**CRYPTONITE**

**OVERTHEWIRE: BANDIT**

*TASK PHASE 1*

**WRITE-UP**

NAME: Kunal Komre

REGISTRATION NUMBER: 230932186

So before starting the Task Phase, I first learnt about the basics of Linux and its working and about cyber security and its importance.

As per the instructions I was required to install Linux OS such as Ubuntu. But since I am a beginner in these things, I used other alternatives to carry out the Task Phase.

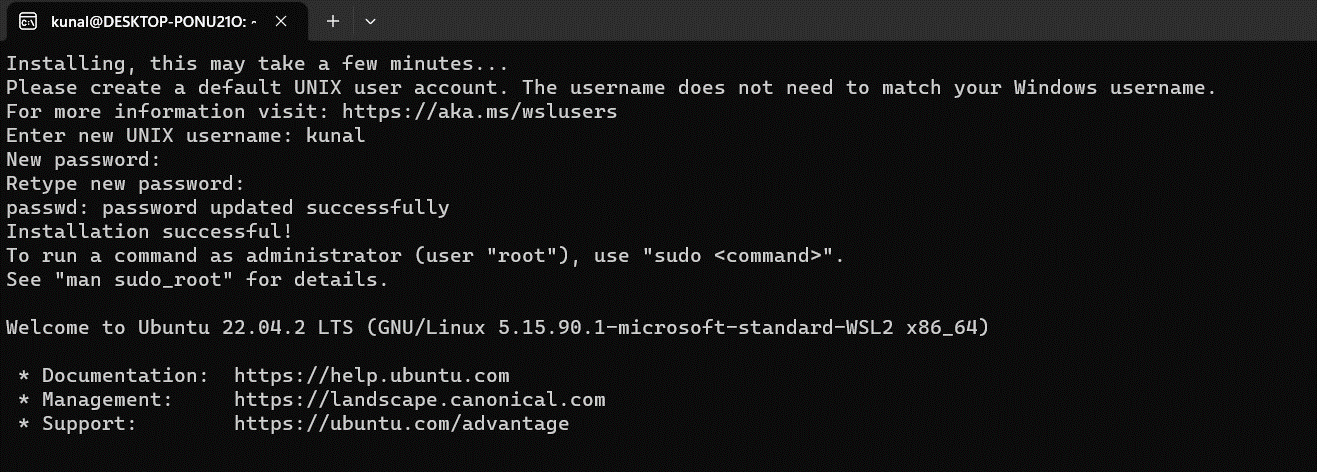
I installed WSL to play the game, since directly downloading Ubuntu or any Unix based OS seemed like a big task to me yet.

I referred to some YouTube videos and some articles about how to install WSL.

Here some videos I referred to:

* <https://youtu.be/YByZ_sOOWsQ?si=KmpMPM9vSB5le7tJ>
* <https://youtu.be/wjbbl0TTMeo?si=SrSREmn_6wRGlDBW>
* <https://learn.microsoft.com/en-us/windows/wsl/install>

Here is a screenshot of the me setting up the account and password.



*OVERTHEWIRE: BANDIT*

The Bandit Wargame aims to improve our cybersecurity and Linux skills by solving challenges from OverTheWire.

While solving all the levels, I was stuck at various problems.

I had to read the *Helpful Reading Material* and the various new commands and their purposes that were used to solve the level. Many commands such as *ls, cat, sort, grep, uniq, etc..* were used to go to the next level. Whenever I got stuck, I had to refer to many Google to learn about a new command that was further used to solve the level.

Some sites that I used are:

* <https://linuxhint.com/>
* <https://www.geeksforgeeks.org/>
* https://www.youtube.com/ (to learn about certain commands)
* https://github.com/

I have mentioned the commands along with their definition that I used. While playing, I got the idea of Linux and its basics and how directories work. I also learnt about the syntax and commands used to correctly show the results.

I have written about the I passed each level, along with the commands used and some mistakes I made along the way.

***LEVEL 0***

The game starts at LEVEL 0. The goal of this level is for us to log into the game using SSH.

SSH, also known as Secure Shell or Secure Socket Shell, is a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network. SSH also refers to the suite of utilities that implement the SSH protocol.

I used the *ssh* command to log into the network. The port, username and password were already given. I had to learn the format with which you can enter the server. As seen in the image, I made mistakes before correcting it up and entering the right way.



One new thing that I learned about Linux is that the password typed is not shown on the screen. It is a better alternative because it seems more secure.

After entering the password, I was logged in the server and entered LEVEL 1

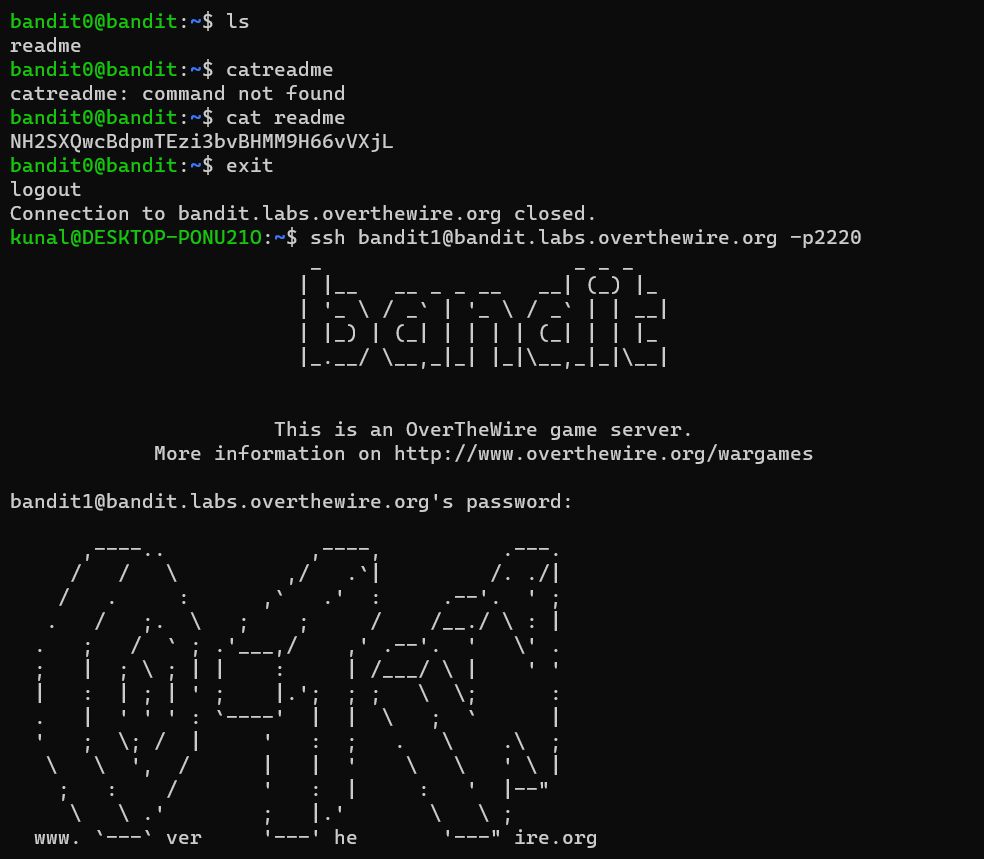
***LEVEL 1***

***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.

For this level I had to learn about different commands operated by Linux.

*COMMANDS USED:*

* “ls” – used to list the contents from the directory of the server
* “cat” – used to concatenate files and print on the standard output



After using the *ls* command, the list directory contents were shown. It only consisted of one file named *readme*. Then I used the *cat* command to concatenate files and print on the standard output. The password to log on the next level was shown.

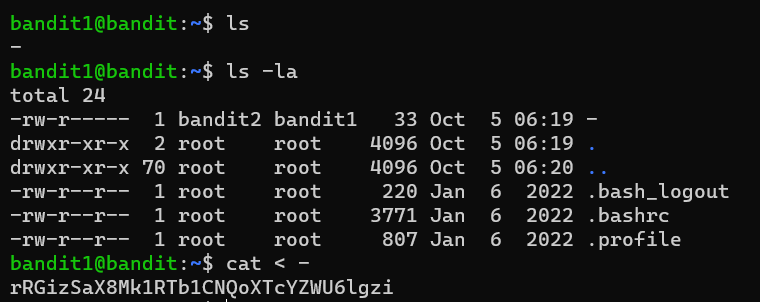
***LEVEL 2***

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

*COMMANDS USED:*

* “’ls -la” - To get a full list of hidden files, type **ls -la** in your terminal. The output displays information about the user, size of the file, and date and time of modification.
* “cat” - used to concatenate files and print on the standard output.

The directory name is “-“ which you cannot type directly after *cat* because it gives error as “-“ is also used for another command. So to concatenate the file we need to use cat < - and cat ./- command to get the password.



Using “cat < -“ we were able to read the “-“ file and the password was shown.

Now we can move onto the next level

***LEVEL 3***

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

*COMMANDS USED:*

* *“ls -alps”* ls -alps is the same as ls -a -l -p -s

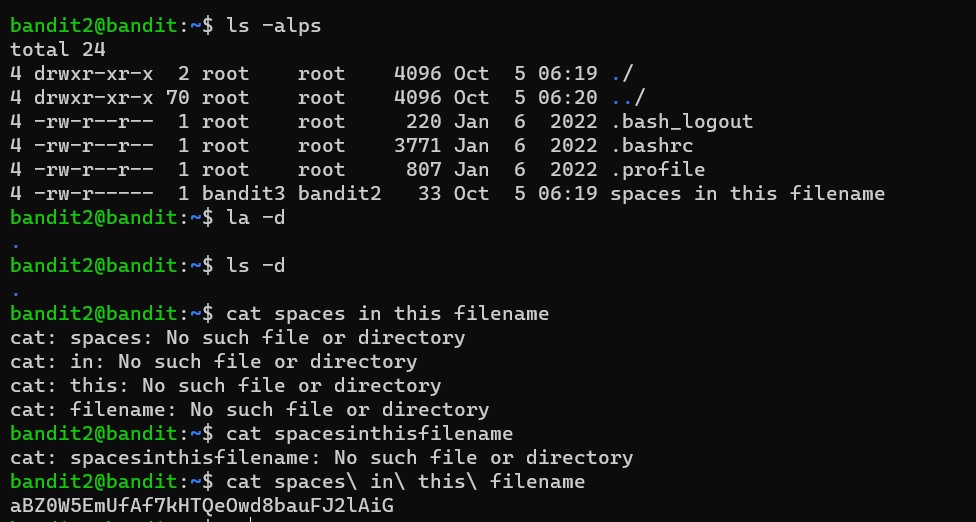
-a: All files are being presented (even if they start if a dot)

-l: The files are being presented in a listing format

-p: A slash (/) is being appended to all directories

-s: The size of each file is being shown

* *“cat”*



Now using the cat command and entering the name of the file produced error (as shown in the picture). To avoid error, I used “\”.

Using “\” instead of spaces removed the error and we were able to see the password in it.

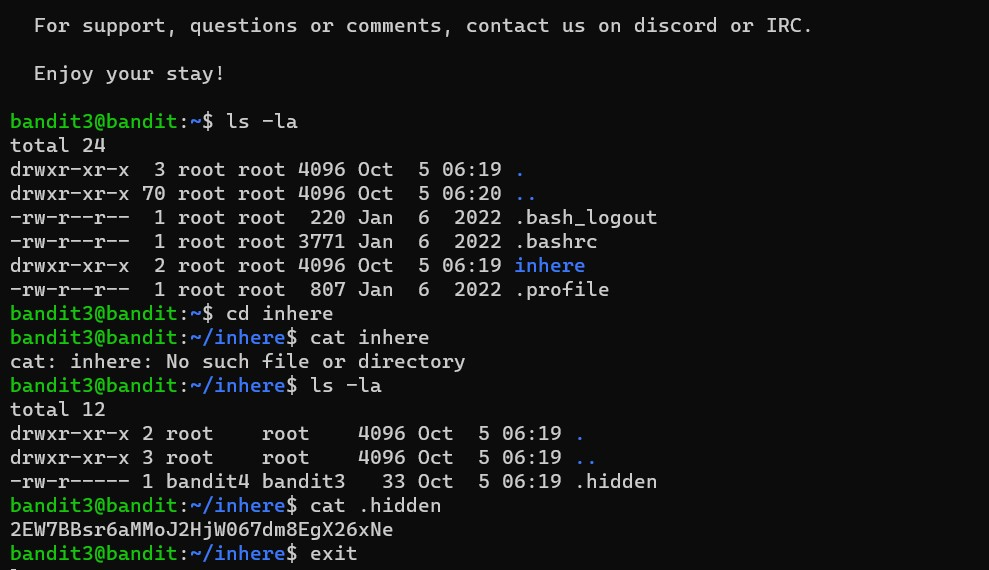
Now we move on to the next level.

***LEVEL 4***

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

*COMMANDS USED:*

* “ls -la”
* “cat”
* “cd” – It is used to change the working directory

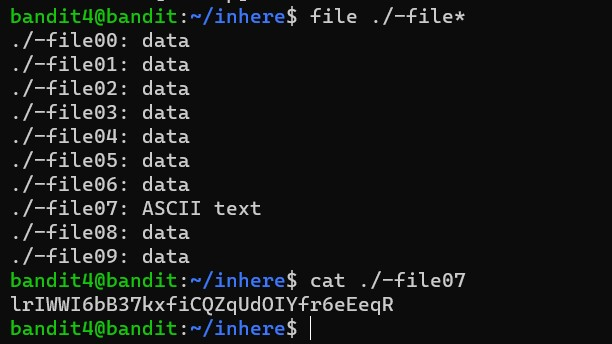


To read the hidden file, I used the *cd* command. It is used to change the working directory. It allowed me to change my directory to the file *inhere*. After that I tried to use the *cat* command to read the files of *inhere* which was wrong since we are already in the *inhere* directory. I again used *cat* to read the *.hidden* directory. The password was shown next.

***LEVEL 5***

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

*COMMANDS USED:*

* “file”: determine file type
* “cat”
* “ls” 

So for this level, I used *ls*  to show the list directory, but it showed the similar results and I was not able to guess which file contained the password. So I used the *file* command along with *\** command to display all files types. It showed that one file contained ASCII text. So I used *cat* to display the password of that directory.

Now onto the next level.

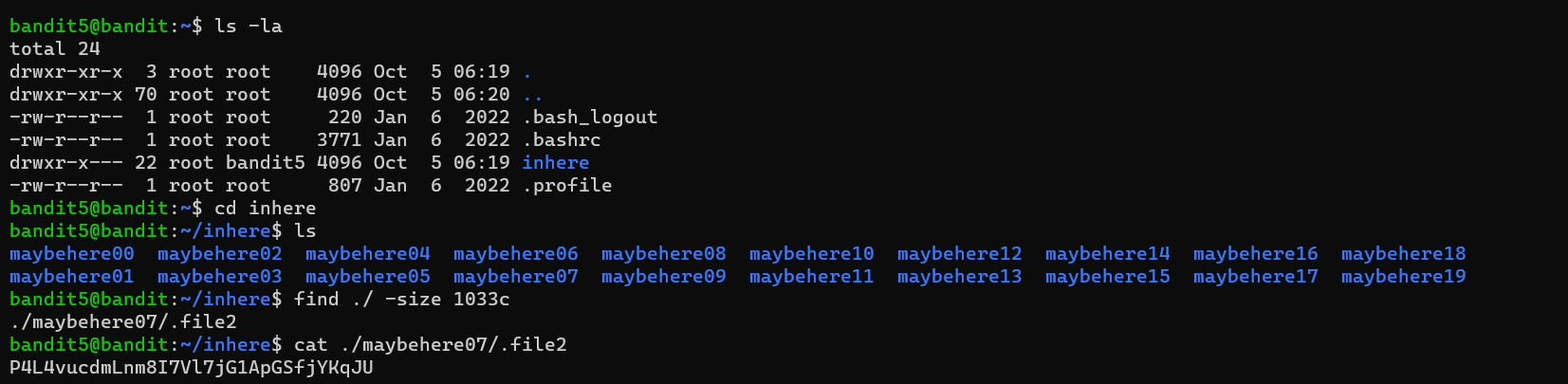
***LEVEL 6***

***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

*COMMANDS USED:*

* “ls”
* “cat”
* “find -size”
* “cd”



In this level, I used *ls* to display files. Then used *cd* to change directory to *inhere*. The files in this directory were of same types and we had to find in which file it contained the password. In the question it was given that the file is of size 1033 bytes/ 1033c (bytes written as c here). I then used the *find* command to find the file and also specified its size (./ -size 1033c). After specifying the size only, the password was shown.

I could have also specified that the file was not executable and was in ASCII text/ human readable format to get the same result.

Now onto the next level.

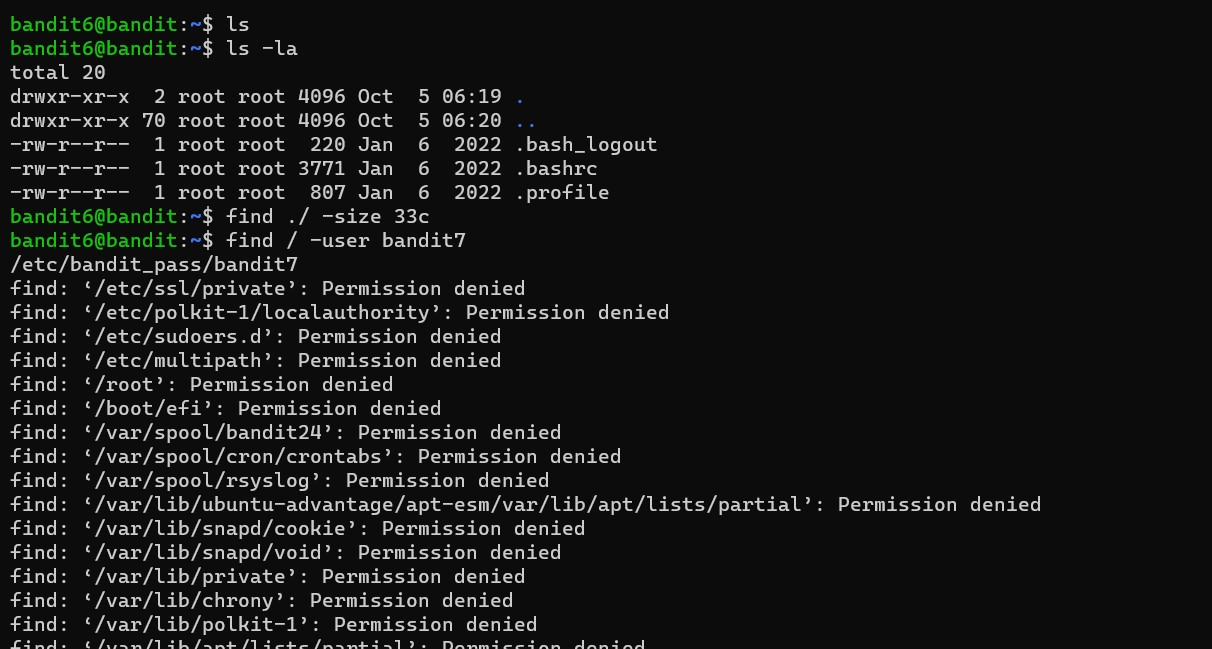
***LEVEL 7***

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

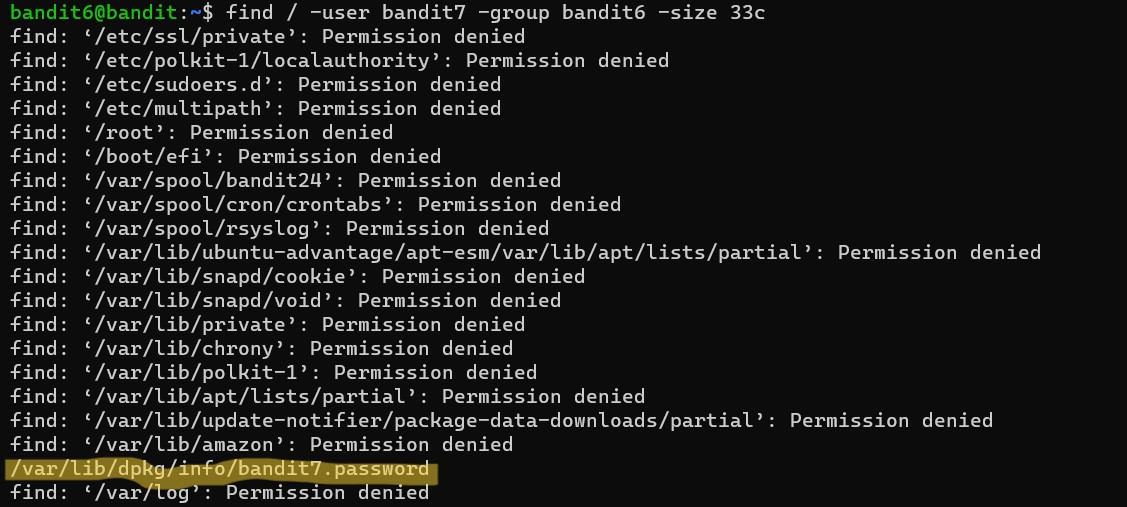
* owned by user bandit7
* owned by group bandit6
* 33 bytes in size

*COMMANDS USED:*

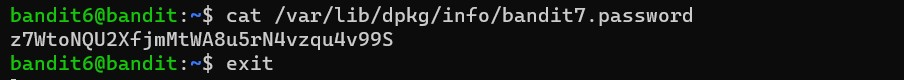
* “ls”
* “find ./size”



On using *ls* a lot of directories are shown. I tried to find the file with password using *find* *./size* and the *user* command but failed.



I used all the information given in the question to find the specified file( user, group, size). All of them showed a lot of data. I then studied the file names and found the address for the password(as highlighted).



I copied the file address and used *cat* to find password.

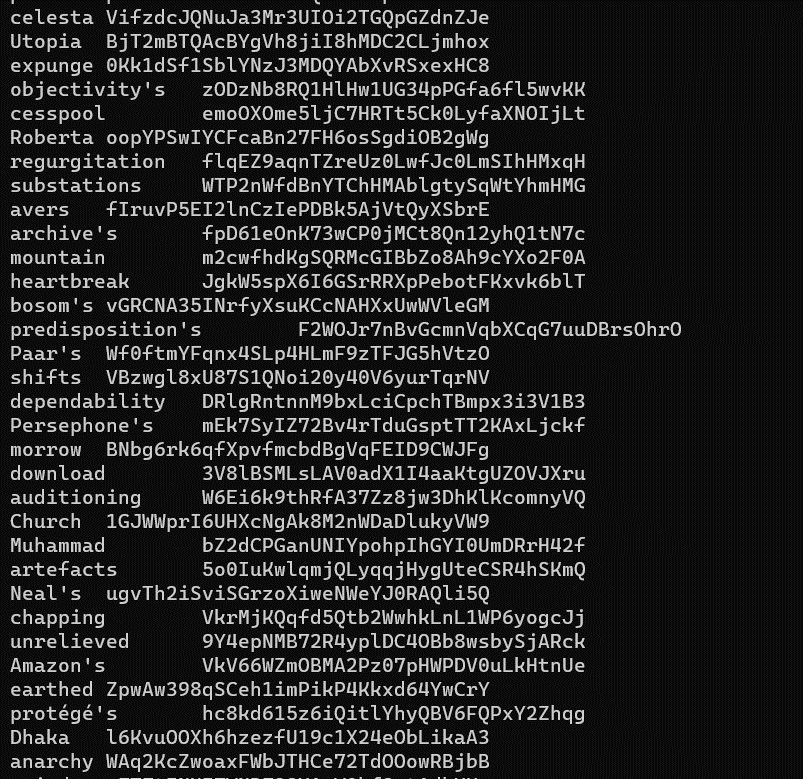
***LEVEL 8***

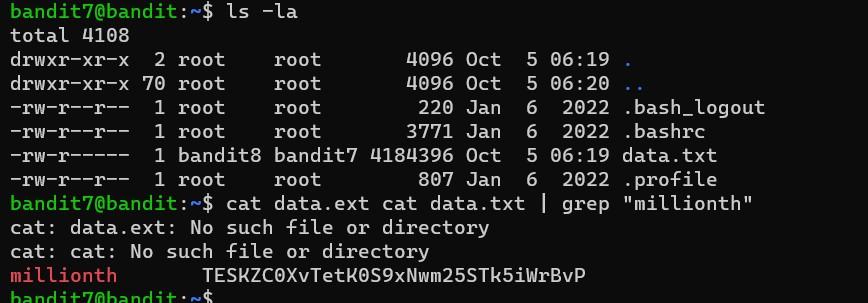
***LEVEL GOAL:*** The password for the next level is stored in the file **data.txt** next to the word **millionth**

*COMMANDS USED:*

* ls
* cat
* grep - searches a file for a particular pattern of characters, and displays all lines that contain that pattern

Showed the directory using *ls.* Used *cat* to read data.txt. It showed a lot of text but not the password. The password is stored next to the word **millionth** so I used *grep* along with *cat* to filter the text and we found the password next to word millionth.





I made a mistake while typing and it did not show the expected at first but after correcting data.ext as data.txt the expected result was shown.

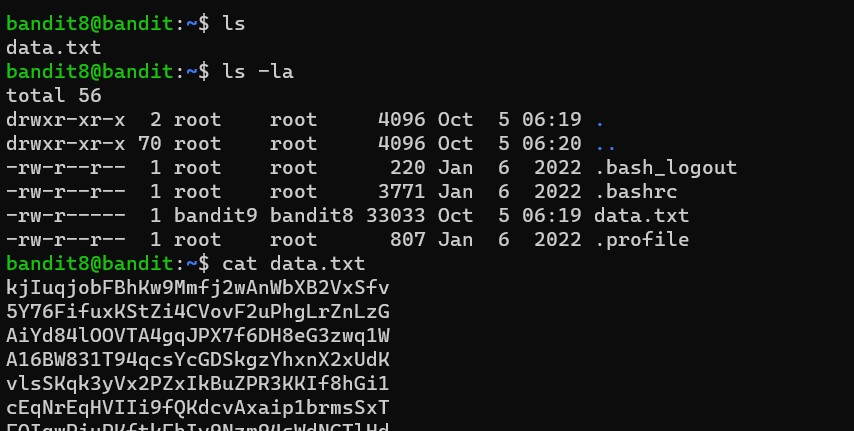
***LEVEL 9***

***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

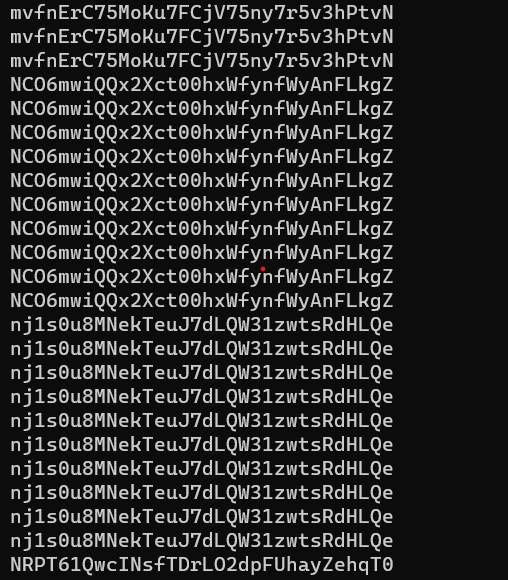
*COMMANDS USED:*

* ls
* cat
* sort
* uniq
* sort

Using *ls* and *cat* on data.txt shows the following result



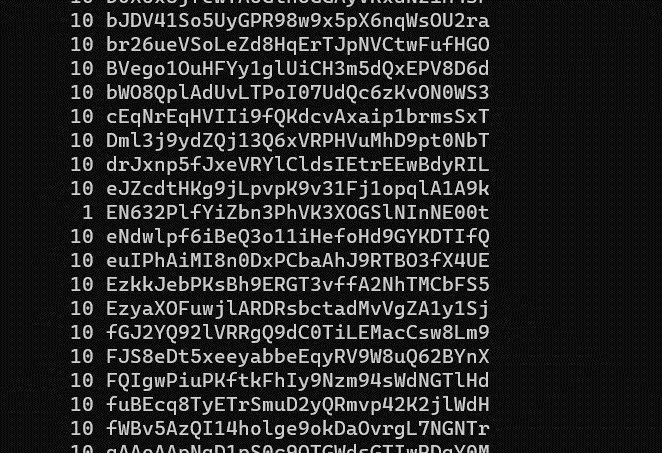
There is a lot of text. I first tried with the *sort* command, to arrange the data. But even after that it was lot of data. I had to check every single line to see if it was repeated or not.



To get better results, I used the *uniq* command alond with *-c*  to get the text along with number of times it was repeated.

On entering it, the password was shown along with number of times it had beeen repeated. All other text was repeated 10 times but the password was repeated only once.

The password is shown in the image below.

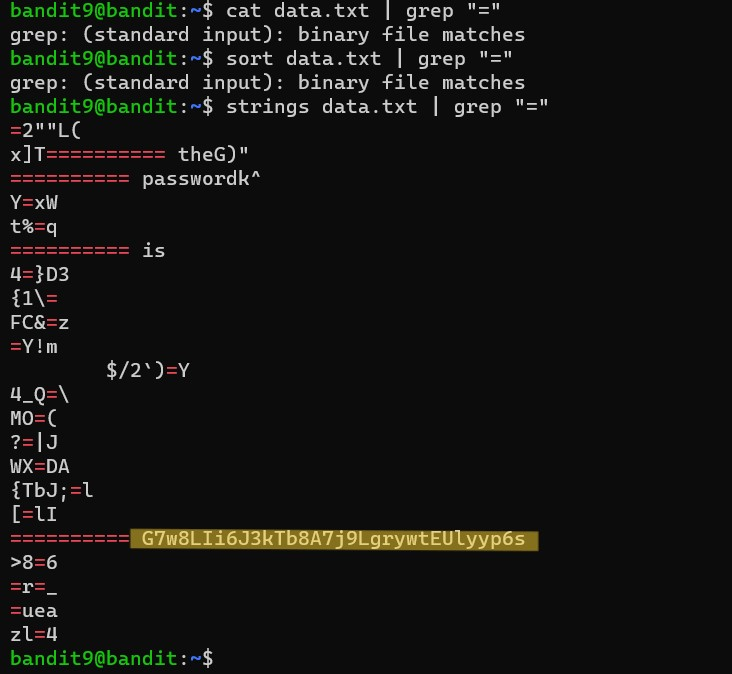


***LEVEL 10***

***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.

*COMMANDS USED:*

* ls
* cat
* grep
* strings - return each string type of characters that are printable in the file



I previously tried to *cat* along with grep but it did not show password because cat processes the non printable characters and output it to the terminal but strings ignore them.



This was result I was getting when I used *cat.* (bunch various characters)

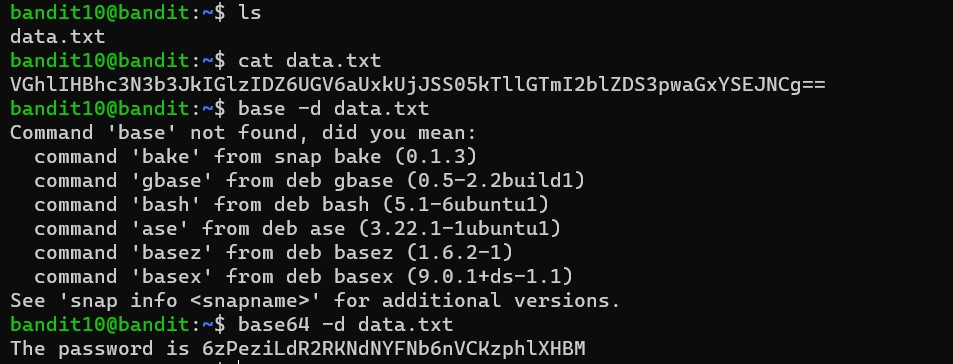
The password is highlighted in the previous image. Now onto the next level

***LEVEL 11***

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

*COMMANDS USED:*

* ls
* car
* base64: You can use different types of options with base64 command. Data can be taken from any file or standard input while encoding or decoding. After encode or decode, you can send the output in a file or print the output in the terminal.



*-d* is used to decode the base64 code.

The converted text shows the password.

LEVEL 12

***LEVEL GOAL:***

LEVEL 13

LEVEL GOAL

CHANGE FILE NAME

<IMG>

COMPRESSED

<IMG>

More compress

<img>

More compress

<img>

More something

<img>