Sri Lanka Institute of Information Technology

Bandit Over the Wire

IE2012 – Systems and Network Programming

Student Name	Registration Number	Date
P.T.D. Minipura	IT23298408	13/08/2024

Introduction

This report provides a comprehensive overview of the “Bandit” wargame which is a part of the OverTheWire platform. This game is introduced for the absolute beginners in order to provide the basic knowledge of Linux and Cybersecurity. It is a beginner-friendly wargame that enhances the hand skills of new players. Each level of Bandit presents a new challenge that requires players to use linux commands and techniques to find the hidden passwords and progress to the next level.

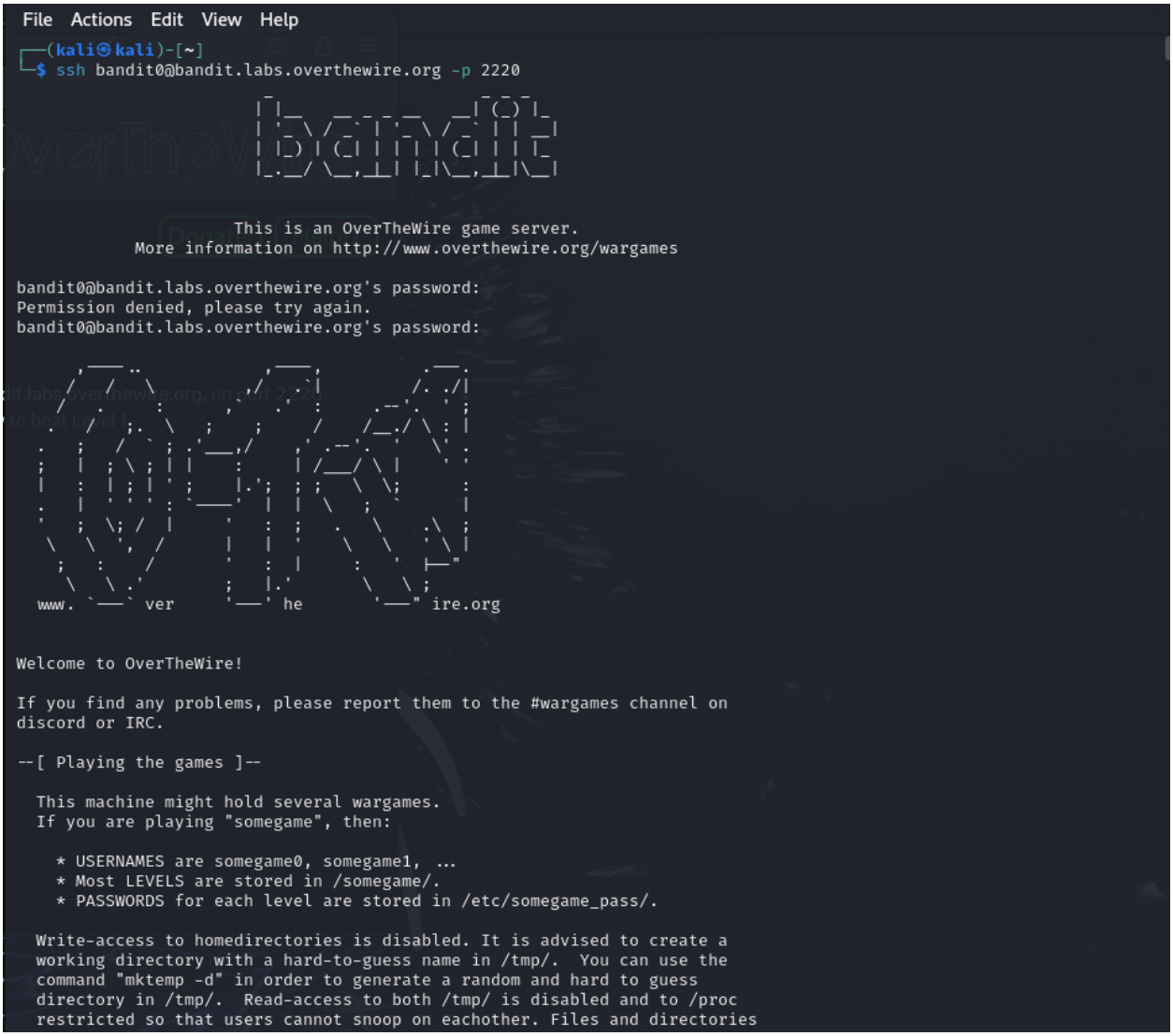
The primary goal of this report is to document/show the various levels encountered in the Bandit wargame, detail the challenges presented, and outline the methods and tools used to overcome each one. This report is structured to provide a clear walkthrough of each level, with explanations of the commands and strategies used.

Attempting this game open the scopes to widen the knowledge of Unix environment, cybersecurity and also the logical thinking and problem-solving abilities also been addressed.

Level 0

Password : bandit0

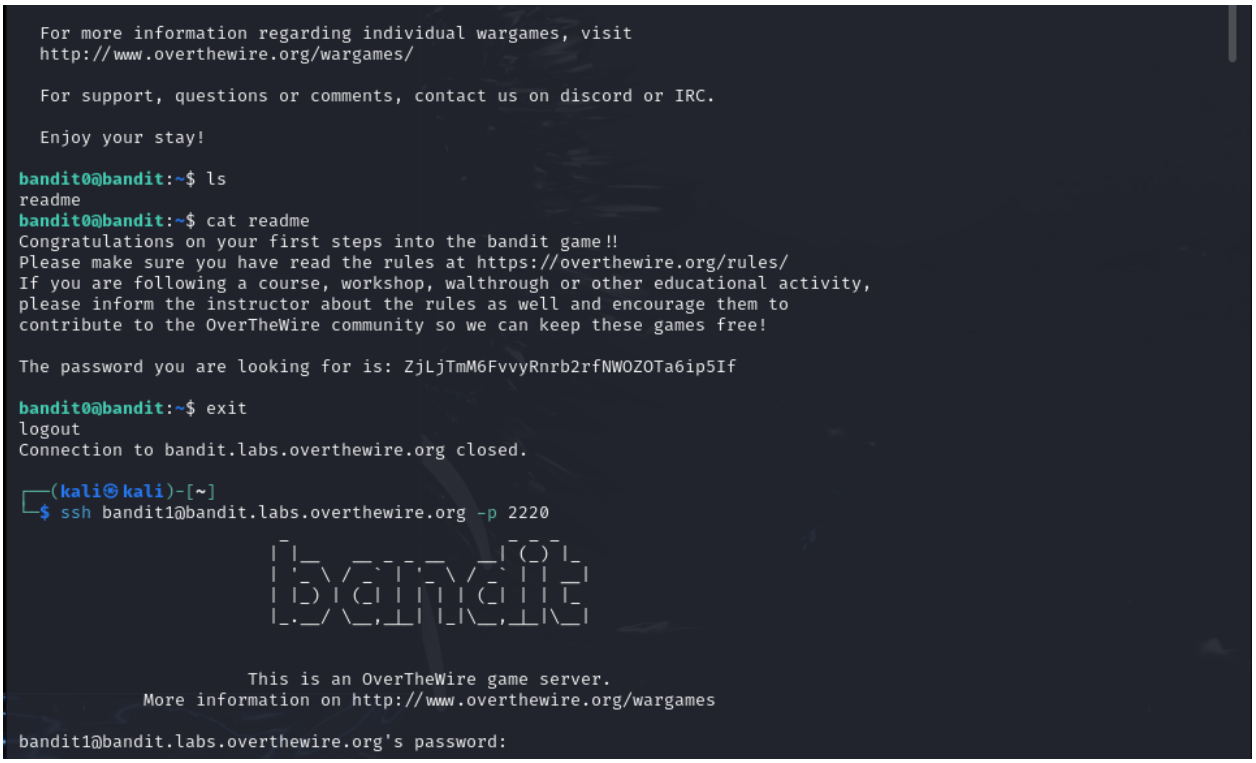
- The goal of the level 0 is to log into the game using SSH. The host needed to connect is bandit.labs.overthewire.org with port 2220.
- The username is ‘bandit0’
- The password id ‘bandit0’



Level 0 – 1

Password: ZjLjTmM6FvvyRnrb2rfNWOZOTa6ip5If

- The password is stored in a ‘readme’ file in home directory.



The commands:

ls – use to list the files

cat readme – use to open the ‘readme’ file

Level 1 – 2

Password: 263JGJPfgU6LtdEvgfWU1XP5yac29mFx

- The password for the next level is stored in a file named ‘ - ‘.

The commands:

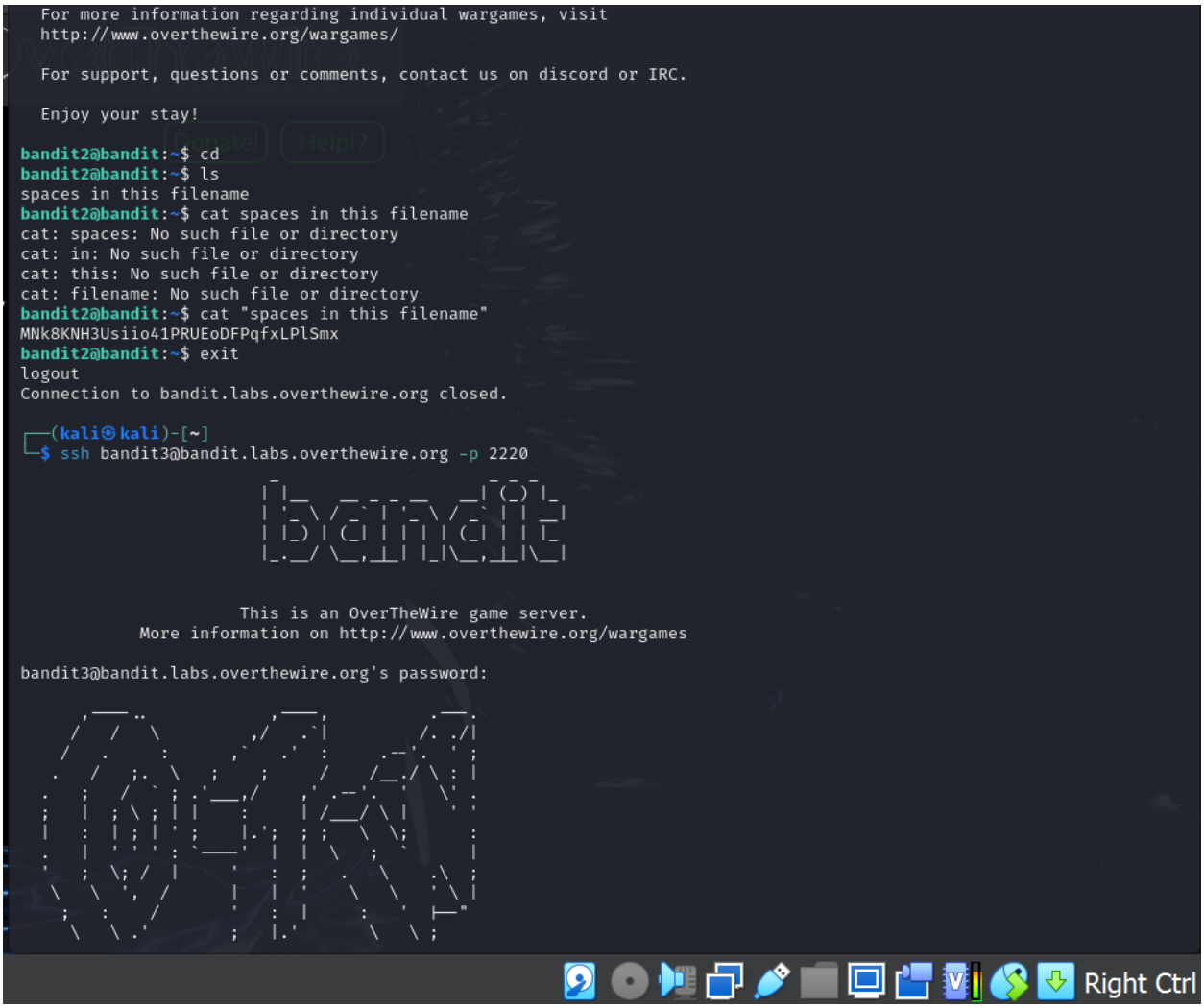
ls – Use to list the files

cat ./- - Use to open the ‘ – ‘ file

Level 2 - 3

Password: MNk8KNH3Usiio41PRUEoDFPqfxLPlSmx

- The password for the next level is stored in a file named “spaces in this filename”.



The commands:

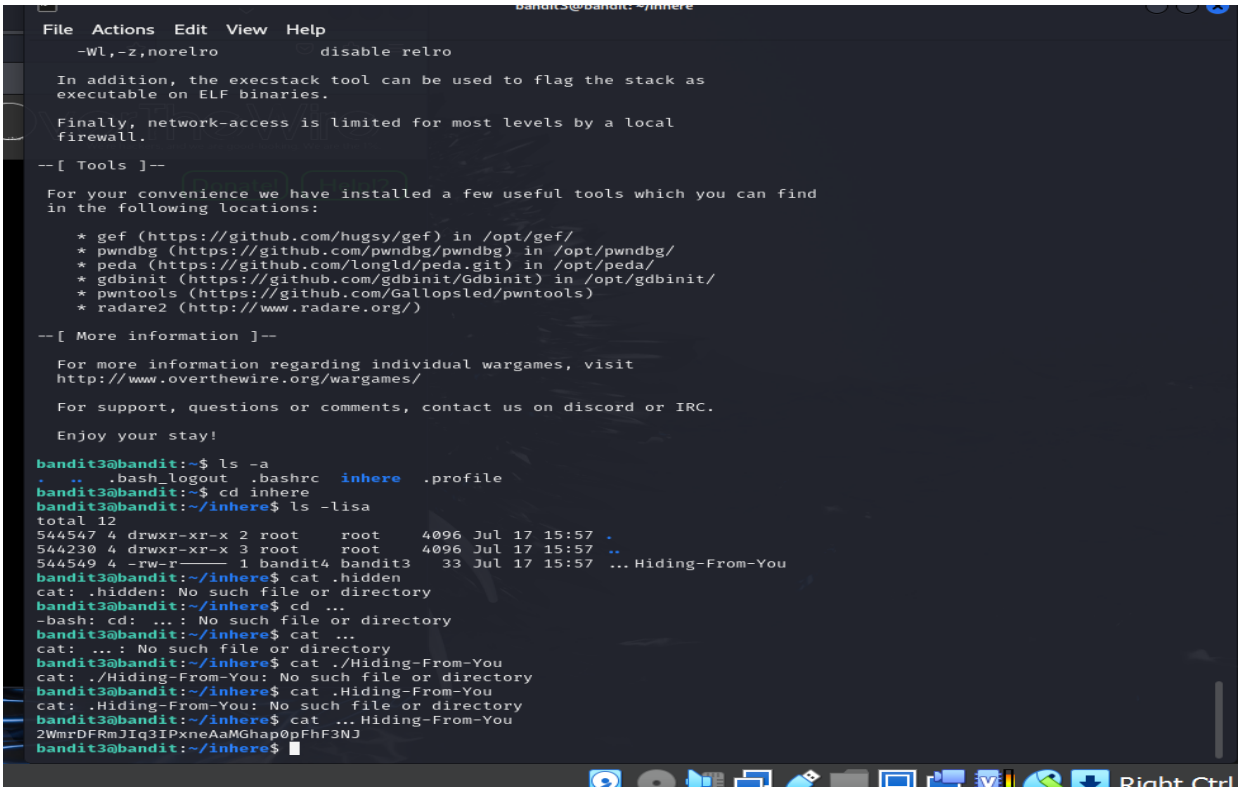
ls – Use to list the files

cat spaces in the filename – Use to access ‘spaces in the filename’ file

Level 3 – 4

Password: 2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ

- The password for the next level is hidden in the directory named ‘inhere’



The commands:

ls -lisa – list the files (Here directories are their) and their permissions

cat ...Hiding-From-You - open the file where password consists

Level 4 – 5

Password: 4oQYVPkxZOOEOO5pTW81FB8j8lxXGUQw

- The password for the next level is stored in the only human-readable file in the '**inhere**' directory.

The commands:

ls – list the files

cd inhere – open the directory 'inhere'

ls – list the files in directory

find -type f -exec file {} + | grep "ASCII text"

find: This is a command used to search for files and directories within a directory hierarchy.

-type f: This option restricts the search to files only (not directories)

-exec: This option allows you to execute a command on the files found by find.

file: The file command is used to determine the type of file. It reads the file and outputs a description of its contents, such as "ASCII text", "binary", "image", etc.

{}: This is a placeholder that represents each file found by the find command.

+ : This allows find to pass multiple files at once to the file command, which is more efficient than executing file separately for each file.

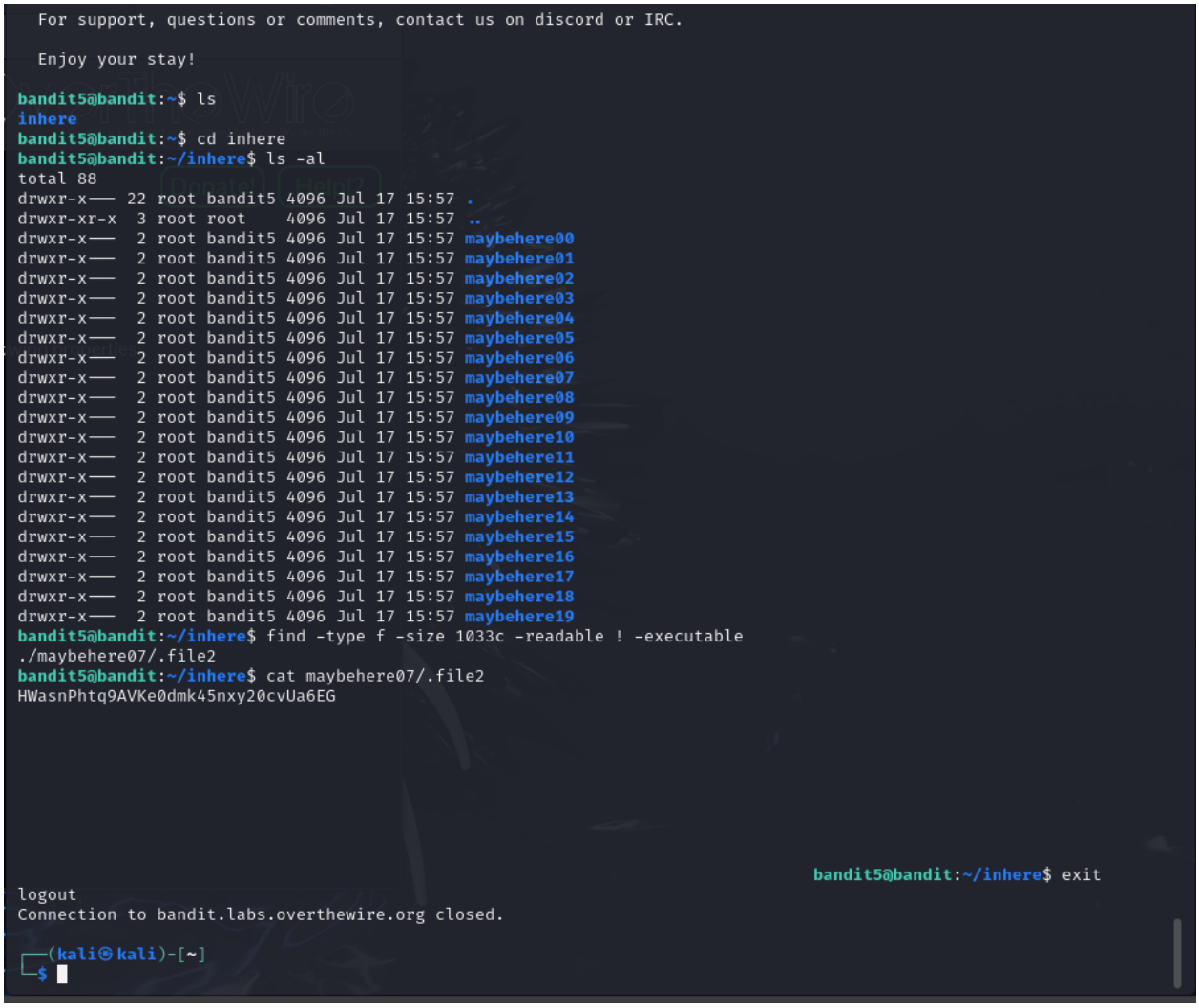
'|' : This is a pipe, which passes the output of the previous command (find -exec file) as input to the next command (grep).

grep "ASCII text": grep searches the output for the string "ASCII text". This filters the results to show only files that are identified as ASCII text files by the file command.

Level 5 – 6

Password: HWasnPhtq9AVKe0dmk45nxy20cvUa6EG

- The password for the next level is stored in a file somewhere under the **inhere** directory and has all the following properties:
  1. human-readable 2. 1033 bytes in size 3. not executable



The commands:

Find inhere/ -type f -size 1033c -readable ! -executable

Find here – searches within the ‘inhere’ directory

-type f - only search for files

-size 1033c – search files with exactly 1033 bytes in size (c is used represent bytes)

-readable – ensures the file is human-readable

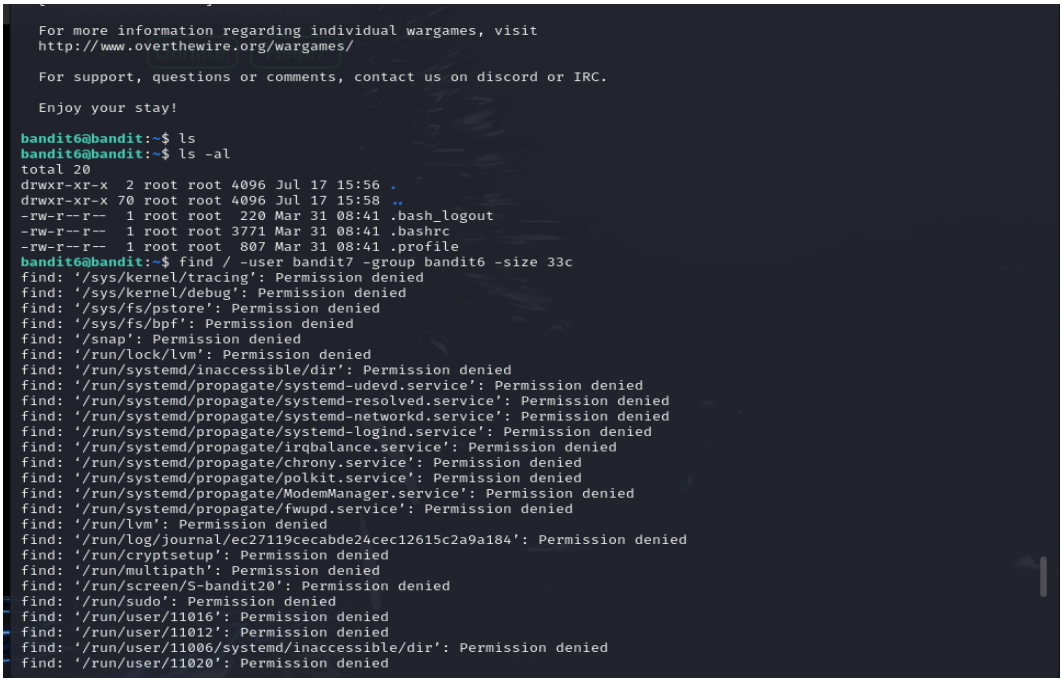
! -executable - ensures the file is not executable

Level 6 - 7

Password: morbNTDkSW6jIlUc0ymOdMaLnOlFVAaj

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



The commands:

Find / -user bandit7 -group bandit6 -size 33c

Find / - starts searching from root directory

-user bandit7 – searches files owned by the user ‘bandit7’

-group bandit6 - searches files owned by the group ‘bandit6’

-size 33c – searches files size 33bytes exactly

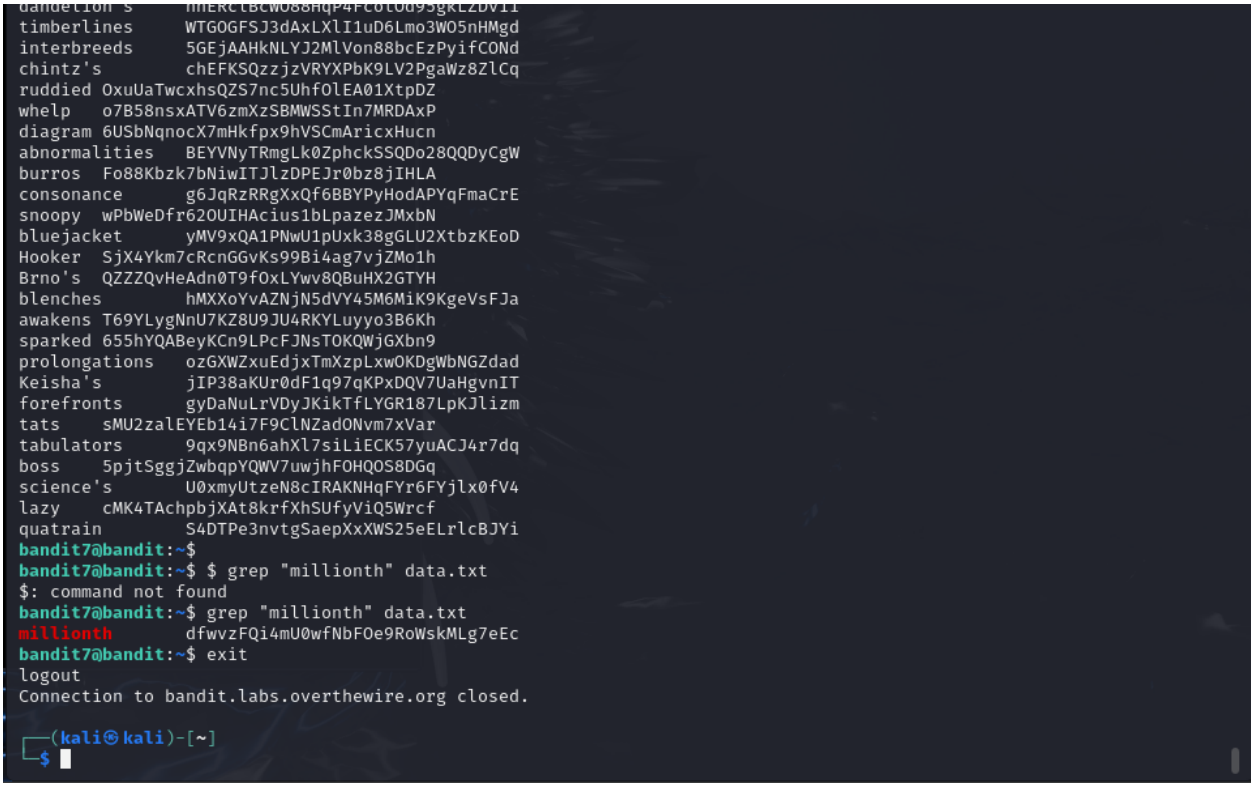
- Find the password file path from the list and open it with ‘cat’ command

  cat /var/lib/dpkg/info/bandit7.password

Level 7 - 8

Password: dfwvzFQi4mU0wfNbFOe9RoWskMLg7eEc

- The password for the next level is stored in the file **data.txt** next to the word **millionth.**
- The word ‘millionth’ has be filter out.



The commands:

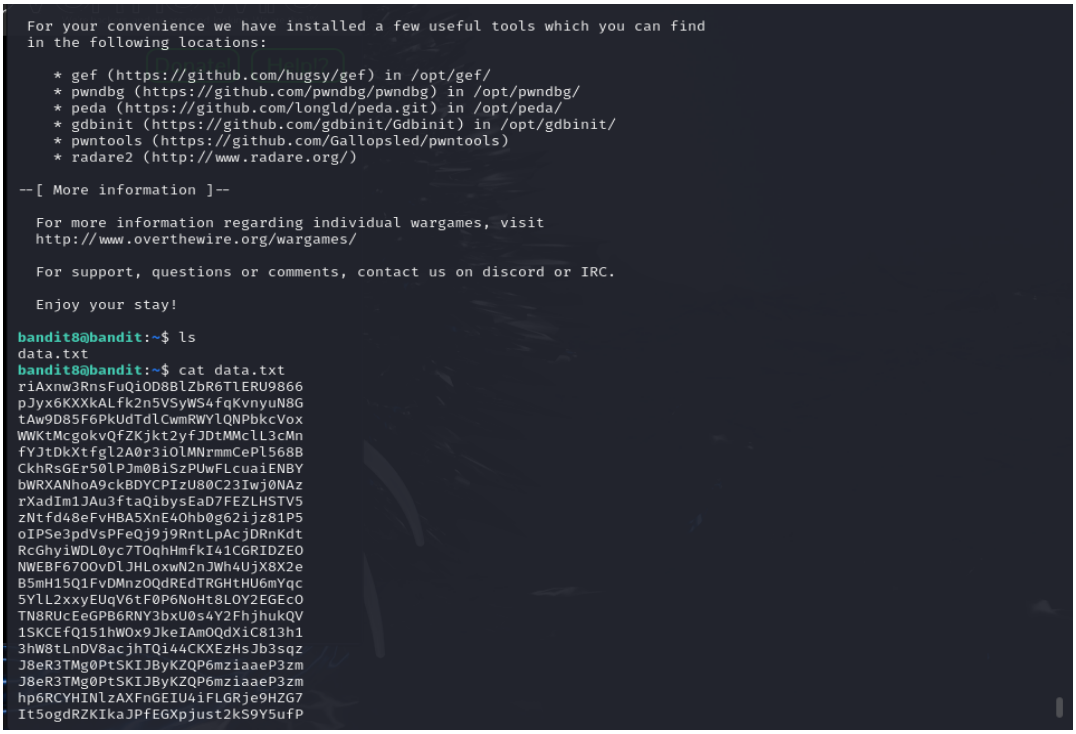
grep "millionth" data.txt

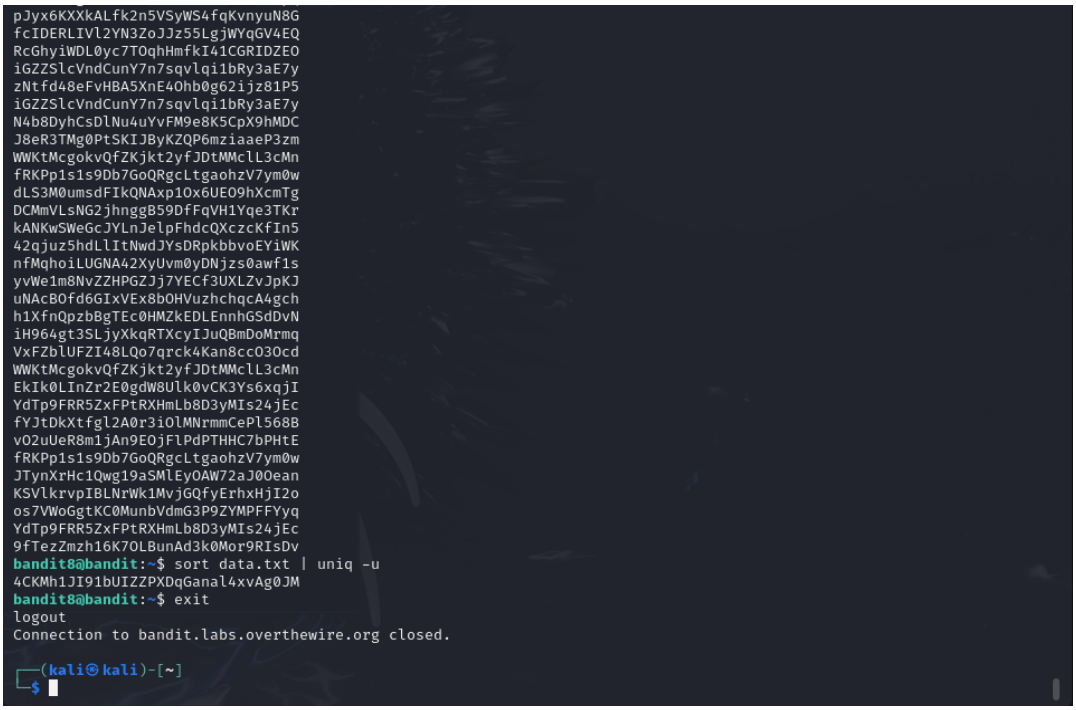
This command will search for the line that contains the word “millionth”

Level 8 – 9

Password: 4CKMh1JI91bUIZZPXDqGanal4xvAg0JM

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





The command:

sort data.txt | uniq -u

sort data.txt – sorts the lines in alphabetical order

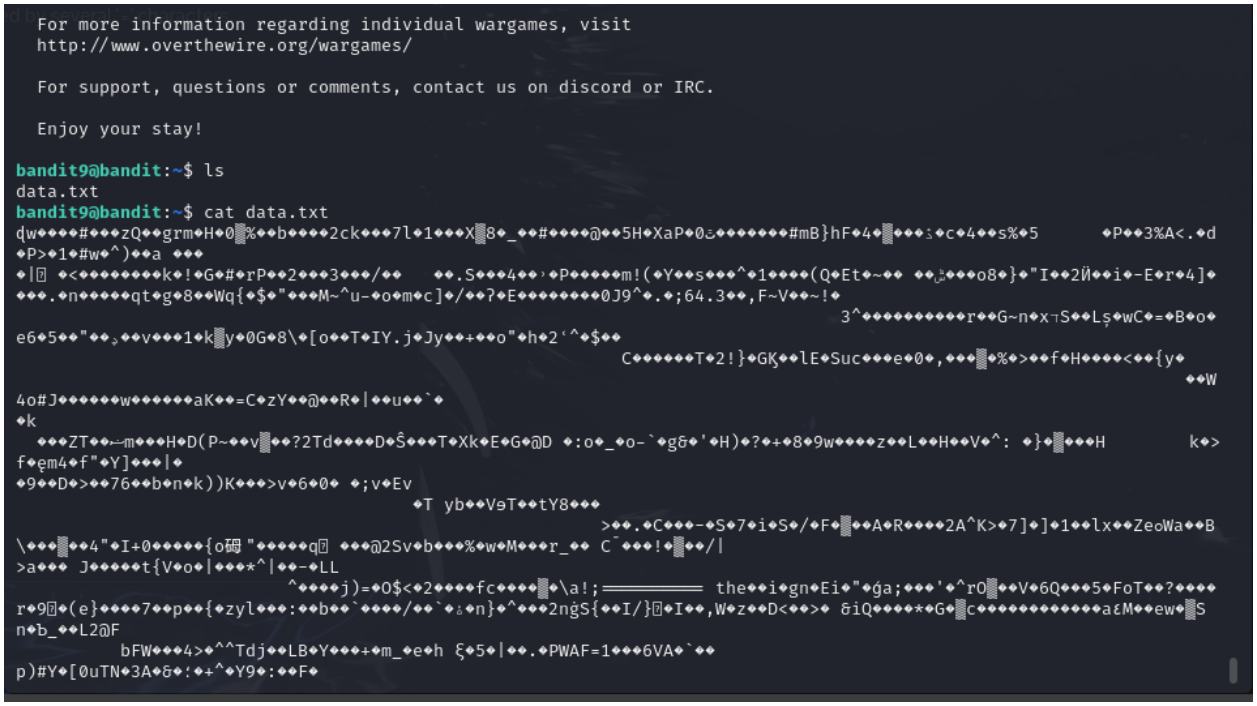
| *(pipe)* - The pipe (|) takes the output of the sort command and passes it as input to the next command (uniq -u)

uniq -u – gives the unique lines

Level 9 - 10

Password: FGUW5ilLVJrxX9kMYMmlN4MgbpfMiqey

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



The commands:

strings data.txt | grep '==='

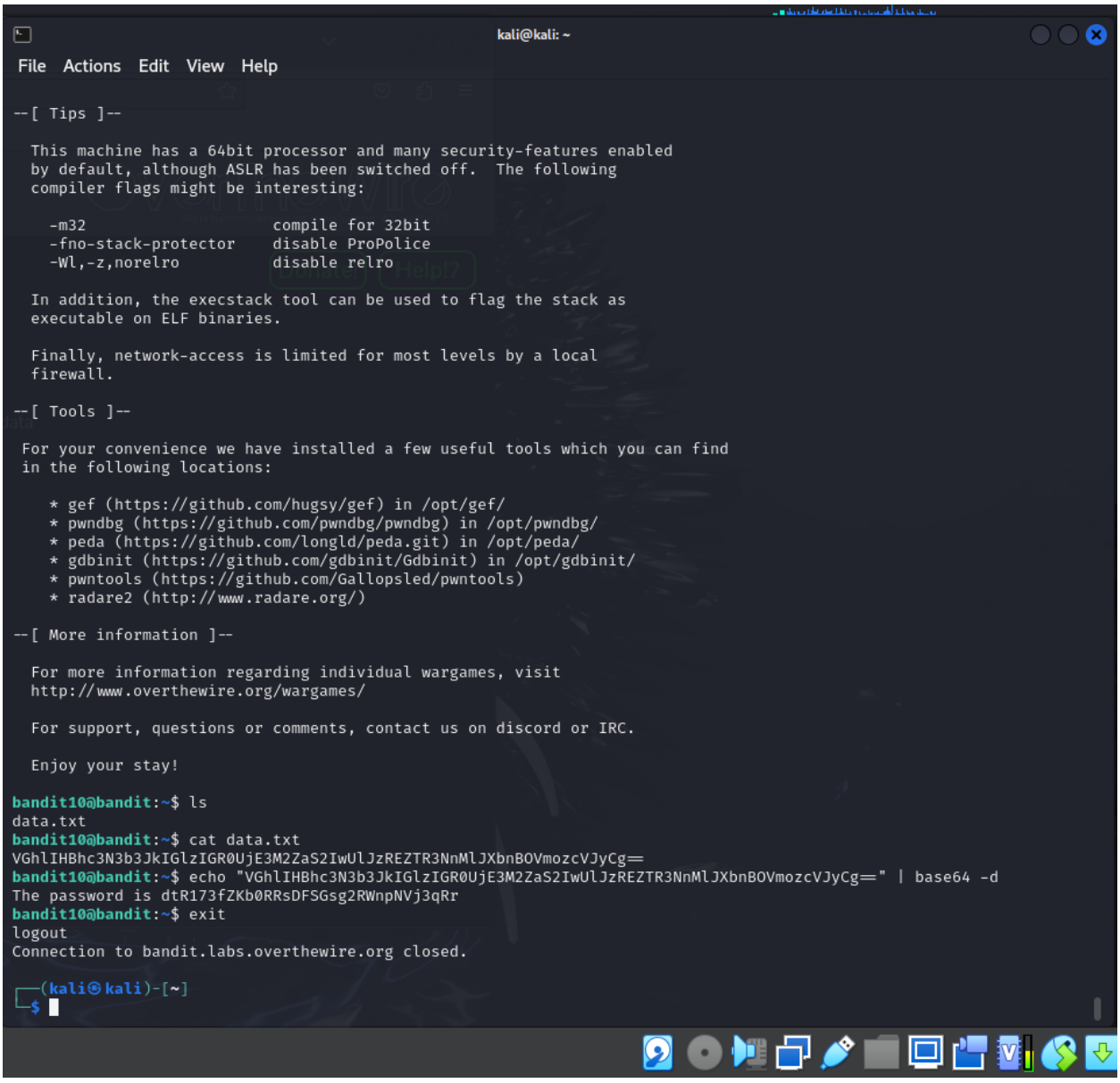
strings data.txt – extract human-readable strings (useful when dealing with binary)

grep === - command searches for lines containing the pattern '==='

Level 10 - 11

Password: dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr

- The password for the next level is stored in the file **data.txt**, which contains base64 encoded data.
- We must reverse base64 encoded data back to find the password.



The command:

echo "needed encoded string" | base64 -d

OR

base64 -d data.txt

- First get the base64 decoded string by opening the file.
- Then copy the code and using ‘echo’ command give the copied decoded string
- The *base64 -d* command in Linux is used to decode data that has been encoded in the Base64 encoding scheme.

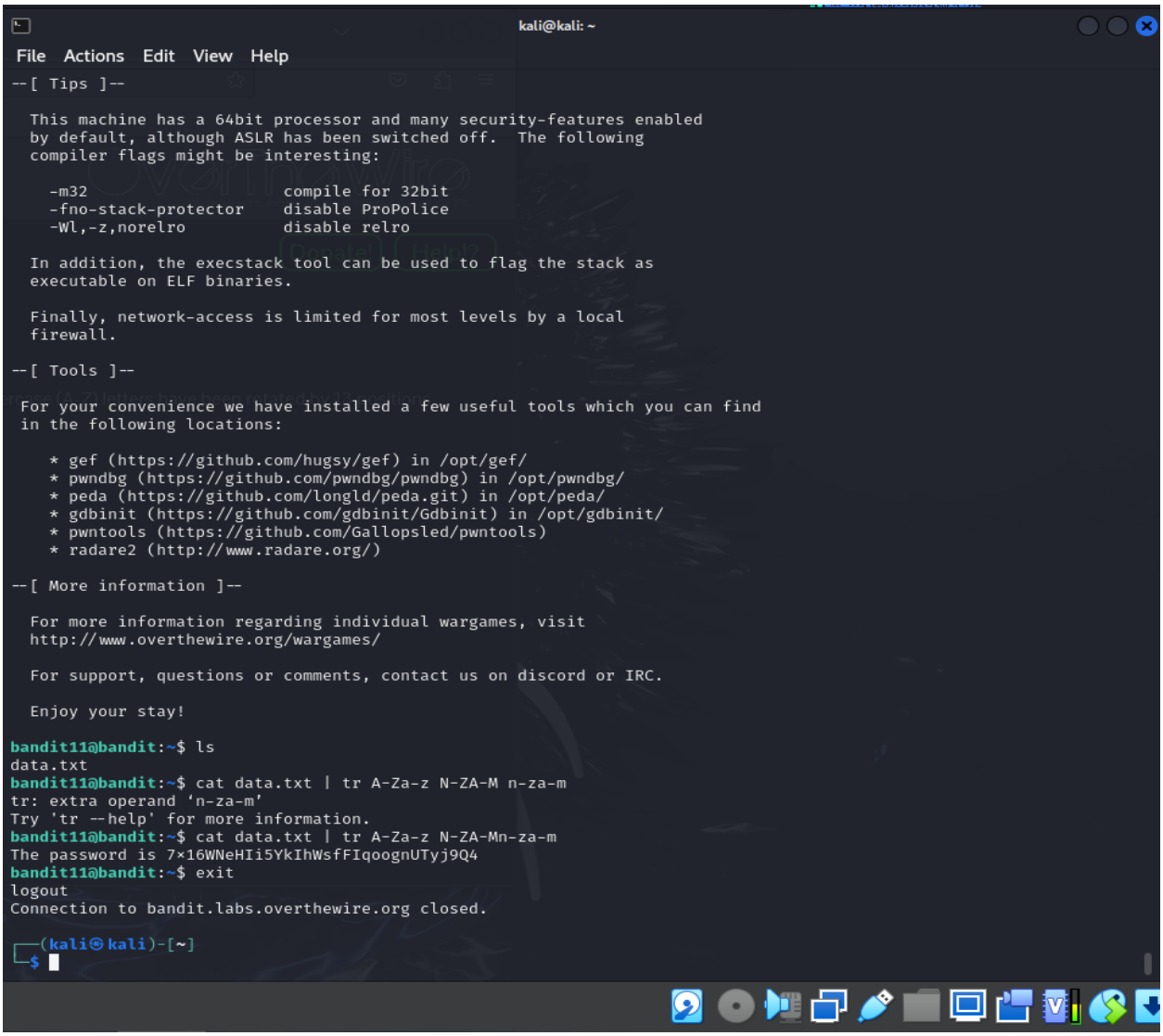
Level 11 - 12

Password: 7x16WNeHIi5YkIhWsfFIqoognUTyj9Q4

- 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.
- Here using ROT13 we are going to find the original message.
- **ROT13** is a simple substitution cipher used to encode and decode text by shifting each letter in the alphabet by 13 places. It is a specific case of the Caesar cipher, which shifts letters by a fixed number of places; ROT13 stands for "rotate by 13 places."

How ROT13 Works

- **Alphabet Shift**: In ROT13, each letter in the original text (plaintext) is replaced by the letter that is 13 positions later in the alphabet. For example:
    - A becomes N
    - B becomes O
    - C becomes P
    - ...
    - N becomes A
    - O becomes B
    - P becomes C
    - ...
    - Z becomes M
- **Bidirectional**: ROT13 is its own inverse, meaning applying ROT13 to a text twice will return the original text. For example, if you apply ROT13 to the string "HELLO", you'll get "URYYB". Applying ROT13 again to "URYYB" will return "HELLO".



The command:

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

The tr command is used to translate and delete characters in a string.

A-Z – represents letter capital from A-Z

a-z – represents letters simple from a-z

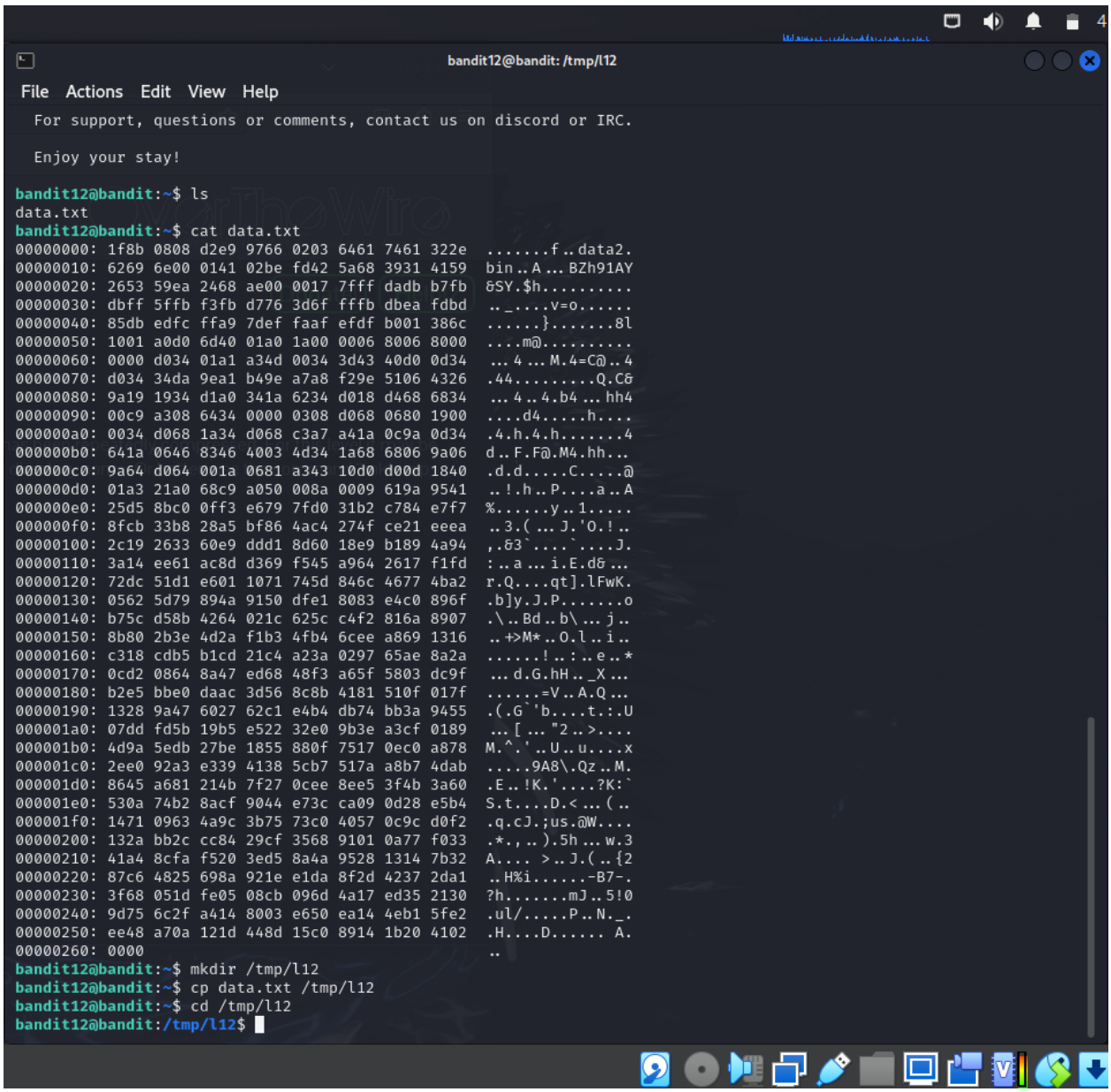
N-Z – represents letters from capital N-Z from 13 letters from alphabetical order according to ROT13

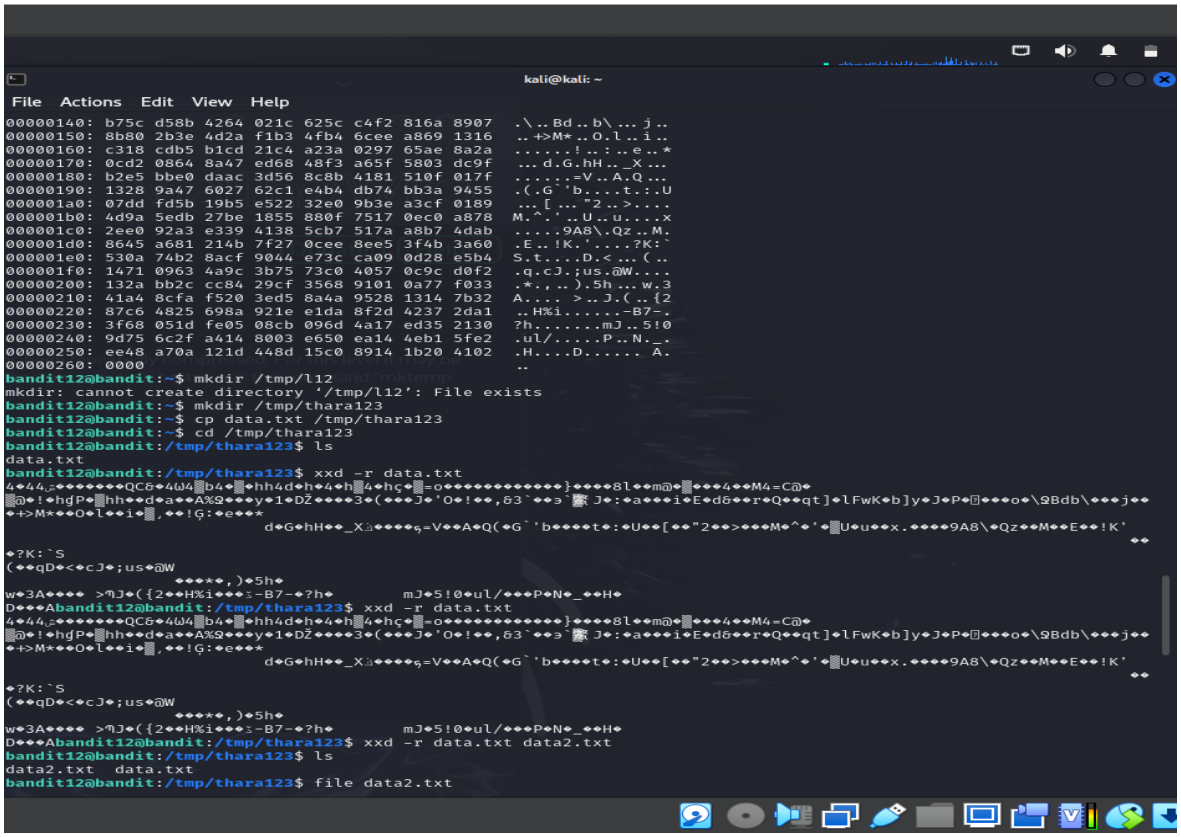
A-M – represents letter in capital form from A-Z first 13 letters

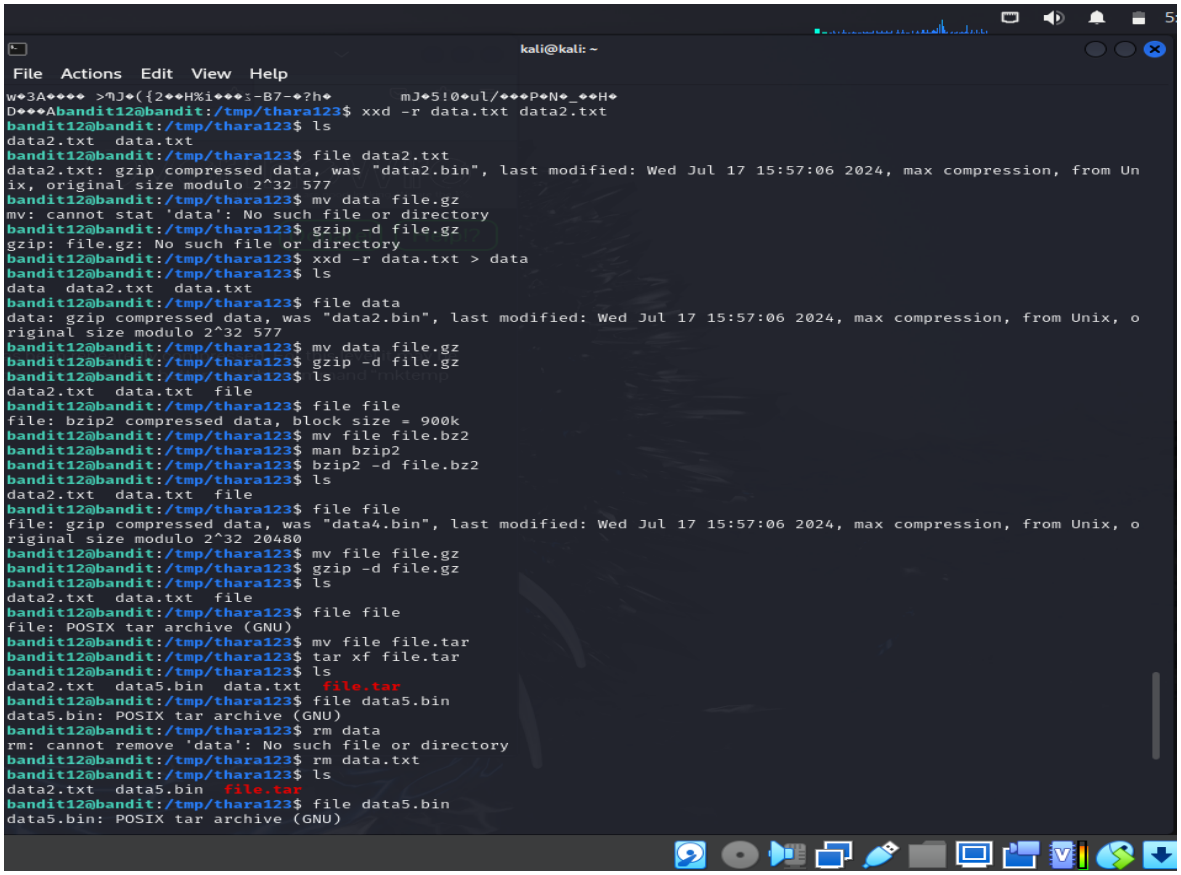
Level 12 - 13

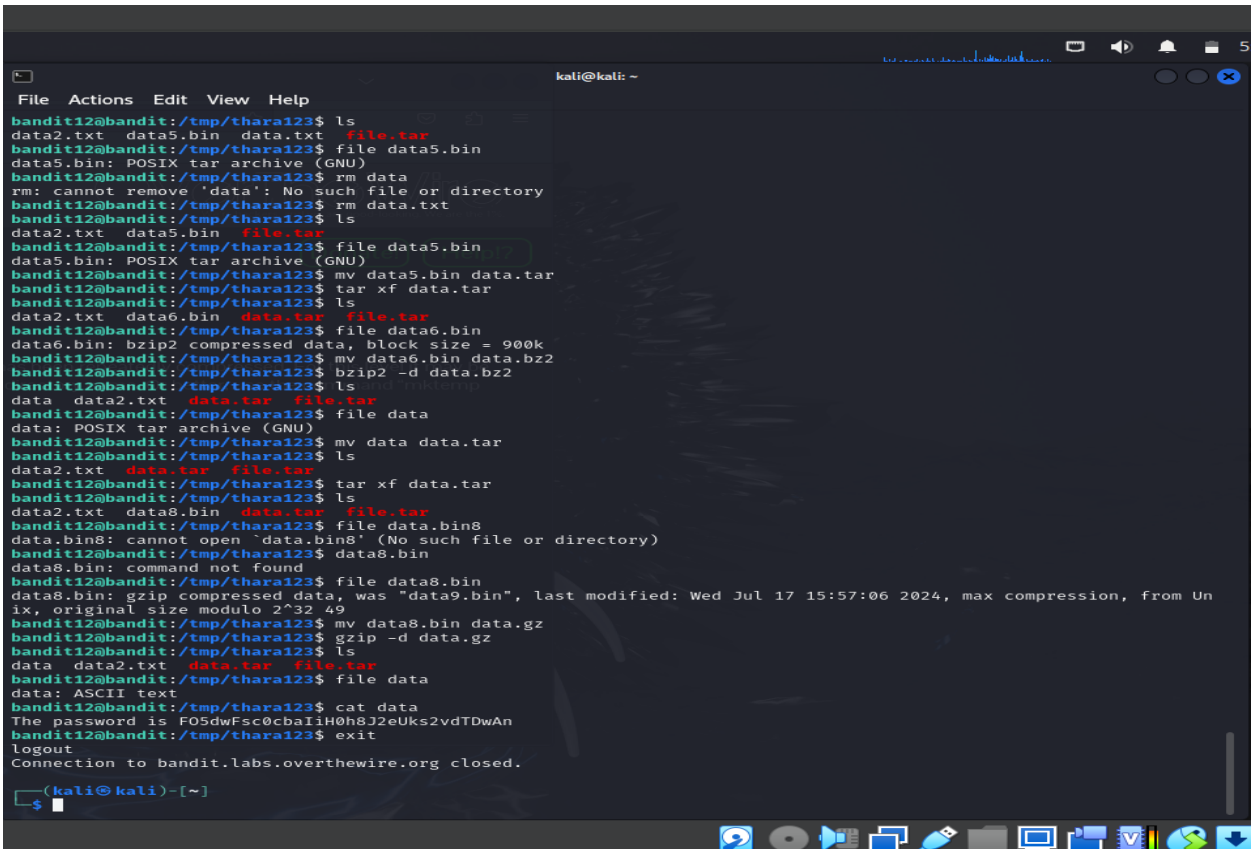
Password: FO5dwFsc0cbaIiH0h8J2eUks2vdTDwAn

- 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. Use mkdir with a hard to guess directory name. Or better, use the command “mktemp -d”. Then copy the datafile using cp, and rename it using mv (read the manpages!)




kali@kali: ~
File Actions Edit View Help
00000140: b75c d58b 4264 021c 625c c4f2 816a 8907 .\..Bd..b\...j..
00000150: 8b80 2b3e 4d2a f1b3 4fb4 6cee a869 1316 ..+>M*..O.l..i..
00000160: c318 cdb5 b1cd 21c4 a23a 0297 65ae 8a2a ......!..:..e..*
00000170: 0cd2 0864 8a47 ed68 48f3 a65f 5803 dc9f ...d.G.hH.._X...
00000180: b2e5 bbe0 daac 3d56 8c8b 4181 510f 017f ......=V..A.Q...
00000190: 1328 9a47 6027 62c1 e4b4 db74 bb3a 9455 .(.G`'b....t.:.U
000001a0: 07dd fd5b 19b5 e522 32e0 9b3e a3cf 0189 ...[...."2..>....
000001b0: 4d9a 5edb 27be 1855 880f 7517 0ec0 a878 M.^.'..U..u....x
000001c0: 2ee0 92a3 e339 4138 5cb7 517a a8b7 4dab .....9A8\.Qz..M.
000001d0: 8645 a681 214b 7f27 0cee 8ee5 3f4b 3a60 .E..!K.'....?K:`
000001e0: 530a 74b2 8acf 9044 e73c ca09 0d28 e5b4 S.t....D.<...(..
000001f0: 1471 0963 4a9c 3b75 73c0 4057 0c9c d0f2 .q.cJ.;us.@W....
00000200: 132a bb2c cc84 29cf 3568 9101 0a77 f033 .*.,..).5h...w.3
00000210: 41a4 8cfa f520 3ed5 8a4a 9528 1314 7b32 A.... >..J.(..{2
00000220: 87c6 4825 698a 921e e1da 8f2d 4237 2da1 ..H%i.....-B7-.
00000230: 3f68 051d fe05 08cb 096d 4a17 ed35 2130 ?h.......mJ..5!0
00000240: 9d75 6c2f a414 8003 e650 ea14 4eb1 5fe2 .ul/.....P..N._.
00000250: ee48 a70a 121d 448d 15c0 8914 1b20 4102 .H....D...... A.
00000260: 0000 ..
bandit12@bandit:~$ mkdir /tmp/l12
mkdir: cannot create directory '/tmp/l12': File exists
bandit12@bandit:~$ mkdir /tmp/thara123
bandit12@bandit:~$ cp data.txt /tmp/thara123
bandit12@bandit:~$ cd /tmp/thara123
bandit12@bandit:/tmp/thara123$ ls
data.txt
bandit12@bandit:/tmp/thara123$ xxd -r data.txt
bandit12@bandit:/tmp/thara123$ xxd -r data.txt data2.txt
bandit12@bandit:/tmp/thara123$ ls
data2.txt data.txt
bandit12@bandit:/tmp/thara123$ file data2.txt


kali@kali: ~
File Actions Edit View Help
bandit12@bandit:/tmp/thara123$ xxd -r data.txt data2.txt
bandit12@bandit:/tmp/thara123$ ls
data2.txt data.txt
bandit12@bandit:/tmp/thara123$ file data2.txt
data2.txt: gzip compressed data, was "data2.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32 577
bandit12@bandit:/tmp/thara123$ mv data file.gz
mv: cannot stat 'data': No such file or directory
bandit12@bandit:/tmp/thara123$ gzip -d file.gz
gzip: file.gz: No such file or directory
bandit12@bandit:/tmp/thara123$ xxd -r data.txt > data
bandit12@bandit:/tmp/thara123$ ls
data data2.txt data.txt
bandit12@bandit:/tmp/thara123$ file data
data: gzip compressed data, was "data2.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32 577
bandit12@bandit:/tmp/thara123$ mv data file.gz
bandit12@bandit:/tmp/thara123$ gzip -d file.gz
bandit12@bandit:/tmp/thara123$ ls
data2.txt data.txt file
bandit12@bandit:/tmp/thara123$ file file
file: bzip2 compressed data, block size = 900k
bandit12@bandit:/tmp/thara123$ mv file file.bz2
bandit12@bandit:/tmp/thara123$ man bzip2
bandit12@bandit:/tmp/thara123$ bzip2 -d file.bz2
bandit12@bandit:/tmp/thara123$ ls
data2.txt data.txt file
bandit12@bandit:/tmp/thara123$ file file
file: gzip compressed data, was "data4.bin", last modified: Wed Jul 17 15:57:06 2024, max compression, from Unix, original size modulo 2^32 20480
bandit12@bandit:/tmp/thara123$ mv file file.gz
bandit12@bandit:/tmp/thara123$ gzip -d file.gz
bandit12@bandit:/tmp/thara123$ ls
data2.txt data.txt file
bandit12@bandit:/tmp/thara123$ file file
file: POSIX tar archive (GNU)
bandit12@bandit:/tmp/thara123$ mv file file.tar
bandit12@bandit:/tmp/thara123$ tar xf file.tar
bandit12@bandit:/tmp/thara123$ ls
data2.txt data5.bin data.txt file.tar
bandit12@bandit:/tmp/thara123$ file data5.bin
data5.bin: POSIX tar archive (GNU)
bandit12@bandit:/tmp/thara123$ rm data
rm: cannot remove 'data': No such file or directory
bandit12@bandit:/tmp/thara123$ rm data.txt
bandit12@bandit:/tmp/thara123$ ls
data2.txt data5.bin file.tar
bandit12@bandit:/tmp/thara123$ file data5.bin
data5.bin: POSIX tar archive (GNU)



Hexdump:- is a built-in Linux utility to filter and display the contents of different files in hex, decimal, octal, or ASCII formats. It functions well as an inspection tool and can be used for data recovery. We can also use it to view the executable code of different programs.

The command:

/tmp – creating temporary directory

mkdir /tmp/l12 – make a new directory

cp data.txt /tmp/l12 – copy the file ‘data.txt’ to new directory

xxd -r data.txt – used convert hexdump back to binary file or reverse hexdump

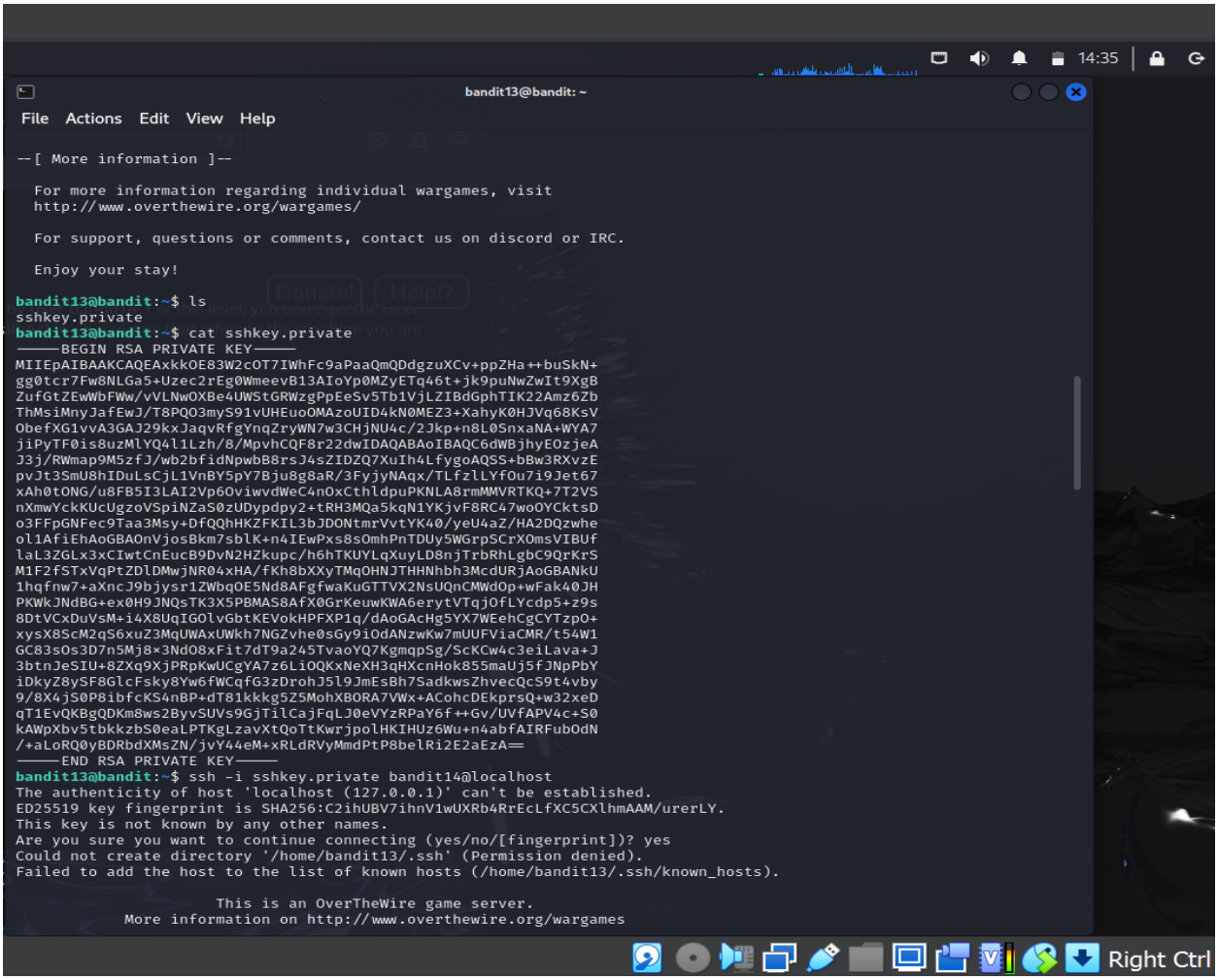
file data2.txt – used to find how it has been compressed

mv data file.gz – change the extension

gzip -d file.gz

Level 13 - 14

- ***Password: MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS***
- The password for the next level is stored in /etc/bandit_pass/bandit14 and can only be read by user bandit14.
- For this level we do not get a password, instead get a private key to log into the next level.

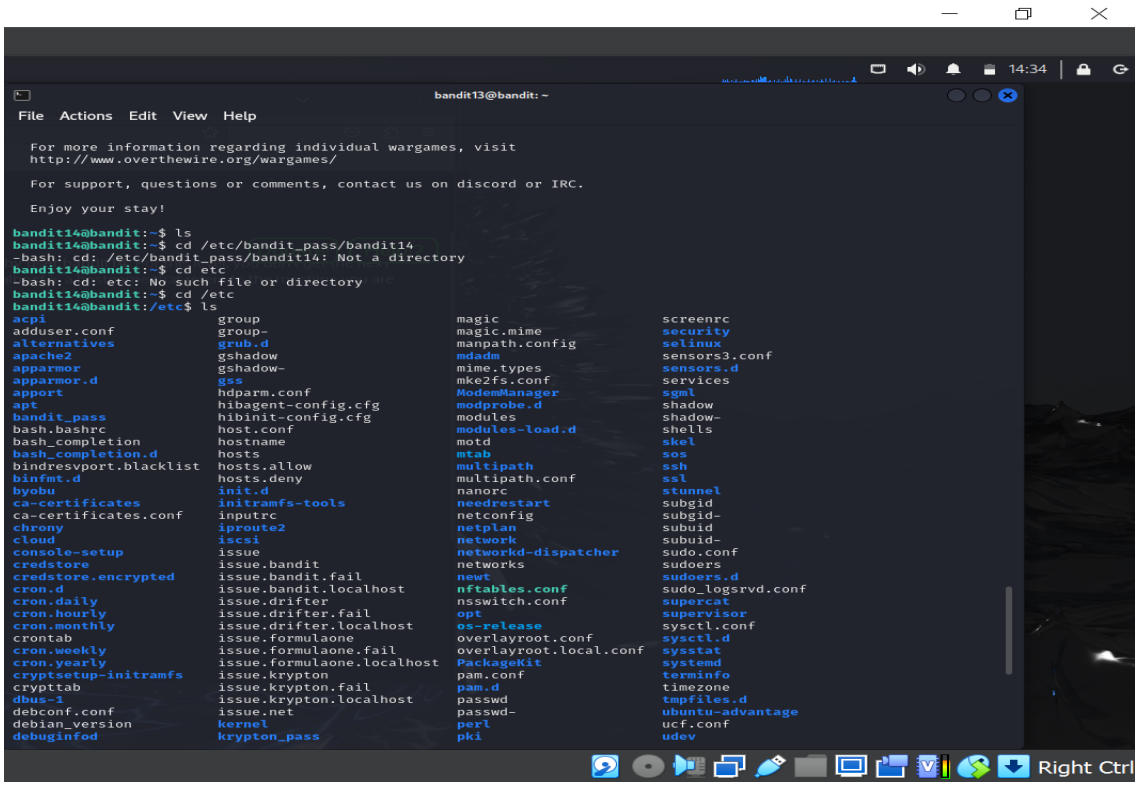


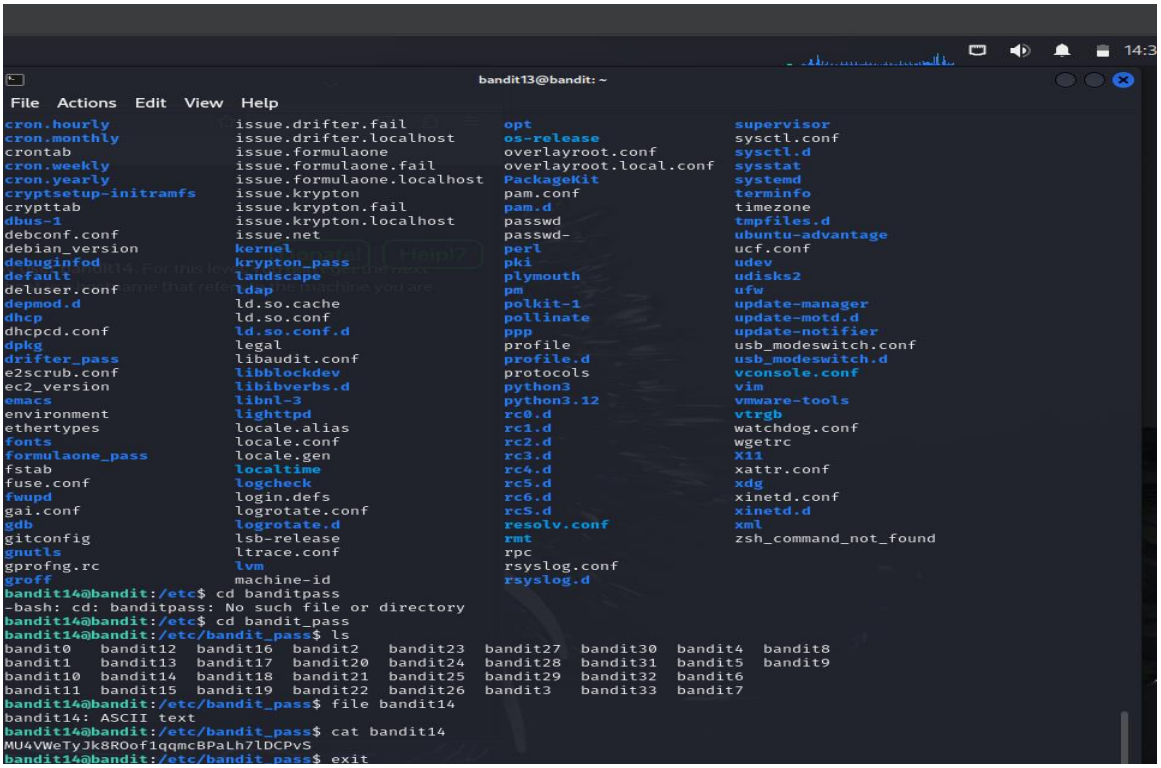
The command:

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

- Log into bandit14 without password using private key

- The password for the next level is stored in /etc/bandit_pass/bandit14 and can only be read by user bandit14.





The command:

cd /etc – opened the directory and list the files inside it

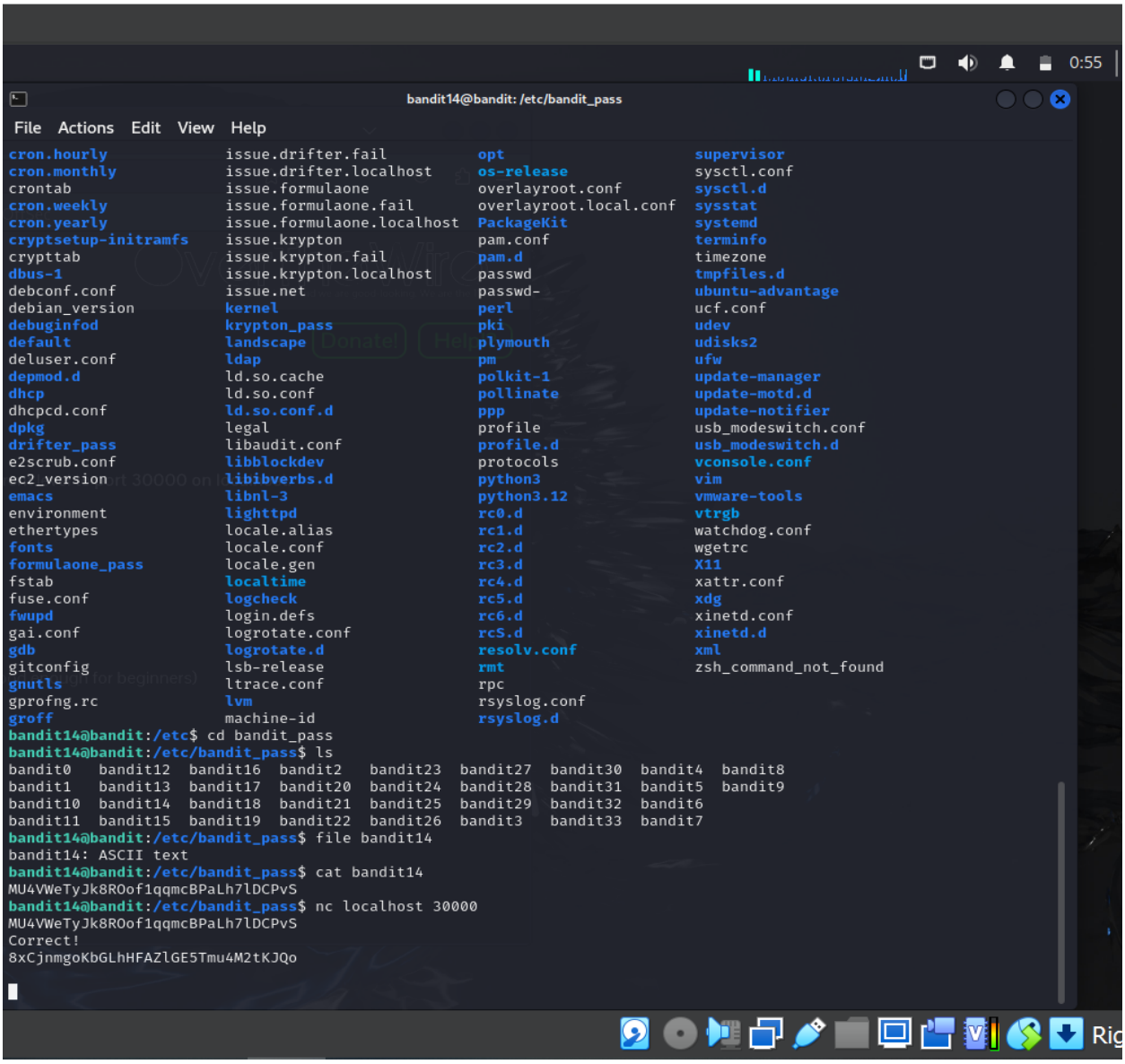
cd bandit_pass – open the directory and list the files

file bandit14 – checks for the type of file

Level 14 - 15

Password: 8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo

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



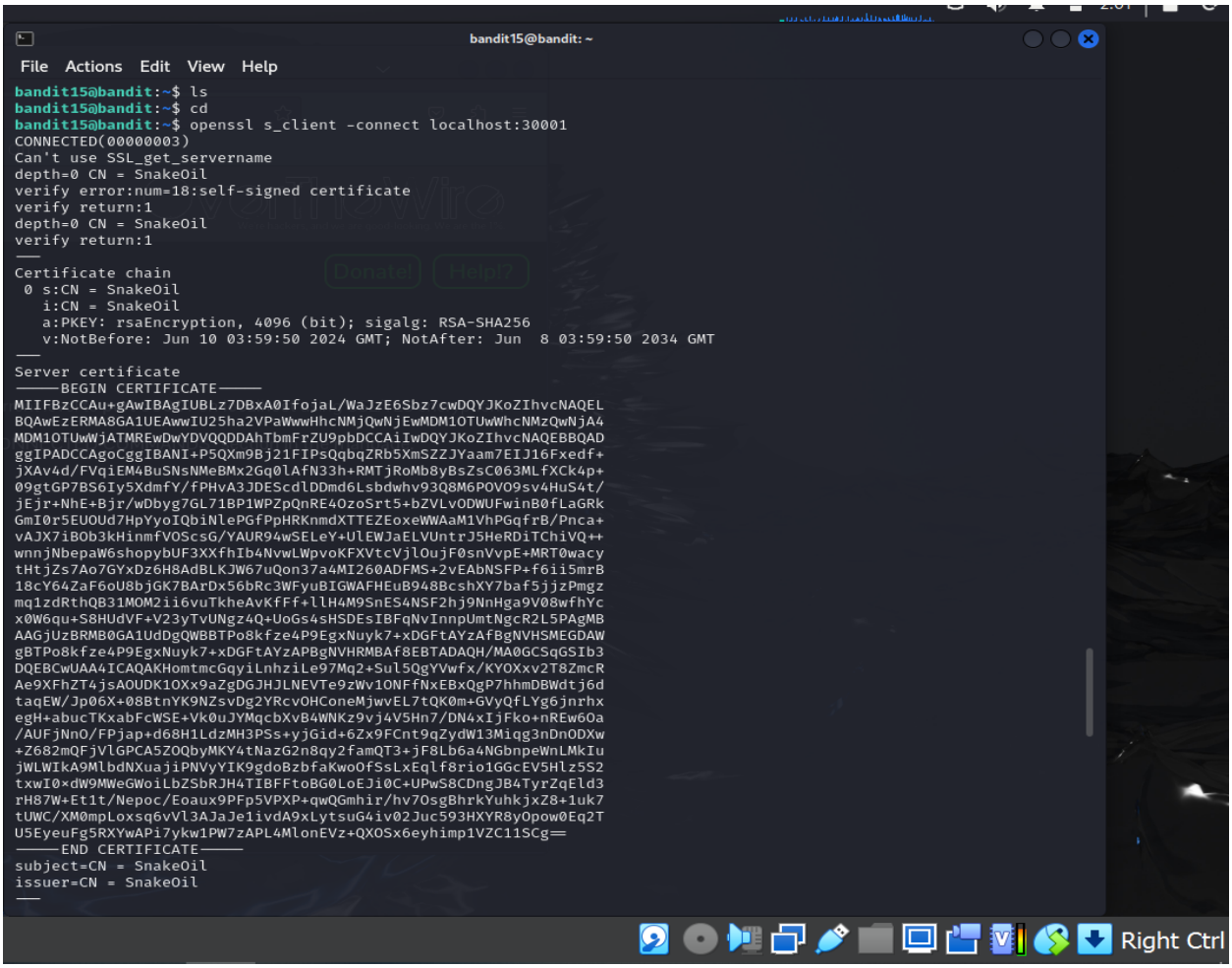
The command:

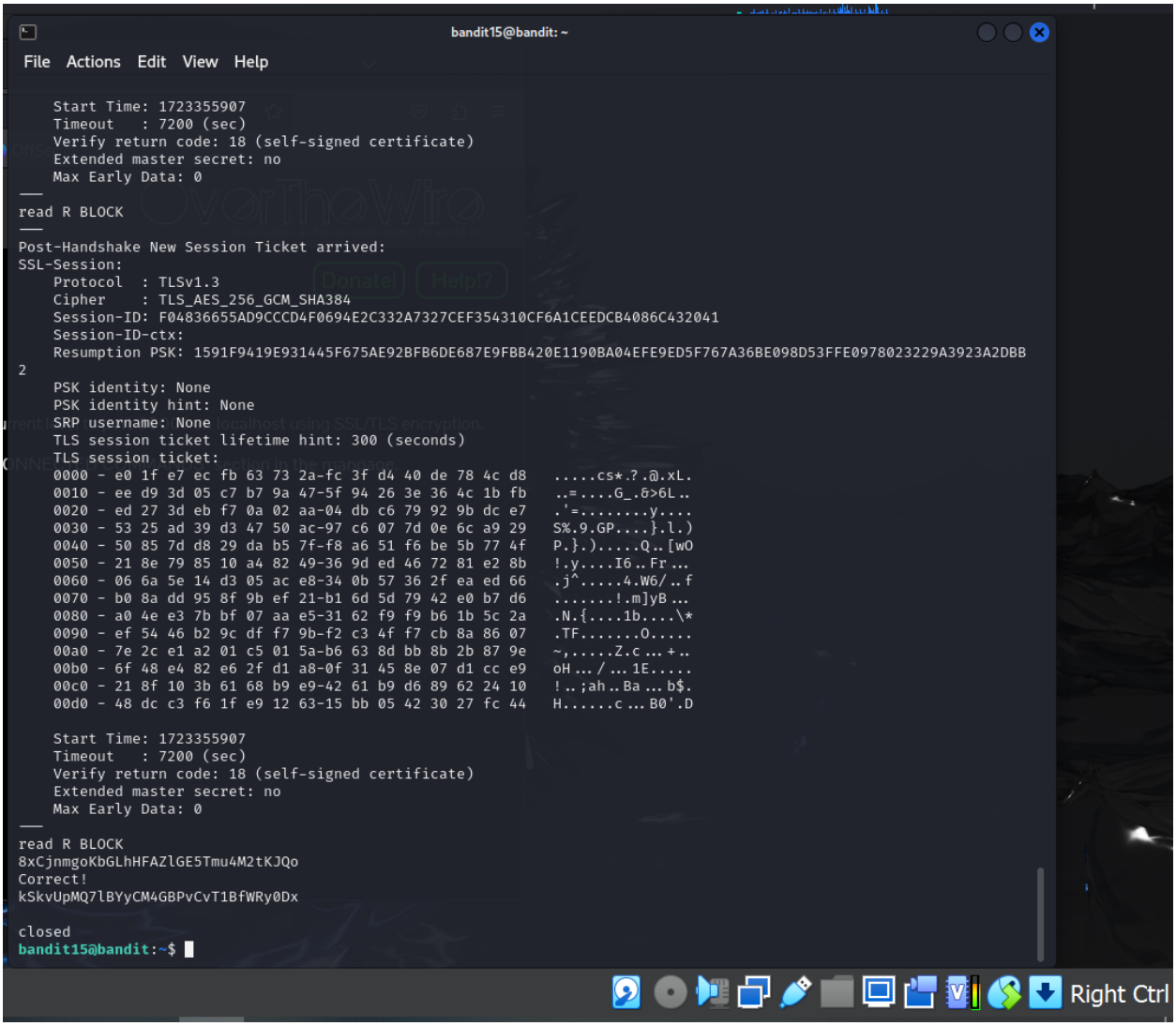
nc localhost 30000 - The nc localhost 30000 command in Linux is used to establish a network connection to a service running on the local machine (localhost) at port 30000 using the nc command, which stands for Netcat.

Level 15 - 16

Password: kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx

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





The command:

openssl s_client -connect localhost:30001

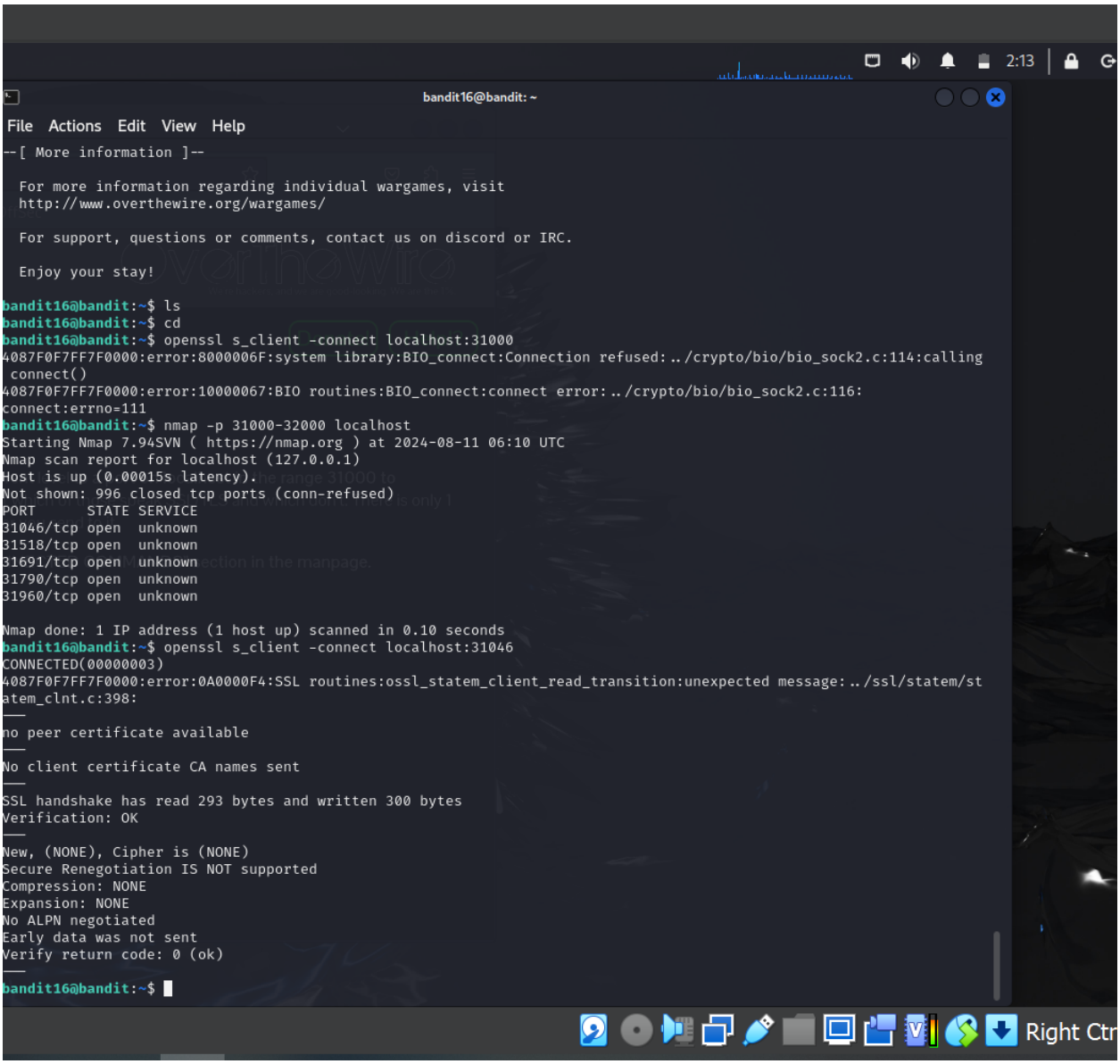
openssl s_client - This command initiates an SSL/TLS connection to port 30001 on localhost.

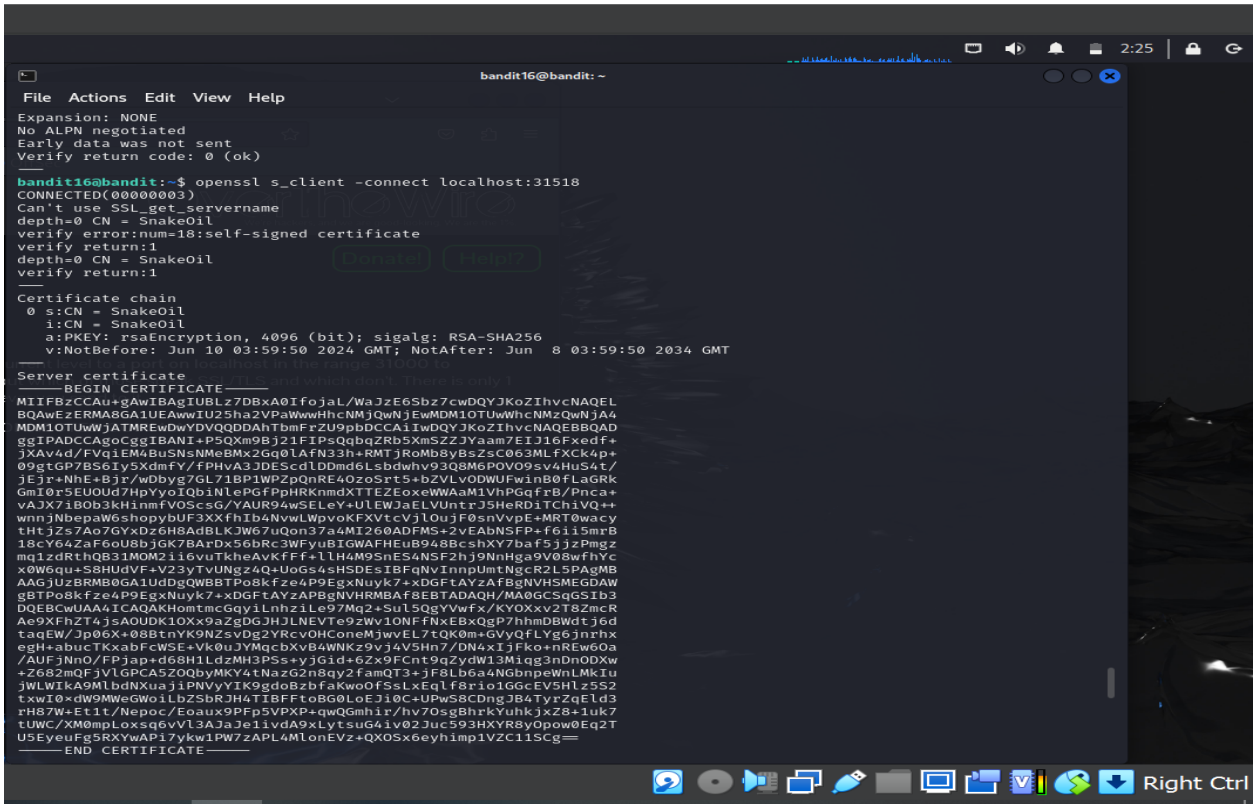
Then supply the current password and press enter for the new password.

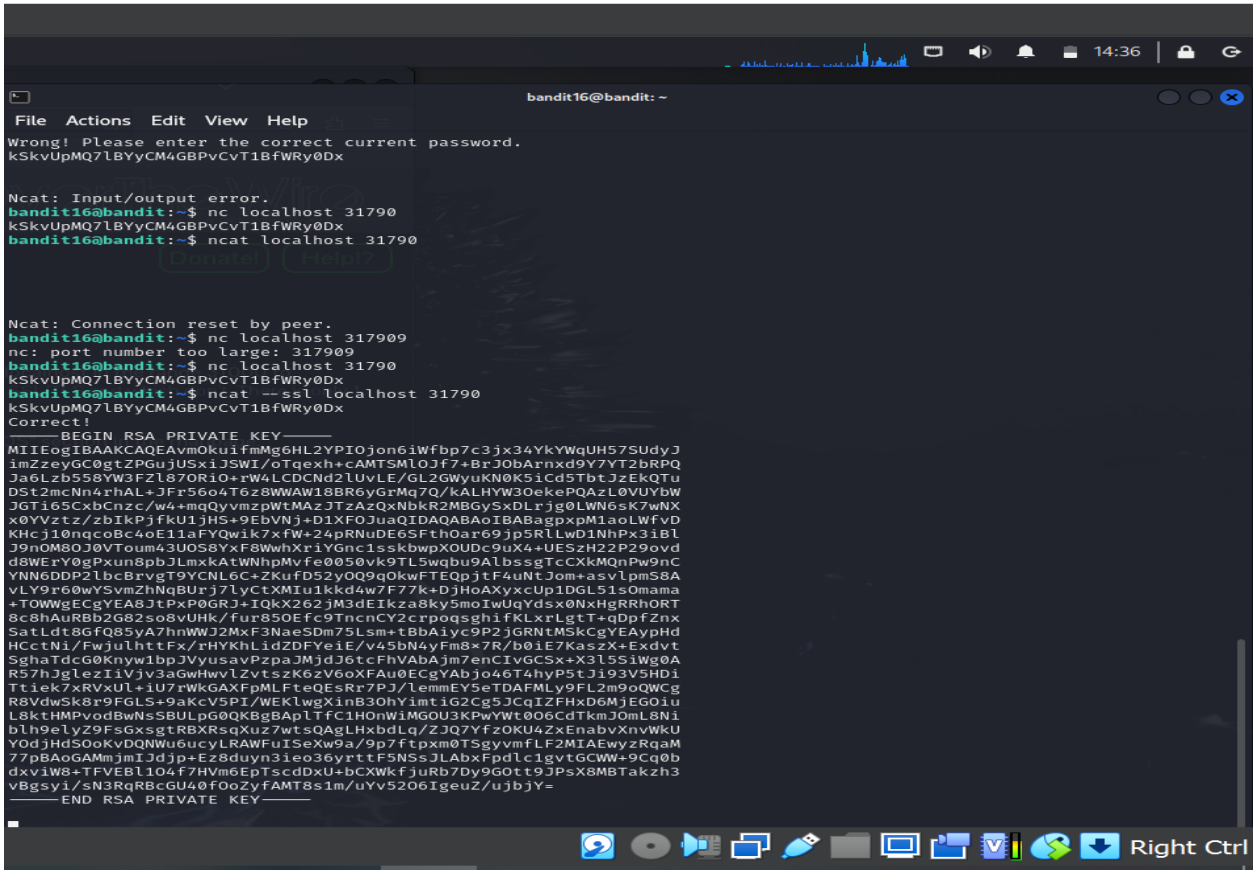
Level 16 – 17

Password: There is no password in this level, a private key is used access next level

- 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**.
- But first need to find out which of these ports have a server listening on them.
- Then find out which of those speak SSL/TLS and which don't.
- Instructions mention that there is only 1 server that will give the next credentials, the others will simply send back to you whatever you send to it.




2:25
bandit16@bandit: ~
File Actions Edit View Help
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
bandit16@bandit:~$ openssl s_client -connect localhost:31518
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 CN = SnakeOil
verify error:num=18:self-signed certificate
verify return:1
depth=0 CN = SnakeOil
verify return:1
Certificate chain
0 s:CN = SnakeOil
i:CN = SnakeOil
a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256
v:NotBefore: Jun 10 03:59:50 2024 GMT; NotAfter: Jun 8 03:59:50 2034 GMT
Server certificate
-----BEGIN CERTIFICATE-----
MIIFBzCCAu+gAwIBAgIUBLz7DBxA0IfojaL/WaJzE6Sbz7cwDQYJKoZIhvcNAQEL
BQAwEzERMA8GA1UEAwwIU25ha2VpAWwwHhcNMjQwNjEwMDM1OTUwWhcNMzQwNjA4
MDM1OTUwWjATMREwDwYDVQQDDAhTbmFrZU9pbDCCAiIwDQYJKoZIhvcNAQEBBQAD
ggIPADCCAgoCggIBANI+P5QXm9Bj21FIPsQqbqZRb5XmSZZJYaam7EIJ16Fxedf+
jXAv4d/FVqiEM4BuSNsNMeBMx26q0lAfN33h+RMTjRoMb8yBsZsC063MLfXCk4p+
09gtGP7BS6Iy5XdmfY/fPHvA3JDESCdlDDmd6LsbdWhv93Q8M6POVO9sv4HuS4t/
jEjr+NhE+Bjr/wDbyg7GL71BP1WPZpQnRE4OzoSrt5+bZVLvODWUFwinB0fLaGRk
GmI0r5EUOUd7HpYyoIQbiNlePGfPpHRKnmdXTTEZEoxeWWAaM1VhPGqfrB/Pnca+
vAJX7iBOb3kHinmfVOScsG/YAUR94wSELeY+UlEWJaELVUntrJ5HeRDiTChiVQ++
wnnjNbepaW6shopybUF3XXfhIb4NvwLWpvoKFXVtcVjlOujF0snVvpE+MRT0wacy
tHtjZs7Ao7GYxDz6H8AdBLKJW67uQon37a4MI260ADFMS+2vEAbNSFP+f6ii5mrB
18cY64ZaF6oU8bjGK7BArDx56bRc3WFyuBIGWAFHEuB948BcshXY7baf5jjzPmgz
mq1zdRthQB31MOM2ii6vuTkheAvKfFf+llH4M9SnES4NSF2hj9NnHga9V08wfhYc
x0W6qu+S8HUdVF+V23yTvUNgz4Q+UoGs4sHSDEsIBFqNvInnpUmtNgcR2L5PAgMB
AAGjUzBRMB0GA1UdDgQWBBTPo8kfze4P9EgxNuyk7+xDGFtAYzAfBgNVHSMEGDAW
gBTPo8kfze4P9EgxNuyk7+xDGFtAYzAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3
DQEBCwUAA4ICAQAKHomtmcGqyiLnhziLe97Mq2+Sul5QgYVwFx/KYOXxv2T8ZmcR
Ae9XFhZT4jsAOUDK1OXx9aZgDGJHJLNEVTe9zWv1ONFfNxEBxQgP7hhmDBWdtj6d
tagEW/Jp06X+08BtnYK9NZsvDg2YRcvOHConeMjwvEL7tQK0m+GVyQfLYg6jnrbx
egH+abucTKxabFcWSE+Vk0uJYMqcbXvB4WNKz9vj4V5Hn7/DN4xIjFko+nREw6Oa
/AUFjNnO/FPjap+d68H1LdzMH3PSs+yjGid+6Zx9FCnt9qZydW13Miqg3nDnODXw
+Z682mQFjVlGPCA5ZOQbyMKY4tNazG2n8qy2famQT3+jF8Lb6a4NGbnpeWnLMkIu
jWLWIkA9MlbdNXuajiPNVyYIK9gdoBzbfaKwoOfSsLxEqlf8rio1GGcEV5Hlz5S2
txwI0+dW9MWeGWoiLbZSbRJH4TIBFFtoBG0LoEJi0C+UPwS8CDngJB4TyrZqEld3
rH87W+Et1t/Nepoc/Eoaux9PFp5VPXP+qwQGmhir/hv7OsgBhrkYuhkjxZ8+1uk7
tUWC/XM0mpLoxsq6vVl3AJaJe1ivdA9xLytsuG4iv02Juc593HXYR8yOpow0Eq2T
U5EyeuFg5RXYwAPi7ykw1PW7zAPL4MlonEVz+QXOSx6eyhimp1VZC11SCg==
-----END CERTIFICATE-----
Right Ctrl


14:36
bandit16@bandit: ~
File Actions Edit View Help
Wrong! Please enter the correct current password.
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx
Ncat: Input/output error.
bandit16@bandit:~$ nc localhost 31790
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx
bandit16@bandit:~$ ncat localhost 31790
Ncat: Connection reset by peer.
bandit16@bandit:~$ nc localhost 317909
nc: port number too large: 317909
bandit16@bandit:~$ nc localhost 31790
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx
bandit16@bandit:~$ ncat --ssl localhost 31790
kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx
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/fur850Efc9TncnCY2crpoqsghifKLxrLgtT+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/uYv5206IgeuZ/ujbjY=
-----END RSA PRIVATE KEY-----
Right Ctrl

The commands:

nmap -p 31000-32000 localhost

– used to identify which ports have servers listening on them

- This command scans the ports from 31000 to 32000 on localhost and shows which ones are open

openssl s_client -connect localhost:31022(each open port has to be checked)

– testing each port which uses SSL/TLS

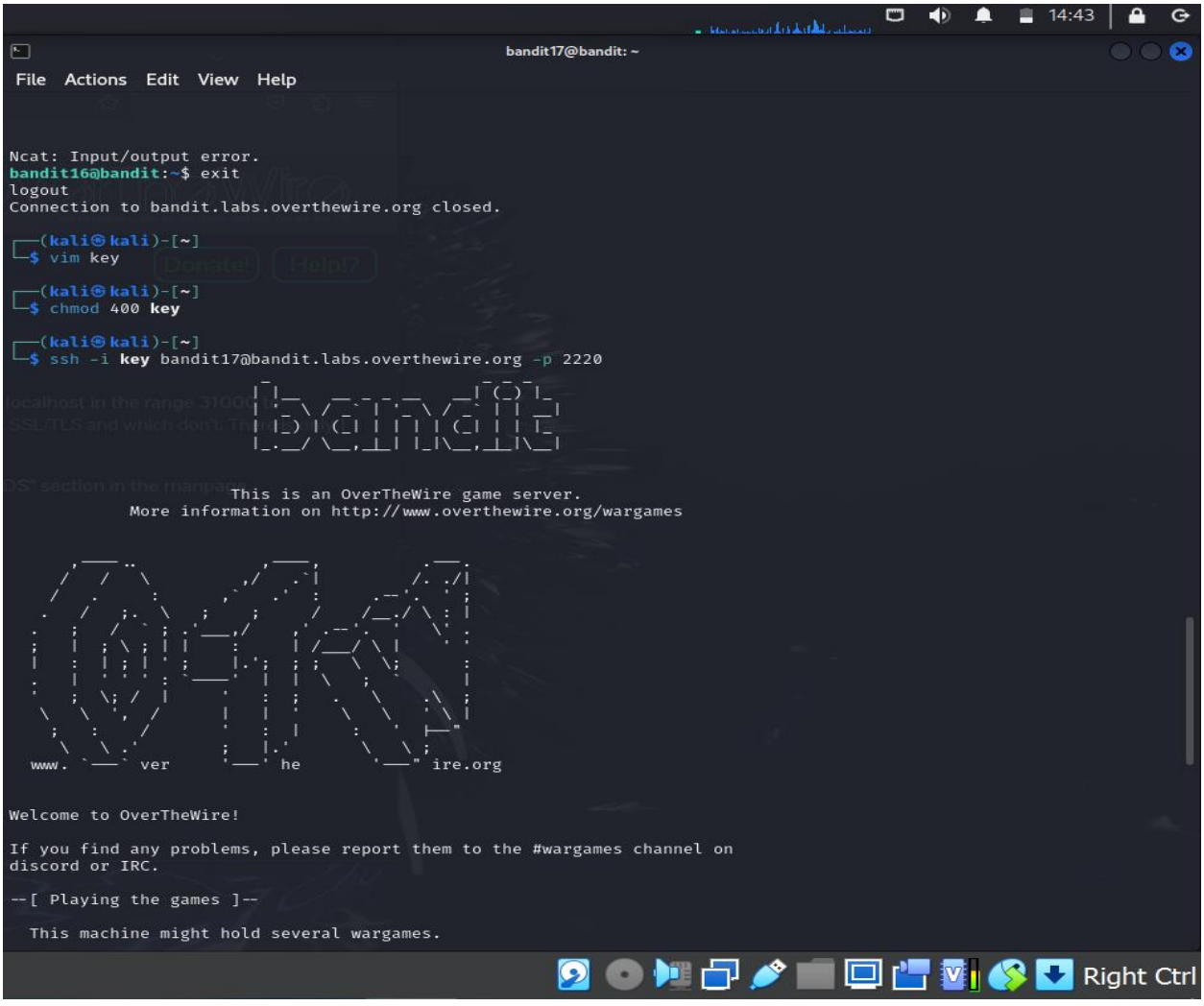
openssl s_client -connect localhost:31790

Finding the private key and save it in ‘key’ directory

Level 17 - 18

Password: x2gLTTjFwMOhQ8oWNbMN362QKxfRqGlO

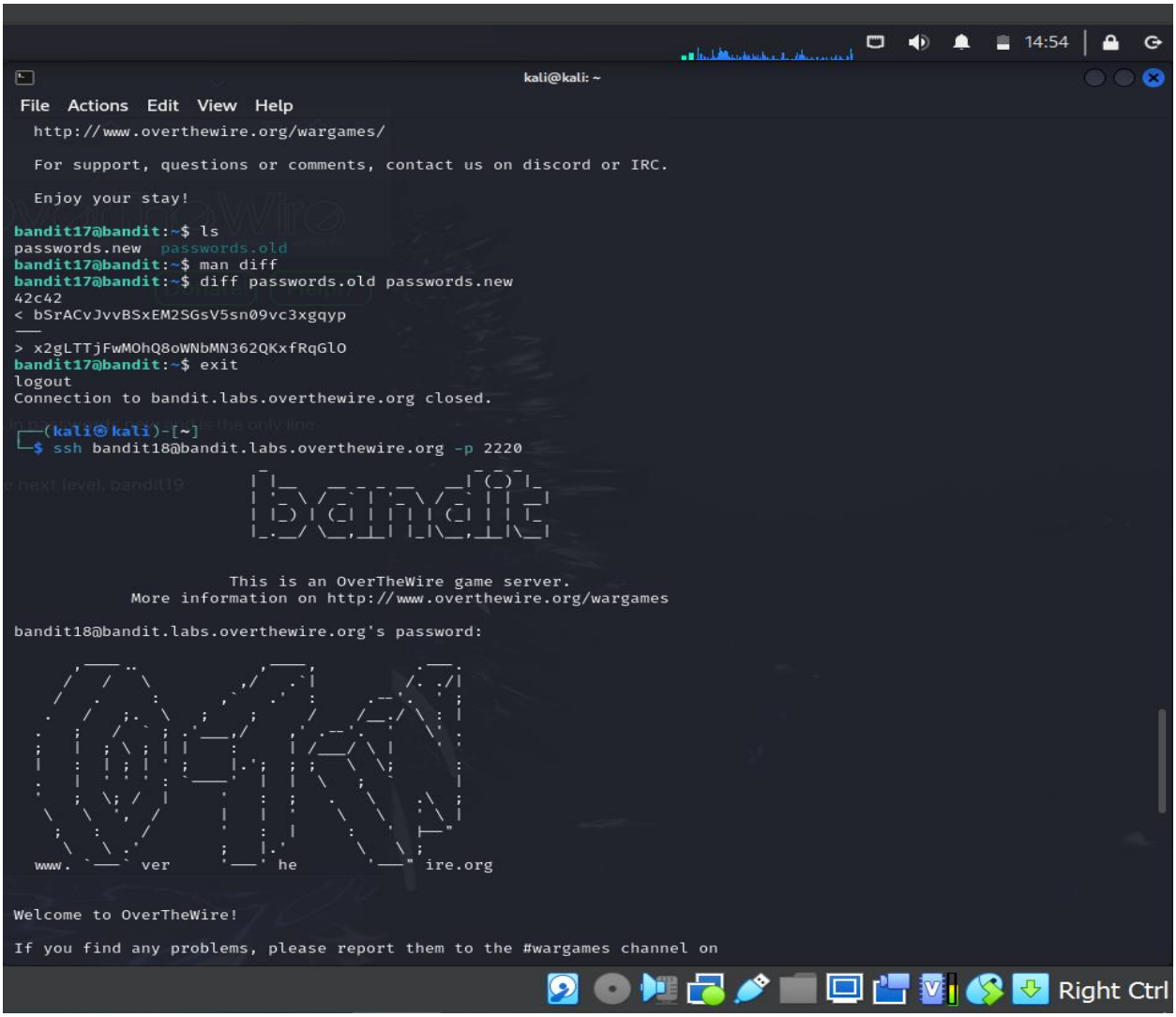
- There are 2 files in the home directory
    1. passwords.old
    2. password.new
- The password for the next level is in **passwords.new** and is the only line that has been changed between passwords.old and passwords.new



*****Log into the next level with the private key*****

chmod 400 key - used to change the permissions of a file named key so that it is readable only by the file owner

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



The command:

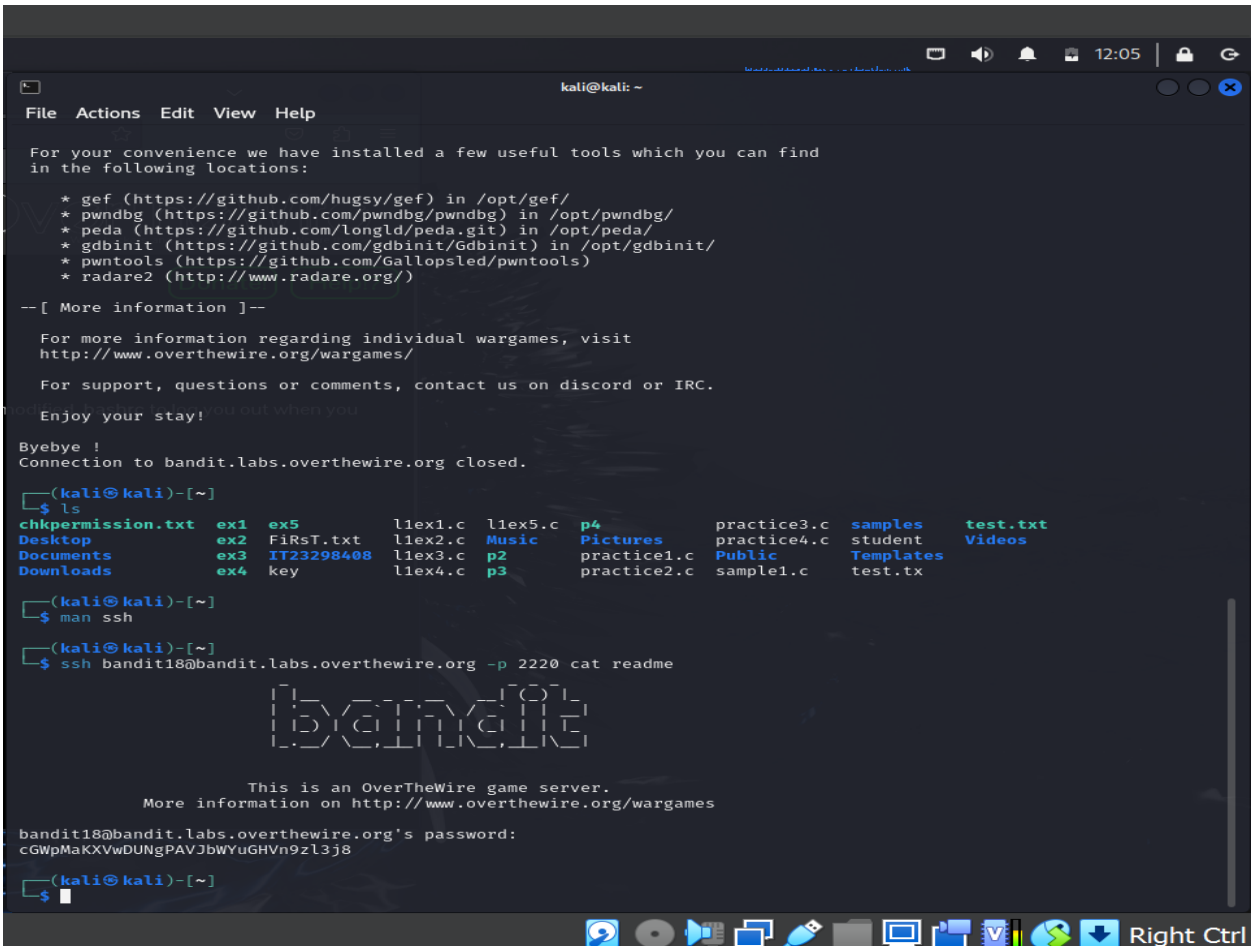
diff passwords.old passwords.new

compares two files (passwords.old and passwords.new) and shows the differences between them

Level 18 – 19

Password: cGWpMaKXVwDUNgPAVJbWYuGHVn9zl3j8

- The password for the next level is stored in a file **readme** in the home directory.
- But someone has modified .bashrc to log you when anyone tries to go with SSH



The command:

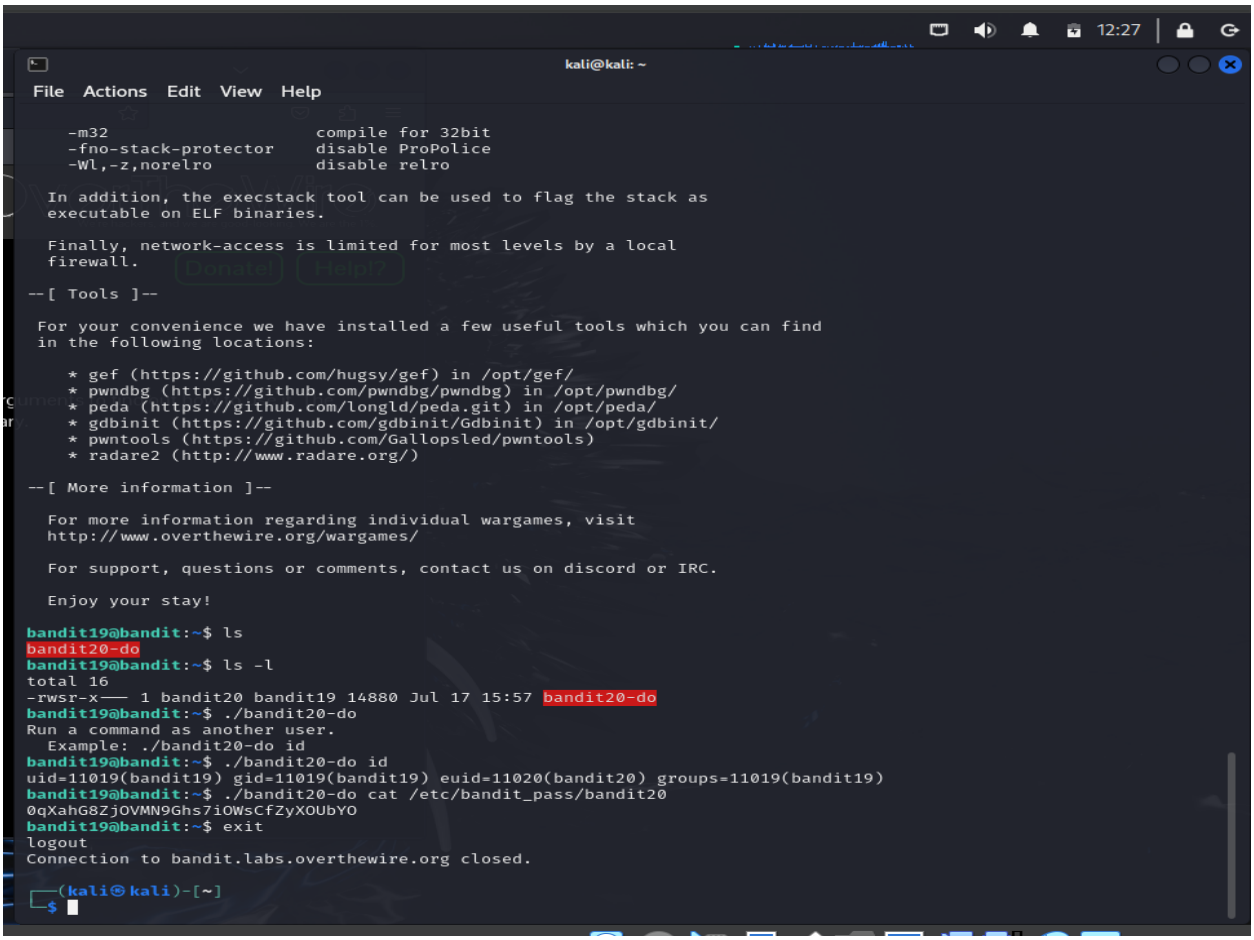
ssh bandit18@bandit.labs.overthewire.org -p 2220 cat readme

- We cannot log with regular ssh because the . *bashrc* has been modified to log out from it when someone try's to log in
- This command effectively bypasses the problem of .bashrc logging you out because it doesn't start an interactive shell. Instead, it directly runs the command and returns the result.

Level 19 -20

Passwords: 0qXahG8ZjOVMN9Ghs7iOWsCfZyXOUbYO

- To gain access to the next level, we should use the *setuid* binary in the home directory.
- And also we must execute it without arguments to find out how to use it.



The commands:

ls -l - List the files to identify the setuid binary. The name is highlighted due to the permissions

./**bandit20-do –** run this function to find out how to use it (without using any arguments)

./bandit20-do cat /etc/bandit_pass/bandit20 - Since the binary runs with elevated privileges, it can access files that you normally wouldn't be able to.