the password for the next level, we navigate to the "inhere" directory and examine its contents using the "ls" command. Among the files listed, we observe that only one, namely "-file07", is identified as **ASCII text**. This indicates that it is potentially human-readable. To confirm this assumption and retrieve the password, we utilize the "file" command to examine the type of file "-file07" is. The output confirms our suspicion, indicating that it is indeed human-readable. To extract the password, we proceed to concatenate the contents of "-file07" using the "cat" command, following the syntax **"cat < -file07"**. This action allows us to access and retrieve the password necessary to proceed to the next level.



```
bandit4@bandit:~$ ls
inhere
bandit4@bandit:~$ cd inhere
bandit4@bandit:~/inhere$ ls
-file00  -file01  -file02  -file03  -file04  -file05  -file06  -file07  -file08  -file09
bandit4@bandit:~/inhere$ pwd
/home/bandit4/inhere
bandit4@bandit:~/inhere$ file /home/bandit4/inhere/*
/home/bandit4/inhere/-file00: data
/home/bandit4/inhere/-file01: data
/home/bandit4/inhere/-file02: data
/home/bandit4/inhere/-file03: data
/home/bandit4/inhere/-file04: data
/home/bandit4/inhere/-file05: data
/home/bandit4/inhere/-file06: data
/home/bandit4/inhere/-file07: ASCII text
/home/bandit4/inhere/-file08: data
/home/bandit4/inhere/-file09: data
bandit4@bandit:~/inhere$ cat ←file07
lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR
bandit4@bandit:~/inhere$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - lrIWWI6bB37kxfiCQZqUdOIYfr6eEeqR

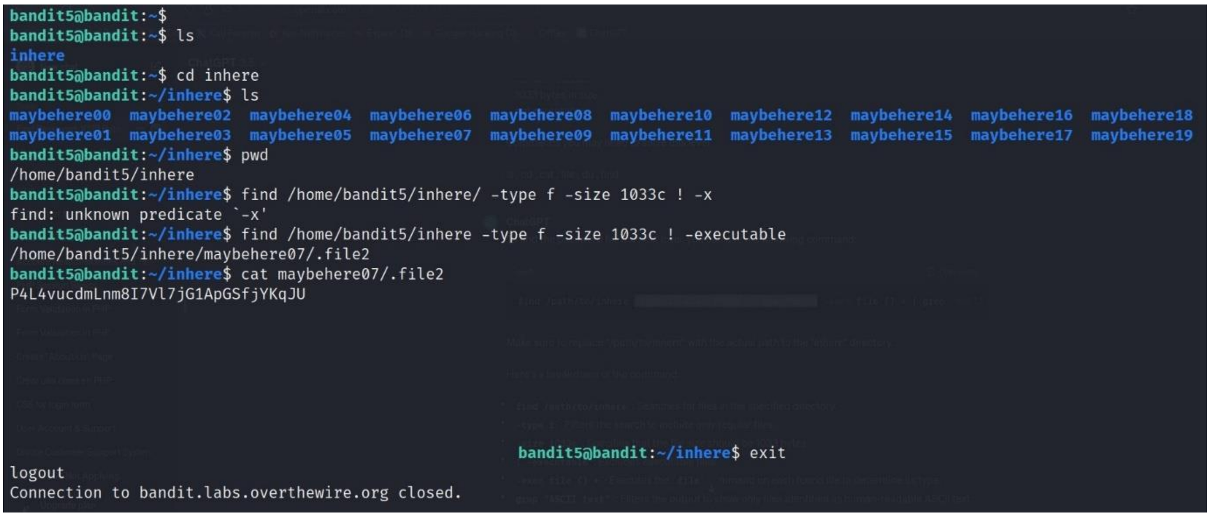
LEVEL 5-6

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

First, check the contents of the "inhere" directory using "ls," finding files named "maybehere" followed by numbers. Then switch to this directory using "cd," confirmed by "pwd" showing "/home/bandit5/inhere" The goal is to find a file of exactly 1033 bytes that is not executable. Successfully locate a file called ".file2" in the "maybehere07" directory. Using "cat," inspect ".file2" and discover a password for the next level.

find /home/bandit5/inhere -type f -size 1033c ! -executable

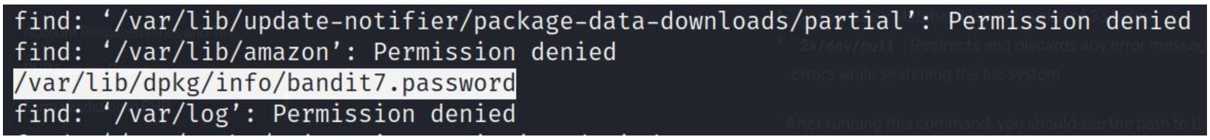


```
bandit5@bandit:~$
bandit5@bandit:~$ ls
inhere
bandit5@bandit:~$ cd inhere
bandit5@bandit:~/inhere$ ls
maybehere00  maybehere02  maybehere04  maybehere06  maybehere08  maybehere10  maybehere12  maybehere14  maybehere16  maybehere18
maybehere01  maybehere03  maybehere05  maybehere07  maybehere09  maybehere11  maybehere13  maybehere15  maybehere17  maybehere19
bandit5@bandit:~/inhere$ pwd
/home/bandit5/inhere
bandit5@bandit:~/inhere$ find /home/bandit5/inhere/ -type f -size 1033c ! -x
find: unknown predicate `-x'
bandit5@bandit:~/inhere$ find /home/bandit5/inhere -type f -size 1033c ! -executable
/home/bandit5/inhere/maybehere07/.file2
bandit5@bandit:~/inhere$ cat maybehere07/.file2
P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU

                                                                           bandit5@bandit:~/inhere$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - P4L4vucdmLnm8I7Vl7jG1ApGSfjYKqJU

LEVEL 6-7

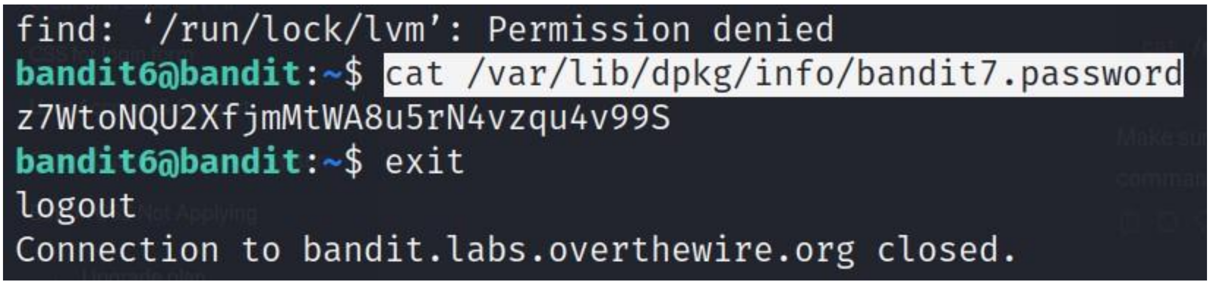
Type **“ find / user bandit7 -group bandit6 -size 33c ”** to find the hint they have given Then you willl find **“/var/lib/dpkg/info/bandit7.password”**



```
find: ‘/var/lib/update-notifier/package-data-downloads/partial’: Permission denied
find: ‘/var/lib/amazon’: Permission denied
/var/lib/dpkg/info/bandit7.password
find: ‘/var/log’: Permission denied
```

Copy **“/var/lib/dpkg/info/bandit7.password ”** and cat this one

Then you will get the password.

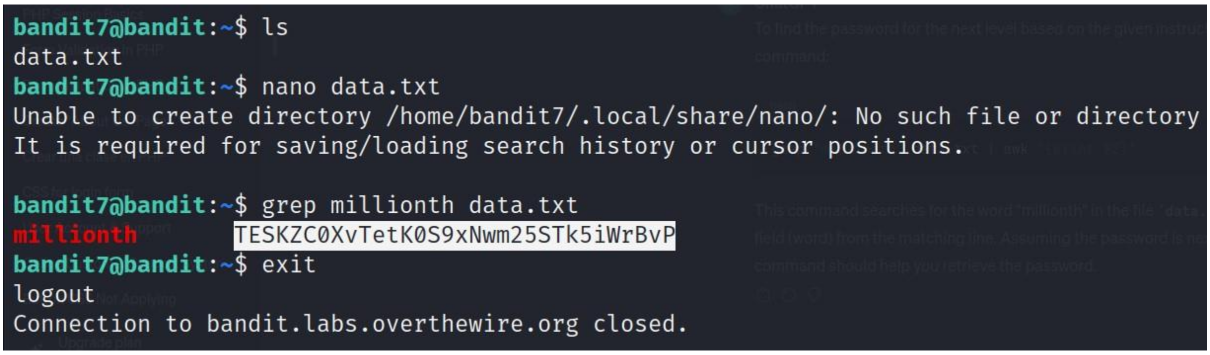


```
find: ‘/run/lock/lvm’: Permission denied
bandit6@bandit:~$ cat /var/lib/dpkg/info/bandit7.password
z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S
bandit6@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - z7WtoNQU2XfjmMtWA8u5rN4vzqu4v99S

LEVEL 7-8

Use the “ls” command to first inspect the file, after which the data.txt file will be visible. After reading it, the password will be located next to the term "millionth." Because it is exceedingly difficult to find, we utilise the special programme **"grep"** to locate the term. You will then discover the password.



```
bandit7@bandit:~$ ls
data.txt
bandit7@bandit:~$ nano data.txt
Unable to create directory /home/bandit7/.local/share/nano/: No such file or directory
It is required for saving/loading search history or cursor positions.

bandit7@bandit:~$ grep millionth data.txt
millionth       TESKZC0XvTetK0S9xNwm25STk5iWrBvP
bandit7@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

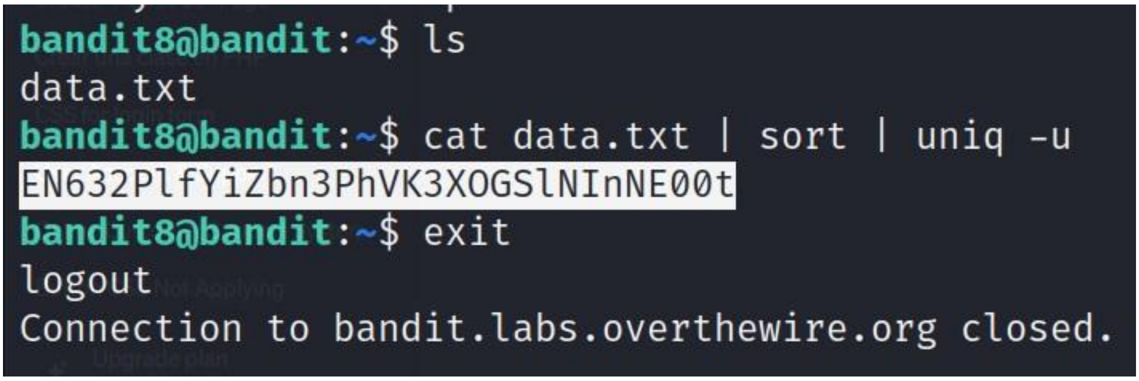
Password - TESKZC0XvTetK0S9xNwm25STk5iWrBvP

LEVEL 8-9

The file data.txt contains the password for the next level, which is the only line of text that appears only once.

I did this by using the command below to eliminate all duplicates from the output and sort the lines alphabetically.

Once you type **"cat data.txt | sort | uniq -u"** the password will appear.



```
bandit8@bandit:~$ ls
data.txt
bandit8@bandit:~$ cat data.txt | sort | uniq -u
EN632PlfYiZbn3PhVK3XOGSlNInNE00t
bandit8@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

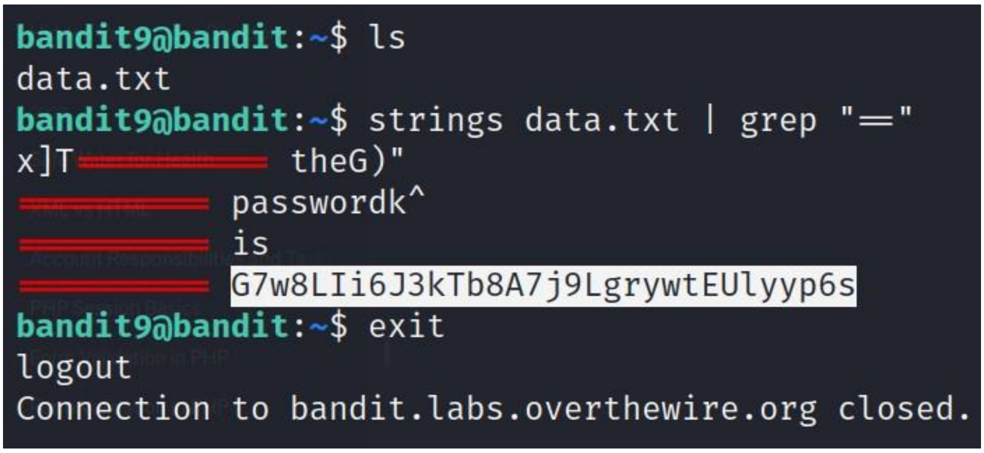
Password - EN632PlfYiZbn3PhVK3XOGSlNInNE00t

LEVEL 9-10

There was a binary file named data.txt at this level. Therefore, we must locate some kind character.

This command will look through the data.txt file and show every line that has the character "=" on it. If a line contains more than one instance of "=", those lines will also be shown.

strings data.txt | grep “=”



```
bandit9@bandit:~$ ls
data.txt
bandit9@bandit:~$ strings data.txt | grep "="
x]T[redacted] theG)"
[redacted] passwordk^
[redacted] is
[redacted] G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s
bandit9@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

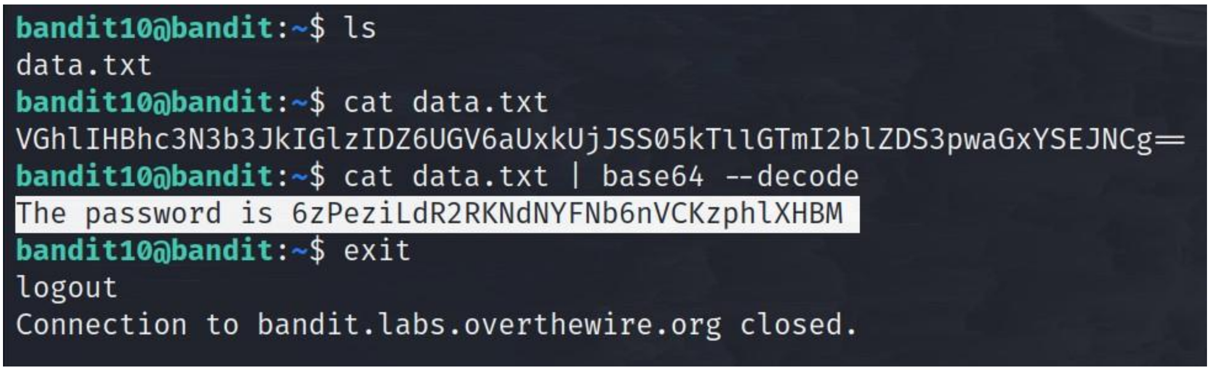
Password - G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s

LEVEL 10-11

This text has a **base64** encoding. Thus, it needs to be decoded.

To encode and decode data in base64 format, use the base64 command. To select the decoding mode, use the -d option. The base64 command can be used to decode data from Base64 format back to its original form when the -d option is applied.

cat data.txt | base64 –decode



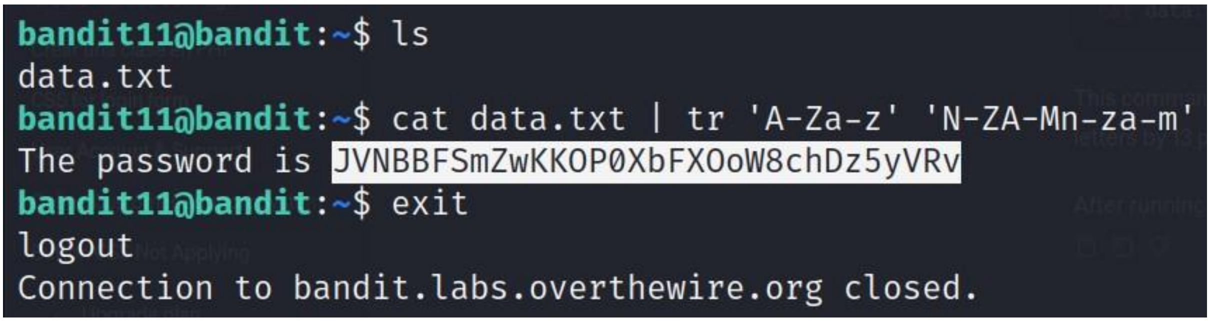
```
bandit10@bandit:~$ ls
data.txt
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b3JkIGlzIDZ6UGV6aUxkUjJSS05kTllGTmI2blZDS3pwaGxYSEJNCg==
bandit10@bandit:~$ cat data.txt | base64 --decode
The password is 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM
bandit10@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM

LEVEL 11-12

The file data.txt contains the password for the next level. All of the letters, lowercase (a-z) and uppercase (A-Z), have been rotated by 13 places.
The ROT13 method was used to encrypt one line in the data.txt file. To decrypt it, I have to replace each letter with the one that is 13 positions ahead. For example, using this encryption, the letter a would become the letter n. Encoding the word "banana" would result in "onanan." I carried out the following command to decode the string.

cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'

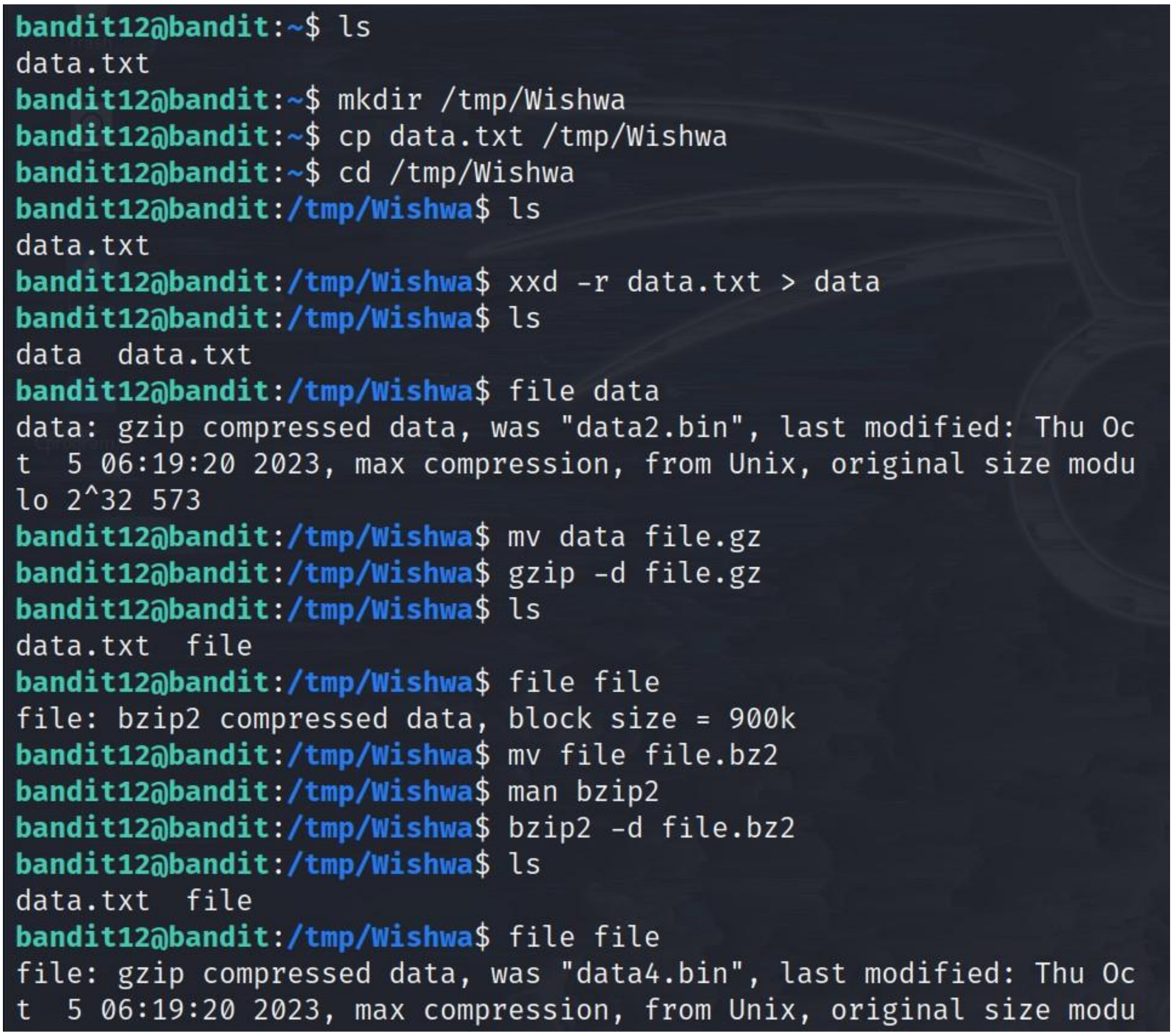


```
bandit11@bandit:~$ ls
data.txt
bandit11@bandit:~$ cat data.txt | tr 'A-Za-z' 'N-ZA-Mn-za-m'
The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv
bandit11@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

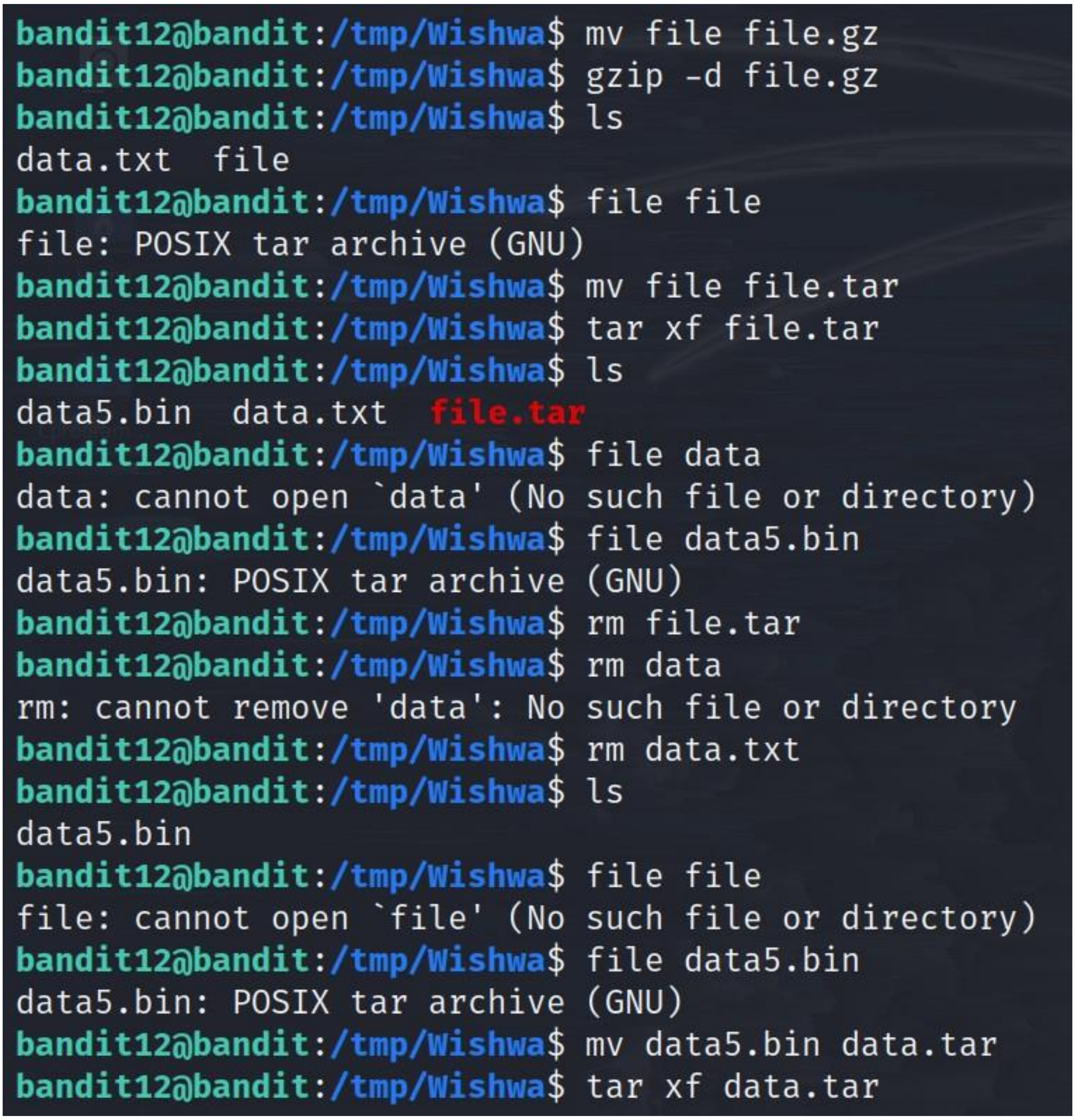
Password - JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv

LEVEL 12-13

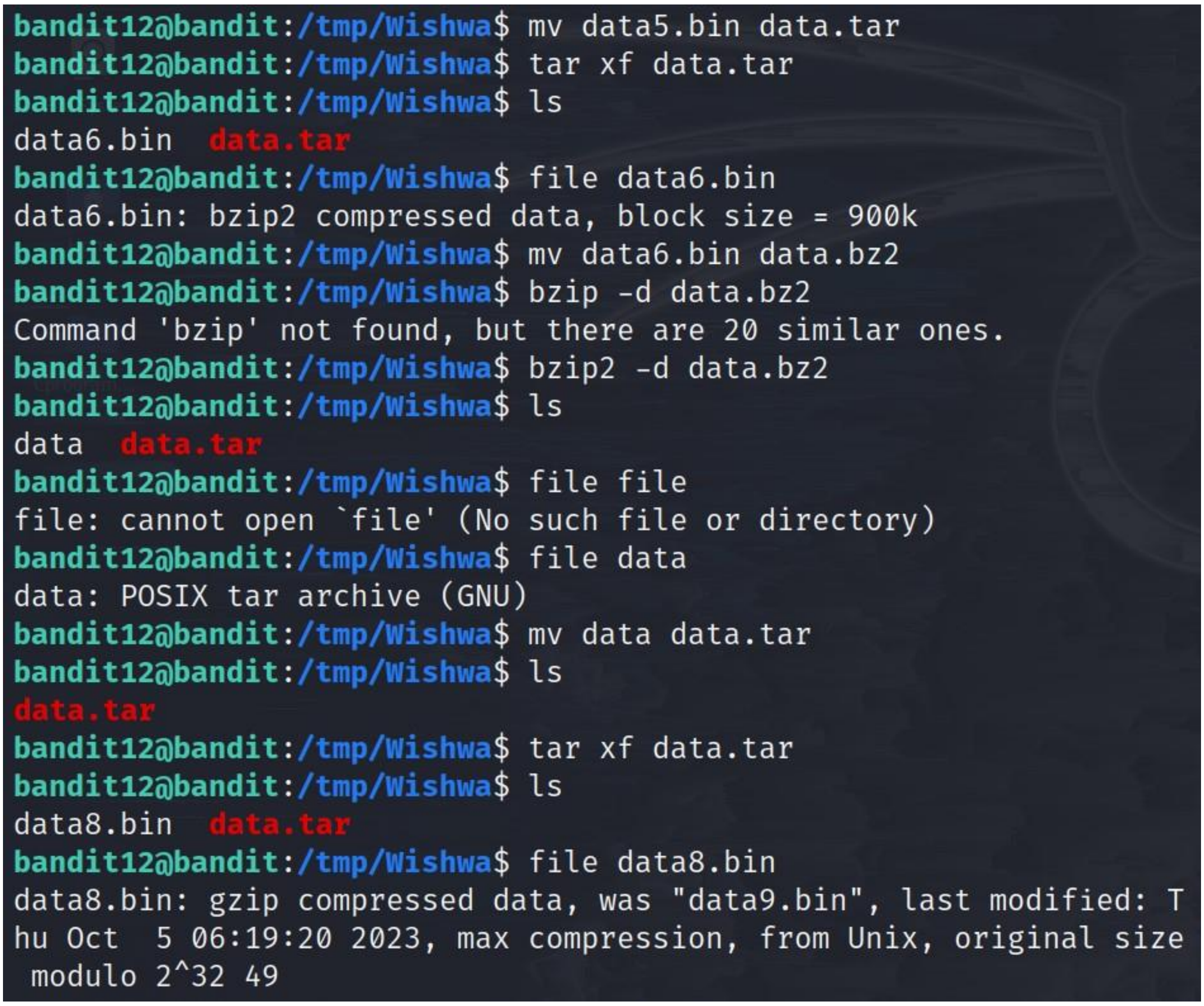
This is a walkthrough of OverTheWire Bandit challenge level 12, which entails using the “xxd” command to decompress a hex dump file, going to the working directory, and then using **gzip** and **bzip2** to decompress the required file extensions.then finds out that a few of the extracted files are actually archives. They keep decompressing until they reach a data file, at which point they advance to the next level by using the "cat" programme to expose the password.



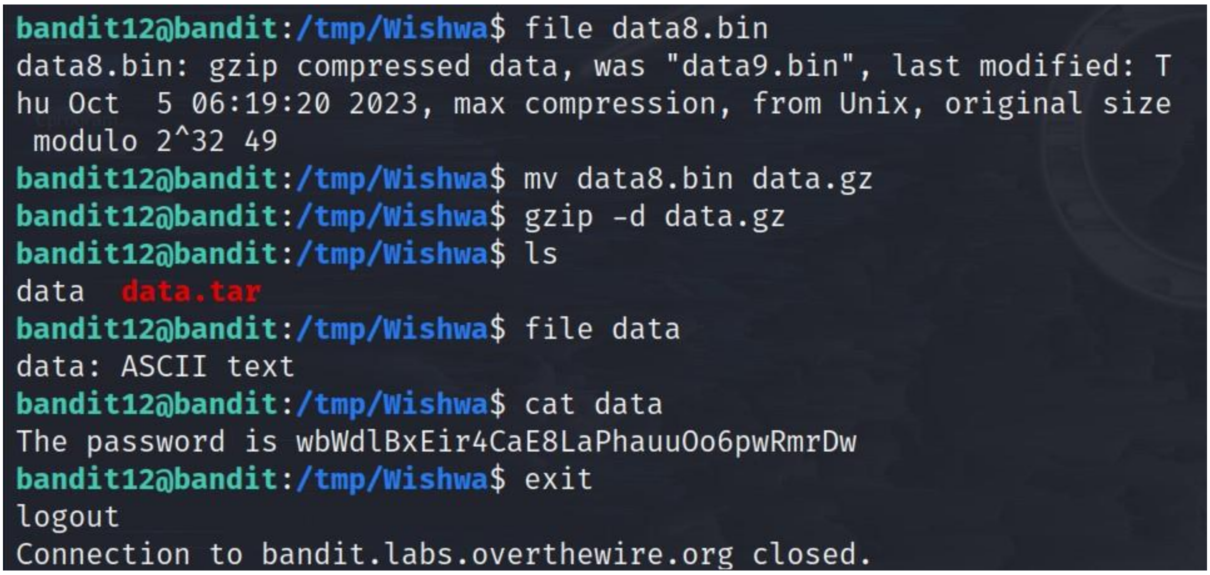
```
bandit12@bandit:~$ ls
data.txt
bandit12@bandit:~$ mkdir /tmp/Wishwa
bandit12@bandit:~$ cp data.txt /tmp/Wishwa
bandit12@bandit:~$ cd /tmp/Wishwa
bandit12@bandit:/tmp/Wishwa$ ls
data.txt
bandit12@bandit:/tmp/Wishwa$ xxd -r data.txt > data
bandit12@bandit:/tmp/Wishwa$ ls
data  data.txt
bandit12@bandit:/tmp/Wishwa$ file data
data: gzip compressed data, was "data2.bin", last modified: Thu Oc
t  5 06:19:20 2023, max compression, from Unix, original size modu
lo 2^32 573
bandit12@bandit:/tmp/Wishwa$ mv data file.gz
bandit12@bandit:/tmp/Wishwa$ gzip -d file.gz
bandit12@bandit:/tmp/Wishwa$ ls
data.txt  file
bandit12@bandit:/tmp/Wishwa$ file file
file: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/Wishwa$ mv file file.bz2
bandit12@bandit:/tmp/Wishwa$ man bzip2
bandit12@bandit:/tmp/Wishwa$ bzip2 -d file.bz2
bandit12@bandit:/tmp/Wishwa$ ls
data.txt  file
bandit12@bandit:/tmp/Wishwa$ file file
file: gzip compressed data, was "data4.bin", last modified: Thu Oc
t  5 06:19:20 2023, max compression, from Unix, original size modu
```



```
bandit12@bandit:/tmp/Wishwa$ mv file file.gz
bandit12@bandit:/tmp/Wishwa$ gzip -d file.gz
bandit12@bandit:/tmp/Wishwa$ ls
data.txt  file
bandit12@bandit:/tmp/Wishwa$ file file
file: POSIX tar archive (GNU)
bandit12@bandit:/tmp/Wishwa$ mv file file.tar
bandit12@bandit:/tmp/Wishwa$ tar xf file.tar
bandit12@bandit:/tmp/Wishwa$ ls
data5.bin  data.txt  file.tar
bandit12@bandit:/tmp/Wishwa$ file data
data: cannot open `data' (No such file or directory)
bandit12@bandit:/tmp/Wishwa$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/Wishwa$ rm file.tar
bandit12@bandit:/tmp/Wishwa$ rm data
rm: cannot remove 'data': No such file or directory
bandit12@bandit:/tmp/Wishwa$ rm data.txt
bandit12@bandit:/tmp/Wishwa$ ls
data5.bin
bandit12@bandit:/tmp/Wishwa$ file file
file: cannot open `file' (No such file or directory)
bandit12@bandit:/tmp/Wishwa$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/Wishwa$ mv data5.bin data.tar
bandit12@bandit:/tmp/Wishwa$ tar xf data.tar
```



```
bandit12@bandit:/tmp/Wishwa$ mv data5.bin data.tar
bandit12@bandit:/tmp/Wishwa$ tar xf data.tar
bandit12@bandit:/tmp/Wishwa$ ls
data6.bin  data.tar
bandit12@bandit:/tmp/Wishwa$ file data6.bin
data6.bin: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/Wishwa$ mv data6.bin data.bz2
bandit12@bandit:/tmp/Wishwa$ bzip -d data.bz2
Command 'bzip' not found, but there are 20 similar ones.
bandit12@bandit:/tmp/Wishwa$ bzip2 -d data.bz2
bandit12@bandit:/tmp/Wishwa$ ls
data  data.tar
bandit12@bandit:/tmp/Wishwa$ file file
file: cannot open `file' (No such file or directory)
bandit12@bandit:/tmp/Wishwa$ file data
data: POSIX tar archive (GNU)
bandit12@bandit:/tmp/Wishwa$ mv data data.tar
bandit12@bandit:/tmp/Wishwa$ ls
data.tar
bandit12@bandit:/tmp/Wishwa$ tar xf data.tar
bandit12@bandit:/tmp/Wishwa$ ls
data8.bin  data.tar
bandit12@bandit:/tmp/Wishwa$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: T
hu Oct  5 06:19:20 2023, max compression, from Unix, original size
 modulo 2^32 49
```



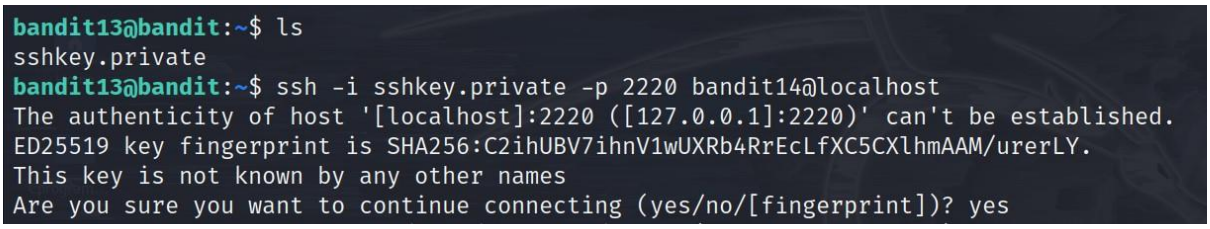
```
bandit12@bandit:/tmp/Wishwa$ file data8.bin
data8.bin: gzip compressed data, was "data9.bin", last modified: T
hu Oct  5 06:19:20 2023, max compression, from Unix, original size
 modulo 2^32 49
bandit12@bandit:/tmp/Wishwa$ mv data8.bin data.gz
bandit12@bandit:/tmp/Wishwa$ gzip -d data.gz
bandit12@bandit:/tmp/Wishwa$ ls
data  data.tar
bandit12@bandit:/tmp/Wishwa$ file data
data: ASCII text
bandit12@bandit:/tmp/Wishwa$ cat data
The password is wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw
bandit12@bandit:/tmp/Wishwa$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

LEVEL 13-14

In Unix-like operating systems, the ssh command is used to establish a secure SSH (Secure Shell) connection to a remote server. When establishing a connection to a distant server, the **-i** option is used to indicate the identity file (private key) that will be used for authentication.

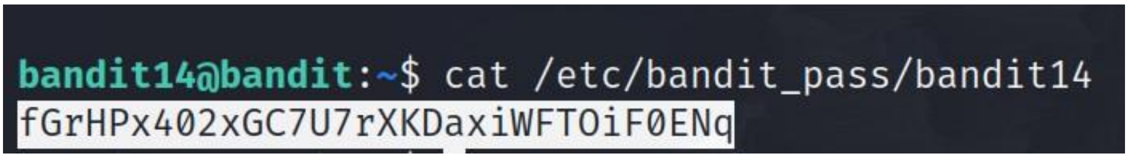
ssh -I sshkey.private -p 2220 bandit14@localhost



```
bandit13@bandit:~$ ls
sshkey.private
bandit13@bandit:~$ ssh -i sshkey.private -p 2220 bandit14@localhost
The authenticity of host '[localhost]: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
```

Through sshkey.private we can ssh into bandit14 and cat "**/etc/bandit_pass/bandit14**"

Then you can get the password for the next level.



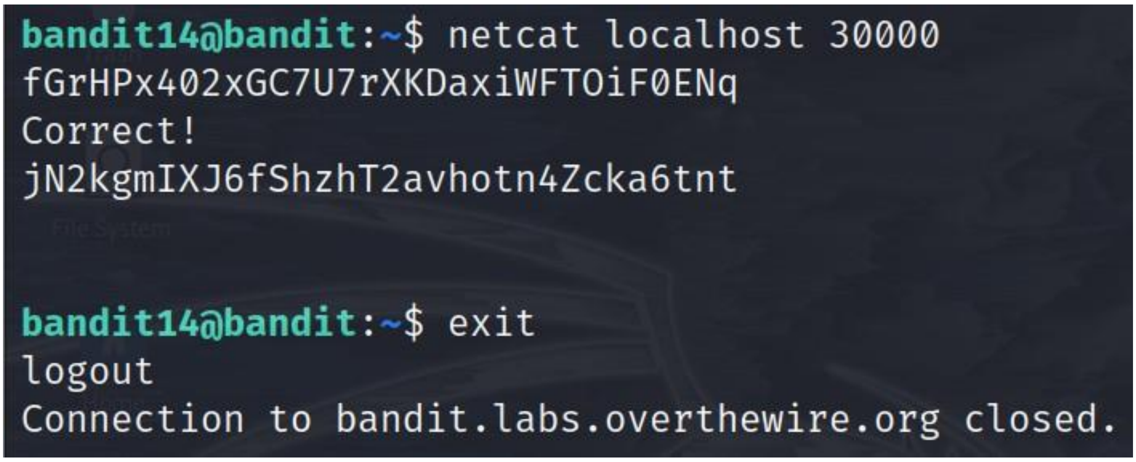
```
bandit14@bandit:~$ cat /etc/bandit_pass/bandit14
fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq
```

Password - fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq

LEVEL 14-15

In our quest to progress to the next level, we were informed about a service running on **port 30000**. To interact with this service, we employ the **"netcat"** command. Upon attempting a connection and providing a random value, we receive a message indicating that the password entered is incorrect. Armed with the knowledge that the current level's password is stored in "/etc/band_pass/bandit14" we decide to input this value into the service to see if we can retrieve the password for the next level. This strategic approach allows us to potentially gain access to the next level by leveraging the existing password information stored in the specified directory.

netcat localhost 30000



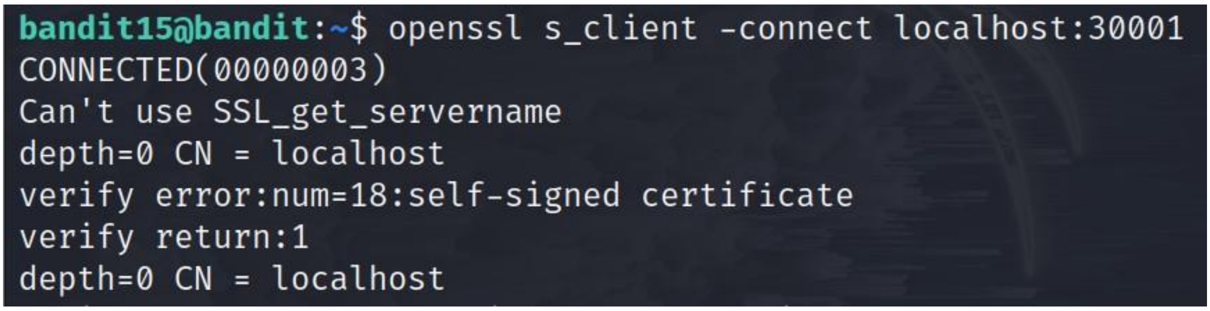
```
bandit14@bandit:~$ netcat localhost 30000
fGrHPx402xGC7U7rXKDaxiWFTOiF0ENq
Correct!
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt


bandit14@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt

LEVEL 15-16

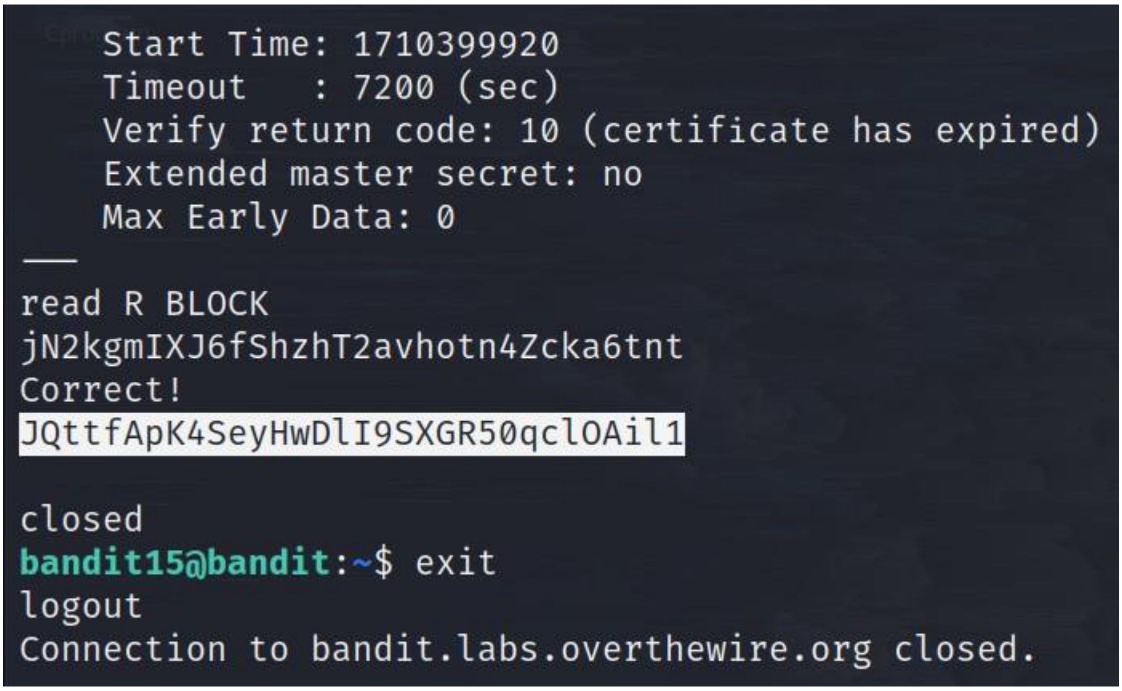
"openssl" and **"s_client"** are the commands we use to establish an SSL-encrypted connection to a service on **port 30001**. We can create a secure SSL connection to our machine's services with this combination. The desired service running on port 30,001 can be connected to over an SSL connection by using **"s_client,"** which allows for secure communication.

openssl s_client -connect localhost:30001



```
bandit15@bandit:~$ openssl s_client -connect localhost:30001
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 CN = localhost
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = localhost
```

Enter the right password to check if you have the one for the next level.

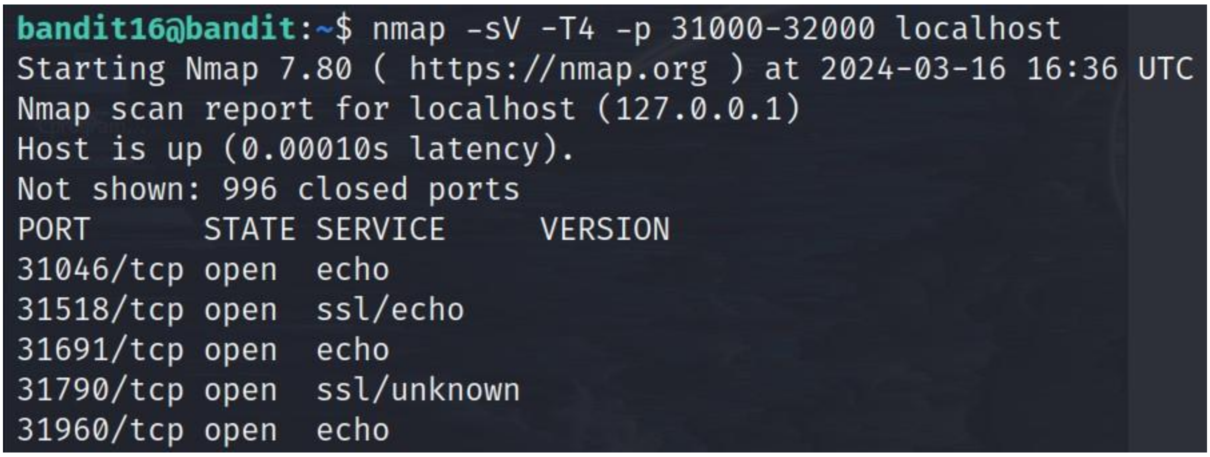


```
    Start Time: 1710399920
    Timeout   : 7200 (sec)
    Verify return code: 10 (certificate has expired)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
jN2kgmIXJ6fShzhT2avhotn4Zcka6tnt
Correct!
JQttfApK4SeyHwDlI9SXGR50qclOAil1

closed
bandit15@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.
```

Password - JQttfApK4SeyHwDlI9SXGR50qclOAil1

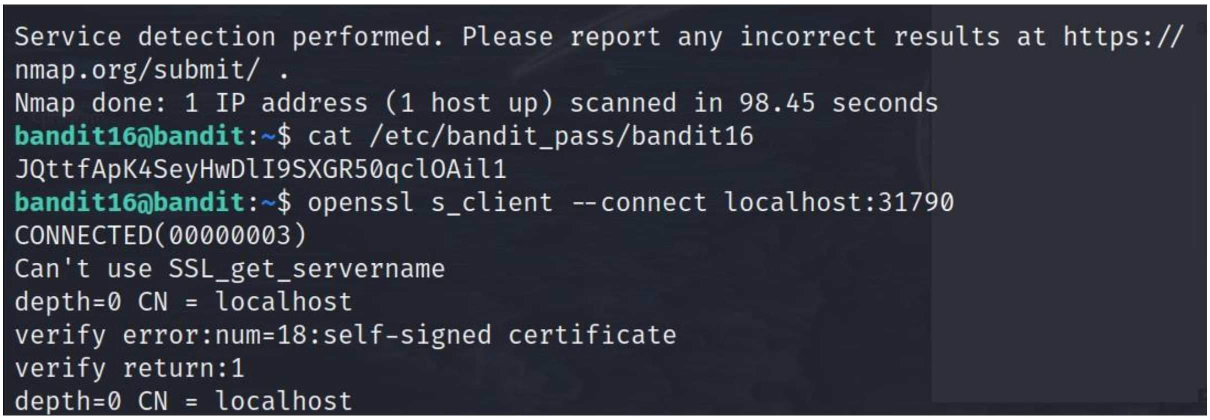
LEVEL 16-17

The **"nmap"** programme is used to locate the service we require, which is operating between ports **31000** and **32000**. While we can define the range of ports to scan using the **"-p"** signal, we can increase the scanning speed by using the **"-T4"** flag. The versions of the services that are operating on such ports are also detected and identified by us using the **"-sV"** flag. We are able to identify the precise service that we need within the designated port range thanks to this thorough search.

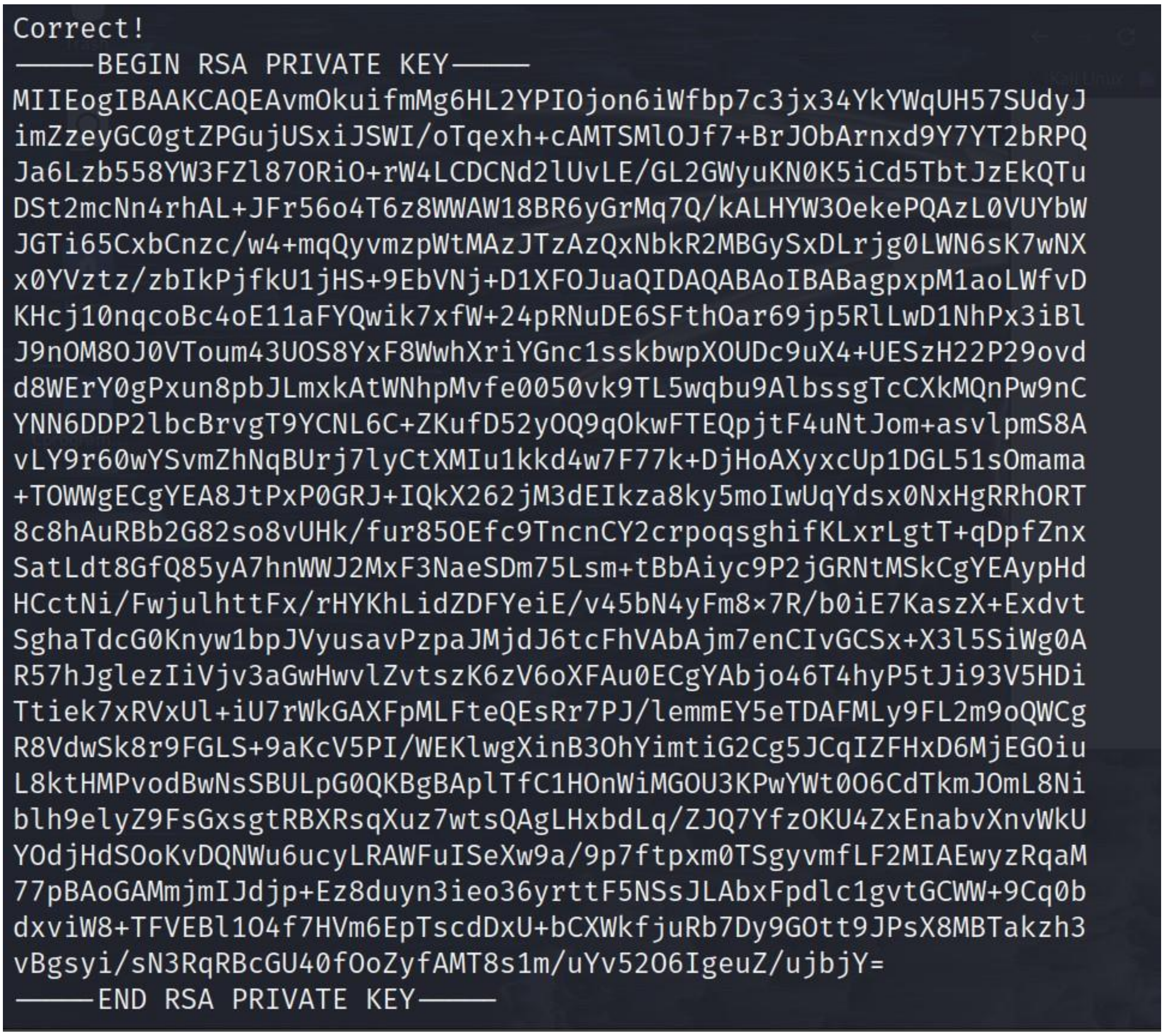


```
bandit16@bandit:~$ nmap -sV -T4 -p 31000-32000 localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-16 16:36 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00010s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE     VERSION
31046/tcp open  echo
31518/tcp open  ssl/echo
31691/tcp open  echo
31790/tcp open  ssl/unknown
31960/tcp open  echo
```

We use the commands **"openssl"** and **"s_client"** to connect to port **31790** and supply the password for the current user because we are aware that the service is secured using SSL.

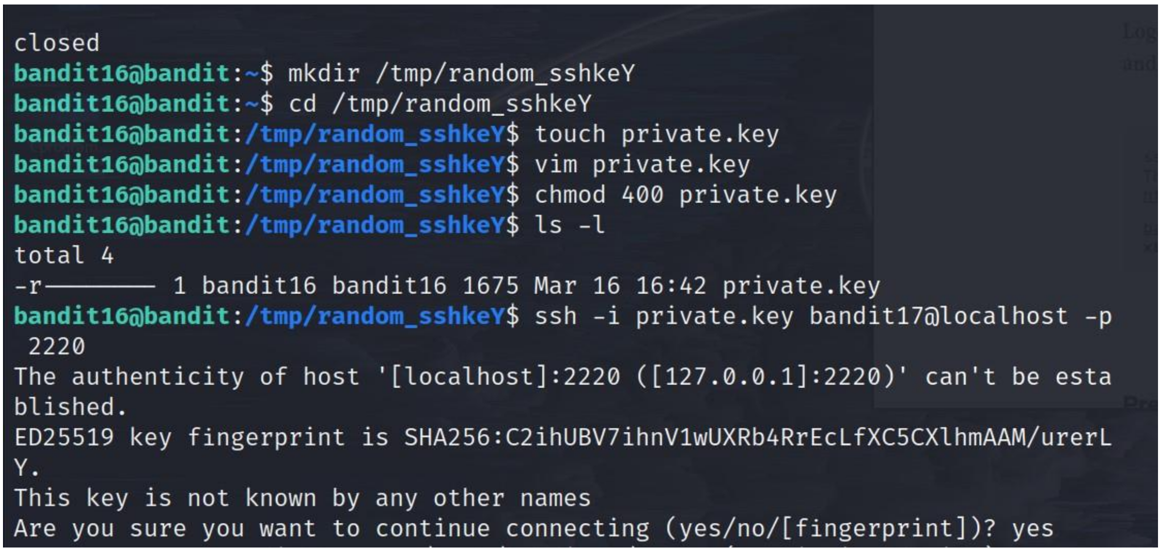


```
Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 98.45 seconds
bandit16@bandit:~$ cat /etc/bandit_pass/bandit16
JQttfApK4SeyHwDlI9SXGR50qclOAil1
bandit16@bandit:~$ openssl s_client --connect localhost:31790
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 CN = localhost
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = localhost
```



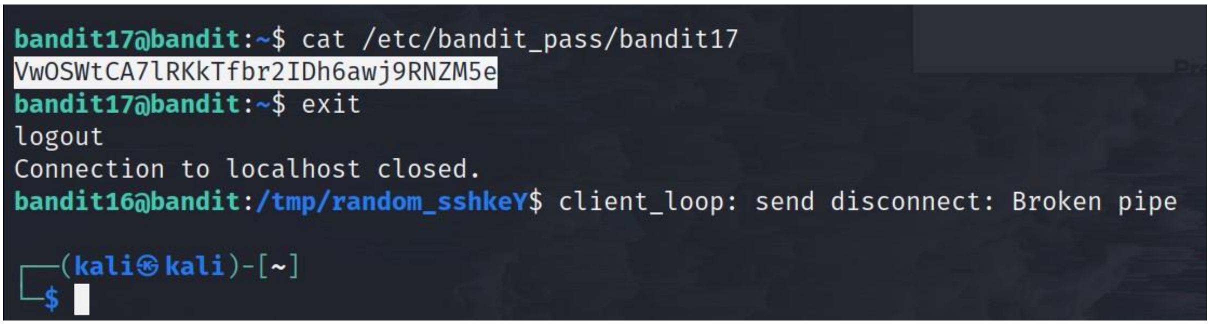
```
Correct!
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAvmOkuifmMg6HL2YPIOjon6iWfbp7c3jx34YkYWqUH57SUdyJ
imZzeyGC0gtZPGujUSxiJSWI/oTqexh+cAMTSMlOJf7+BrJObArnxd9Y7YT2bRPQ
Ja6Lzb558YW3FZl87ORiO+rW4LCDCNd2lUvLE/GL2GWyuKN0K5iCd5TbtJzEkQTu
DSt2mcNn4rhAL+JFr56o4T6z8WWAW18BR6yGrMq7Q/kALHYW3OekePQAzL0VUYbW
JGTi65CxbCnzc/w4+mqQyvmzpWtMAzJTzAzQxNbkR2MBGySxDLrjg0LWN6sK7wNX
x0YVztz/zbIkPjfkU1jHS+9EbVNj+D1XFOJuaQIDAQABAoIBABagpxpM1aoLWfvD
KHcj10nqcoBc4oE11aFYQwik7xfW+24pRNuDE6SFthOar69jp5RlLwD1NhPx3iBl
J9nOM8OJ0VToum43UOS8YxF8WwhXriYGnc1sskbwpXOUDc9uX4+UESzH22P29ovd
d8WErY0gPxun8pbJLmxkAtWNhpMvfe0050vk9TL5wqbu9AlbssgTcCXkMQnPw9nC
YNN6DDP2lbcBrvgT9YCNL6C+ZKufD52yOQ9qOkwFTEQpjtF4uNtJom+asvlpmS8A
vLY9r60wYSvmZhNqBUrj7lyCtXMIu1kkd4w7F77k+DjHoAXyxcUp1DGL51sOmama
+TOWWgECgYEA8JtPxP0GRJ+IQkX262jM3dEIkza8ky5moIwUqYdsx0NxHgRRhORT
8c8hAuRBb2G82so8vUHk/fur85OEfc9TncnCY2crpoqsghifKLxrLgtT+qDpfZnx
SatLdt8GfQ85yA7hnWWJ2MxF3NaeSDm75Lsm+tBbAiyc9P2jGRNtMSkCgYEAypHd
HCctNi/FwjulhttFx/rHYKhLidZDFYeiE/v45bN4yFm8x7R/b0iE7KaszX+Exdvt
SghaTdcG0Knyw1bpJVyusavPzpaJMjdJ6tcFhVAbAjm7enCIvGCSx+X3l5SiWg0A
R57hJglezIiVjv3aGwHwvlZvtszK6zV6oXFAu0ECgYAbjo46T4hyP5tJi93V5HDi
Ttiek7xRVxUl+iU7rWkGAXFpMLFteQEsRr7PJ/lemmEY5eTDAFMLy9FL2m9oQWCg
R8VdwSk8r9FGLS+9aKcV5PI/WEKlwgXinB3OhYimtiG2Cg5JCqIZFHxD6MjEGOiu
L8ktHMPvodBwNsSBULpG0QKBgBAplTfC1HOnWiMGOU3KPwYWt0O6CdTkmJOmL8Ni
blh9elyZ9FsGxsgtRBXRsqXuz7wtsQAgLHxbdLq/ZJQ7YfzOKU4ZxEnabvXnvWkU
YOdjHdSOoKvDQNWu6ucyLRAWFuISeXw9a/9p7ftpxm0TSgyvmfLF2MIAEwyzRqaM
77pBAoGAMmjmIJdjp+Ez8duyn3ieo36yrttF5NSsJLAbxFpdlc1gvtGCWW+9Cq0b
dxviW8+TFVEBl1O4f7HVm6EpTscdDxU+bCXWkfjuRb7Dy9GOtt9JPsX8MBTakzh3
vBgsyi/sN3RqRBcGU40fOoZyfAMT8s1m/uYv52O6IgeuZ/ujbjY=
-----END RSA PRIVATE KEY-----
```

We discover that in order to use SSH to access the next level, we must save the **RSA** Key in a file after obtaining it from the service. We choose to make a folder in the **"/tmp"** directory and continue from there because we do not have rights to create a file in the current directory.



```
closed
bandit16@bandit:~$ mkdir /tmp/random_sshkeY
bandit16@bandit:~$ cd /tmp/random_sshkeY
bandit16@bandit:/tmp/random_sshkeY$ touch private.key
bandit16@bandit:/tmp/random_sshkeY$ vim private.key
bandit16@bandit:/tmp/random_sshkeY$ chmod 400 private.key
bandit16@bandit:/tmp/random_sshkeY$ ls -l
total 4
-r-------- 1 bandit16 bandit16 1675 Mar 16 16:42 private.key
bandit16@bandit:/tmp/random_sshkeY$ ssh -i private.key bandit17@localhost -p
 2220
The authenticity of host '[localhost]:2220 ([127.0.0.1]:2220)' can't be esta
blished.
ED25519 key fingerprint is SHA256:C2ihUBV7ihnV1wUXRb4RrEcLfXC5CXlhmAAM/urerL
Y.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

We may now access the next level by using the key file and the ssh command. Then cat “**/etc/bandit_pass/bandit17**” and get the flag for the next level.



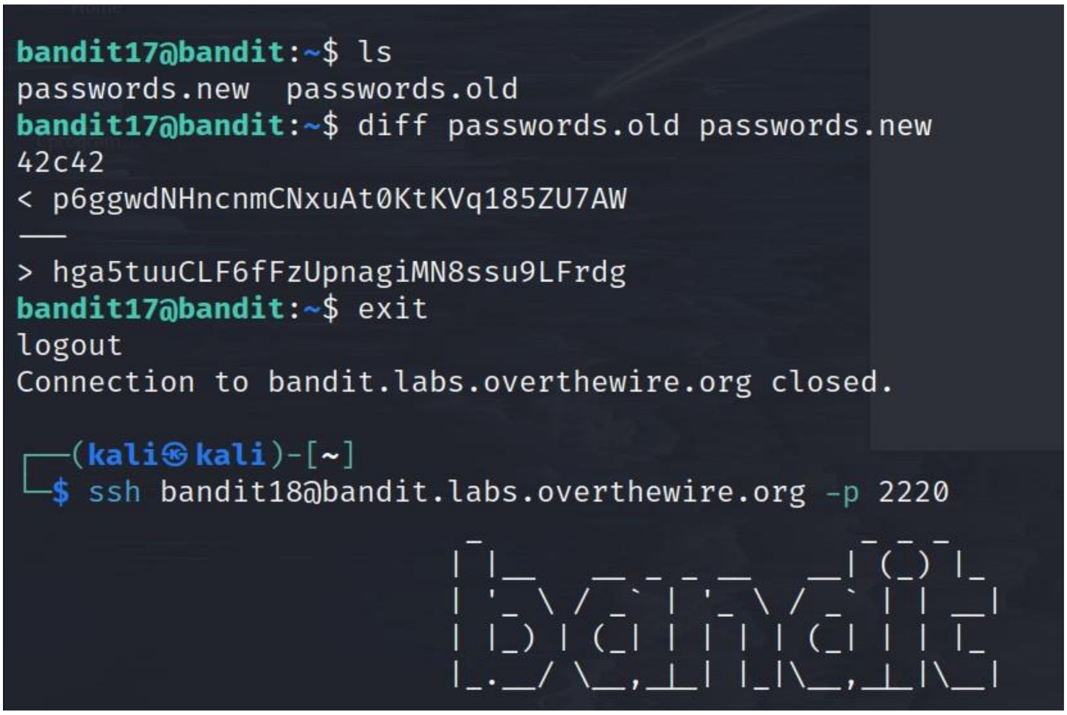
```
bandit17@bandit:~$ cat /etc/bandit_pass/bandit17
VwOSWtCA7lRKkTfbr2IDh6awj9RNZM5e
bandit17@bandit:~$ exit
logout
Connection to localhost closed.
bandit16@bandit:/tmp/random_sshkeY$ client_loop: send disconnect: Broken pipe

┌──(kali㉿kali)-[~]
└─$ 
```

Password - VwOSWtCA7lRKkTfbr2IDh6awj9RNZM5e

LEVEL 17-18

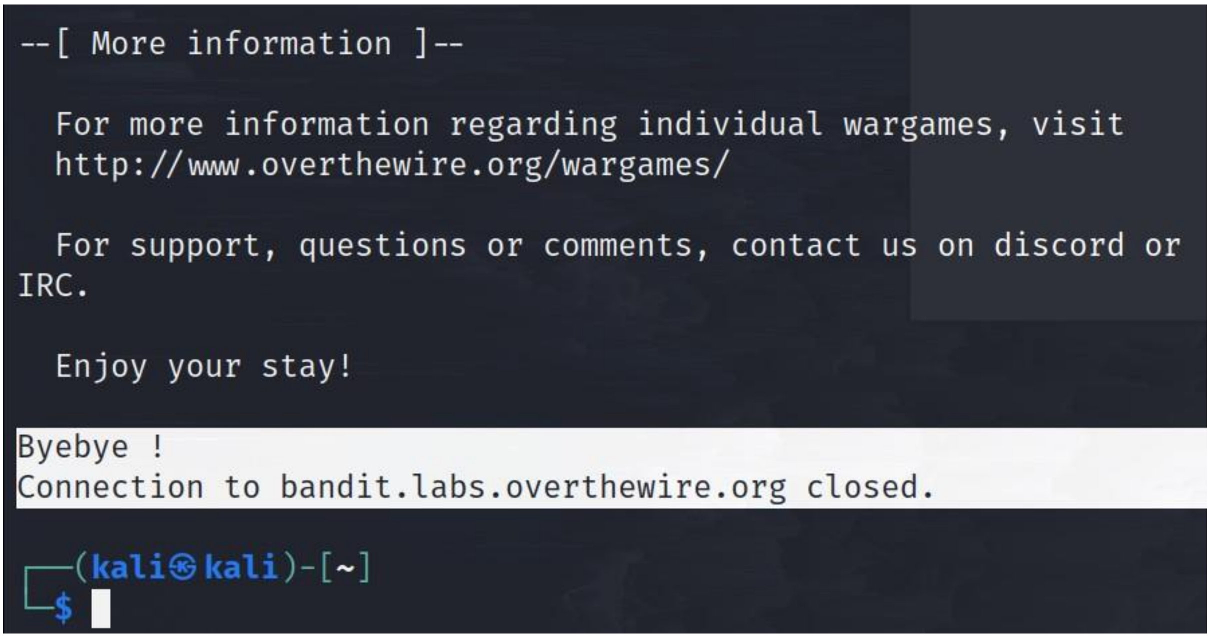
The password we need is included in the one line that separates the two files, as far as we can tell. Use of the **diff** command allows us to see the modifications made to files.

diff passwords.old passwords.new



```
bandit17@bandit:~$ ls
passwords.new  passwords.old
bandit17@bandit:~$ diff passwords.old passwords.new
42c42
< p6ggwdNHncnmCNxuAt0KtKVq185ZU7AW
---
> hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg
bandit17@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.

┌──(kali㉿kali)-[~]
└─$ ssh bandit18@bandit.labs.overthewire.org -p 2220
                      _                     _ _ _
                     | |__   __ _ _ __   __| (_) |_
                     | '_ \ / _` | '_ \ / _` | | __|
                     | |_) | (_| | | | | (_| | | |_
                     |_.__/ \__,_|_| |_|\__,_|_|\__|
```

Exit the current session and use the bandit18 password to get in to the next level.



```
--[ More information ]--

  For more information regarding individual wargames, visit
  http://www.overthewire.org/wargames/

  For support, questions or comments, contact us on discord or
IRC.

  Enjoy your stay!

Byebye !
Connection to bandit.labs.overthewire.org closed.

┌──(kali㉿kali)-[~]
└─$ 
```

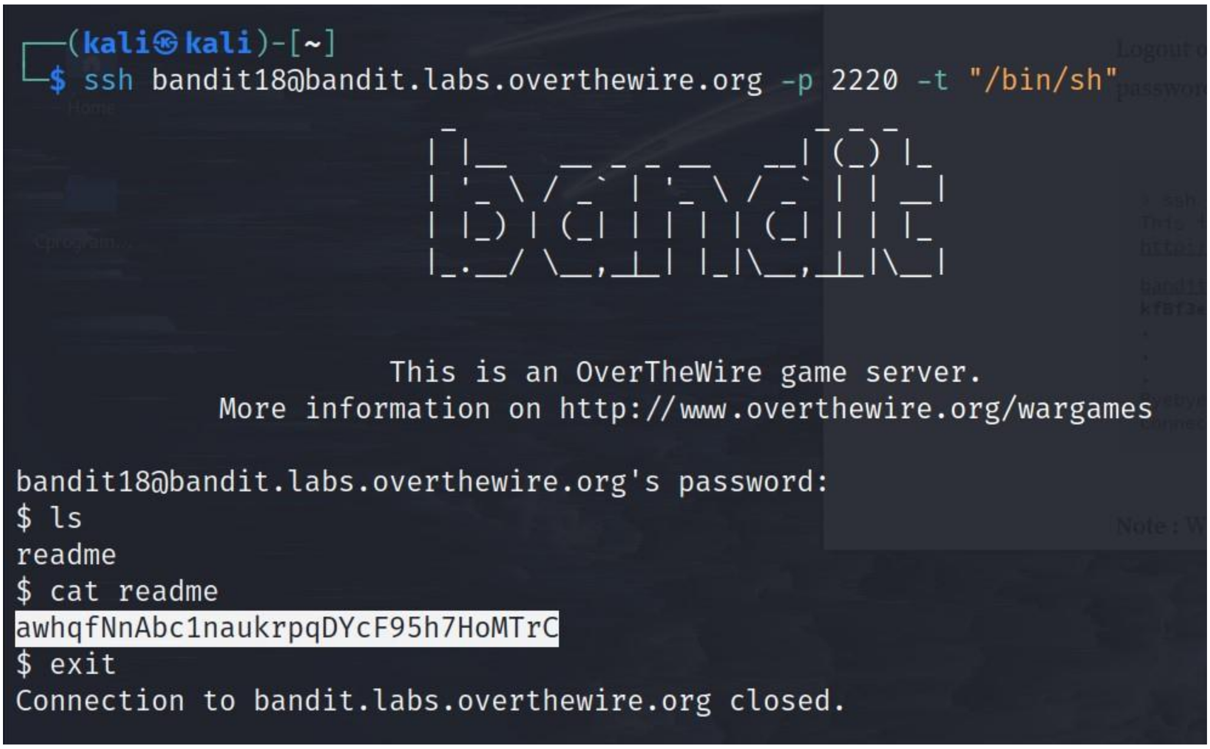
Password - hga5tuuCLF6fFzUpnagiMN8ssu9LFrdg

LEVEL 18-19

We can attempt SSH logging with some of the shells that we believe ought to be pre-configured on every server. The shell to be used to log into the system is specified using the SSH command's -t parameter.

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

Successfully logged in with the **"sh"** shell.Locate the password found in the **readme** file.

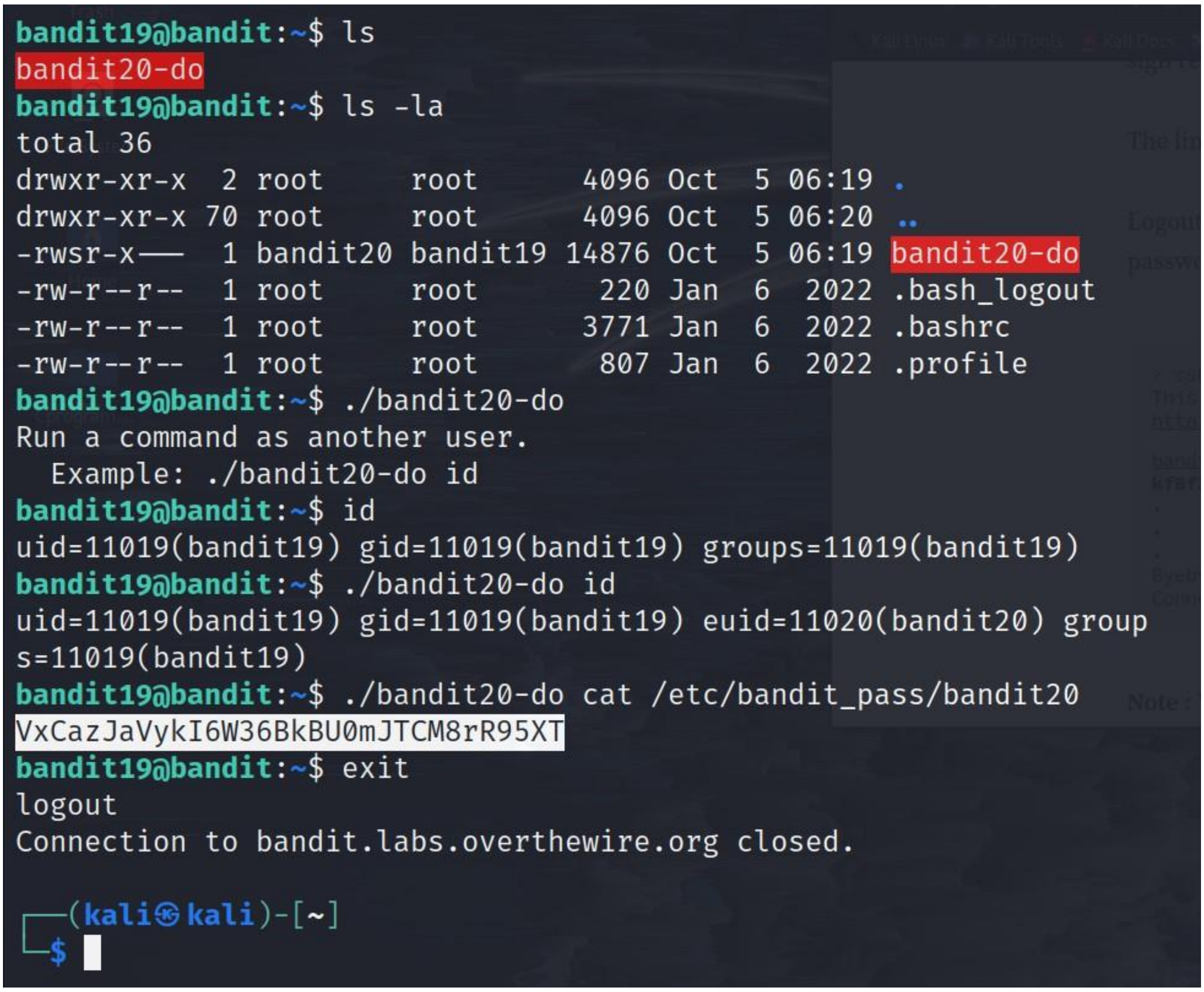


```
┌──(kali㉿kali)-[~]
└─$ ssh bandit18@bandit.labs.overthewire.org -p 2220 -t "/bin/sh"
                         _                     _ _ _
                        | |__   __ _ _ __   __| (_) |_
                        | '_ \ / _` | '_ \ / _` | | __|
                        | |_) | (_| | | | | (_|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


                      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
$ exit
Connection to bandit.labs.overthewire.org closed.
```

Password - awhqfNnAbc1naukrpqDYcF95h7HoMTrC

LEVEL 19-20

Checking the home directory reveals a binary file called **"bandit20-do."** After a brief investigation, it can be seen that bandit20 owns the binary and that it can be executed by the current user, bandit19. We observe that when we run the programme, our user ID switches to bandit20's, essentially giving us the ability to execute commands just like bandit20. Now that we have this capacity, we use the binary to get user bandit20's password.



```
bandit19@bandit:~$ ls
bandit20-do
bandit19@bandit:~$ ls -la
total 36
drwxr-xr-x  2 root     root      4096 Oct  5 06:19 .
drwxr-xr-x 70 root     root      4096 Oct  5 06:20 ..
-rwsr-x---  1 bandit20 bandit19 14876 Oct  5 06:19 bandit20-do
-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
bandit19@bandit:~$ ./bandit20-do
Run a command as another user.
  Example: ./bandit20-do id
bandit19@bandit:~$ id
uid=11019(bandit19) gid=11019(bandit19) groups=11019(bandit19)
bandit19@bandit:~$ ./bandit20-do id
uid=11019(bandit19) gid=11019(bandit19) euid=11020(bandit20) group
s=11019(bandit19)
bandit19@bandit:~$ ./bandit20-do cat /etc/bandit_pass/bandit20
VxCazJaVykI6W36BkBU0mJTCM8rR95XT
bandit19@bandit:~$ exit
logout
Connection to bandit.labs.overthewire.org closed.

┌──(kali㉿kali)-[~]
└─$ 
```

We have found the password for the level 20 !!!

Password - VxCazJaVykI6W36BkBU0mJTCM8rR95XT