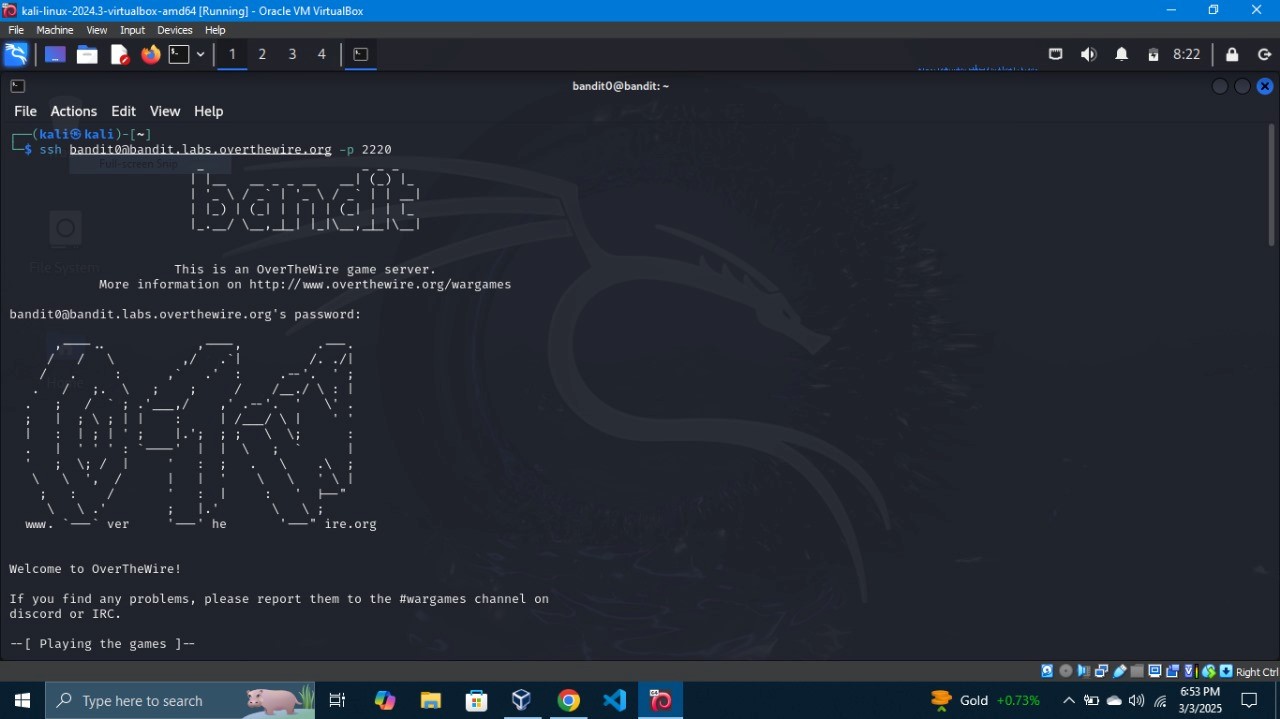
**Bandit OverTheWire Writeup:**

**Level 0:**

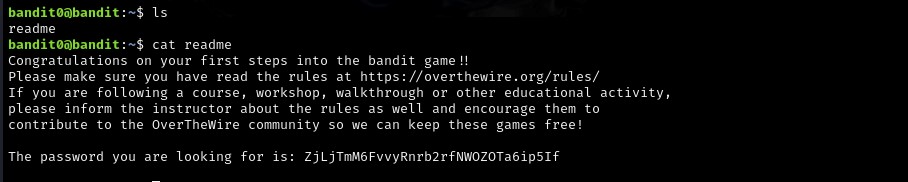
**Task :** The goal of this level is for you to log into the game using SSH. The host to which you need to connect is **bandit.labs.overthewire.org**, on port 2220. The username is **bandit0** and the password is **bandit0**. Once logged in, go to the [Level 1](https://overthewire.org/wargames/bandit/bandit1.html) page to find out how to beat Level 1.



Log into the level with ssh in server:bandit.labs.overthewire.org in the port 2220 .

**command :** ssh bandit0@bandit.labs.overthewire.org -p 2220 **username** : bandit0 **password :** bandit0 **Level 0 - 1:**

Task  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.

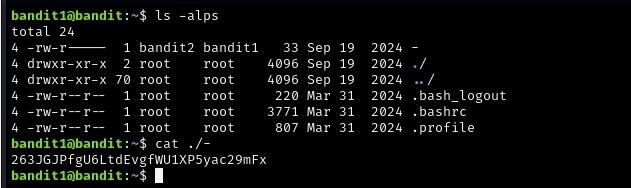


After entering the bandit0 use ls to view the directories.

Use cat to view the content of the readme file. **command :** ls and cat readme

**password**  ZjLjTmM6FvvyRnrb2rfNWOZOTa6ip5If **Level 1 - 2:**

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

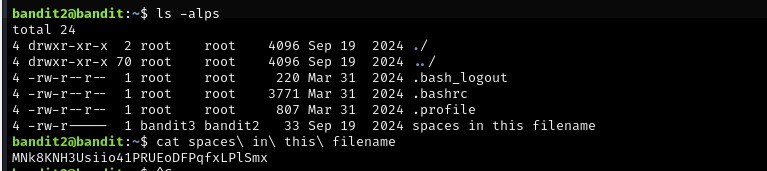


Using the previous level passkey the current level is accessed and password for next level is is in a file called - . use ls and cat to get the password.

**command :** ls -alps and cat ./-

**password :** 263JGJPfgU6LtdEvgfWU1XP5yac29mFx **Level 2 - 3:**

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



**command** : cat spaces\ in\ this\ filename \are used to show the space in the command

**password :** MNk8KNH3Usiio41PRUEoDFPqfxLPlSmx **Level 3 - 4:**

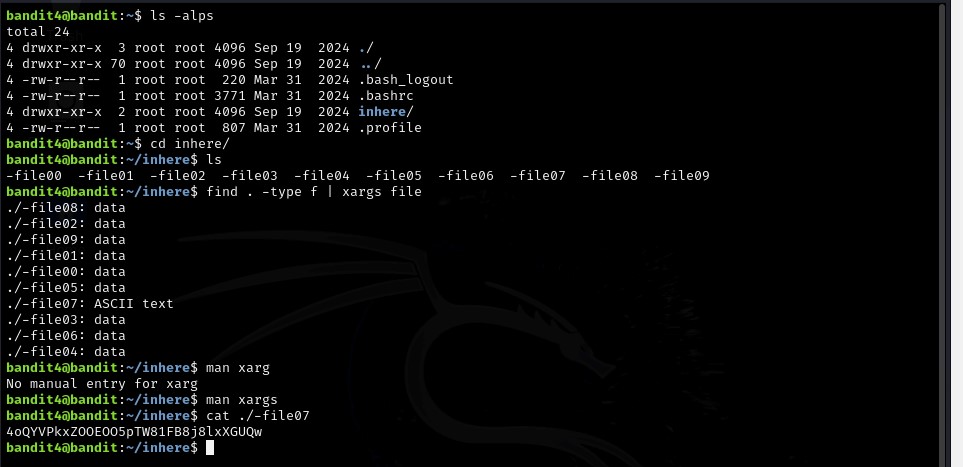
**Task :** The password for the next level is stored in a hidden file in the **inhere** directory.



In this level the password is hidden inside the directory inhere.

**command** : cd for changing directory **password**  2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ **Level 4 - 5:**

**Task :** 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 “resetˮ command.

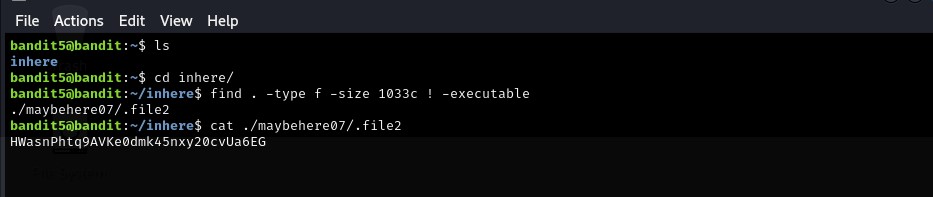


In this level the directories as many files in which the password is in the human readable file. the command find is used to get the readable file in the directory.

**command** : ls -alps, cd, find . -type f | xargs file and cat **password**  4oQYVPkxZOOEOO5pTW81FB8j8lxXGUQw **Level 5 - 6 :**

**Task :** 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



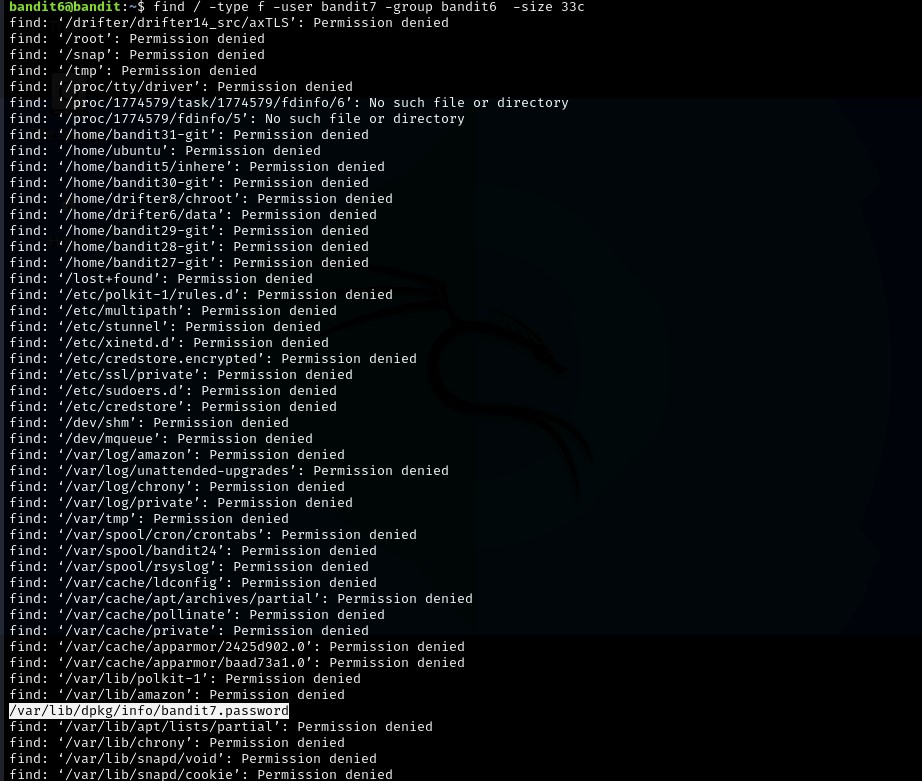
the command find . -type f -size 1033c ! executable says the properties of the given directories.

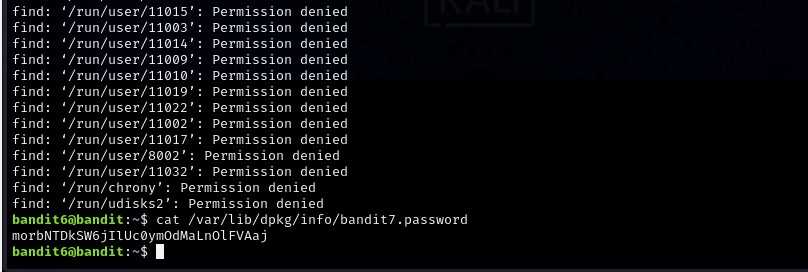
**password**  HWasnPhtq9AVKe0dmk45nxy20cvUa6EG **Level 6 - 7:**

**Task**  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

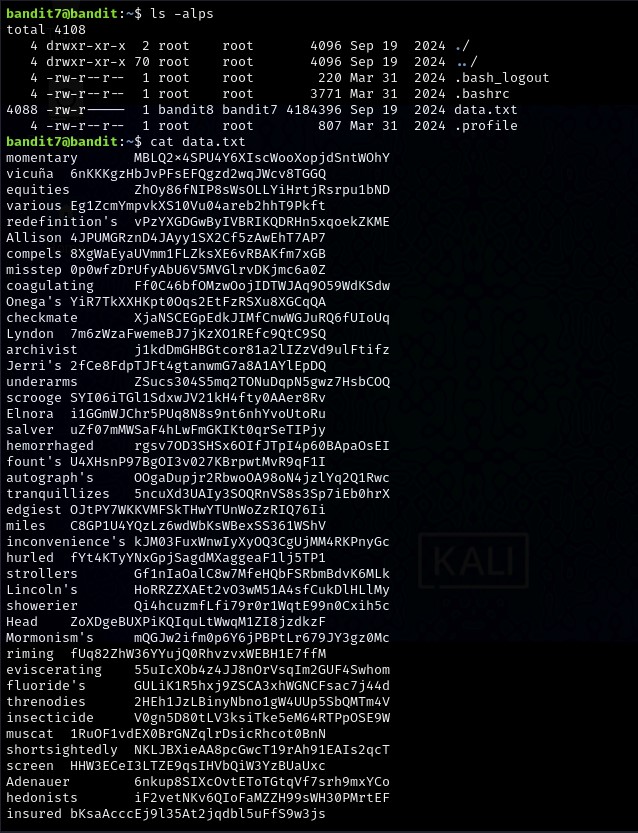


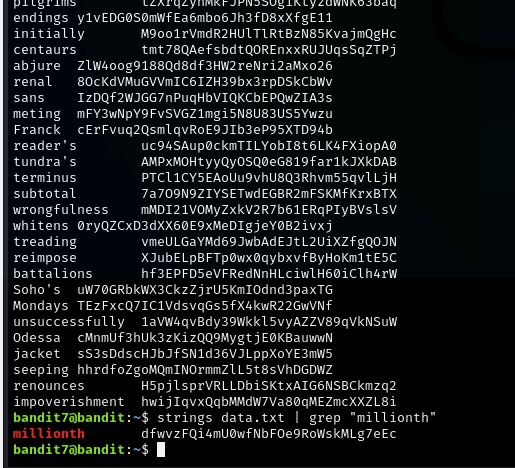


The password is stored somewhere in the server so the command find / -type f user bandit7 -group bandit6 -size 33c is used to get the packets.

**command** : find / -type f -user bandit7 -group bandit6 -size 33c **password** : morbNTDkSW6jIlUc0ymOdMaLnOlFVAaj **Level 7 - 8 :**

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



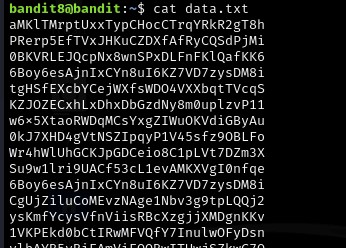


the data.txt file has many strings which are difficult to find . so, the command strings text\_file | grep “wordˮ is given to get the password.

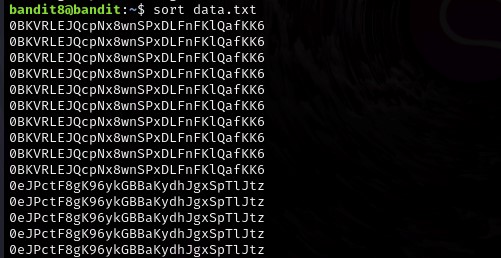
**command** : strings data.txt | grep “millionthˮ

**password** : dfwvzFQi4mU0wfNbFOe9RoWskMLg7eEc **Level 8 - 9:**

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



cat is used to view the data.txt file which is the only file.



sort is used to arrange the strings in order to find the number of times the strings are repeated.

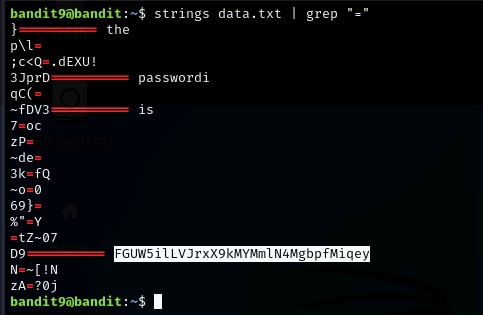


uniq -c is used to get the count of the unique strings present in the data.txt.

**command :** sort data.txt | uniq -c

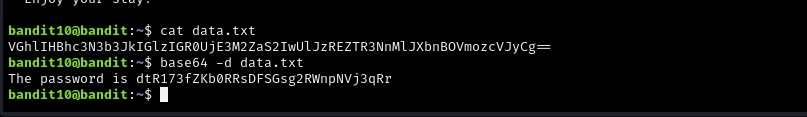
**password :** 4CKMh1JI91bUIZZPXDqGanal4xvAg0JM **Level 9 - 10 :**

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



strings data.txt gives the whole strings content of the file but the | grep “=ˮ gives the strings which has the characters. command : strings data.txt | grep “=ˮ password  FGUW5ilLVJrxX9kMYMmlN4MgbpfMiqey **Level 10 - 11 :**

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



the encoded data was in the data.txt file . We can also use other sources like cyberchief and base64 platform instead of base64 -d in linux.

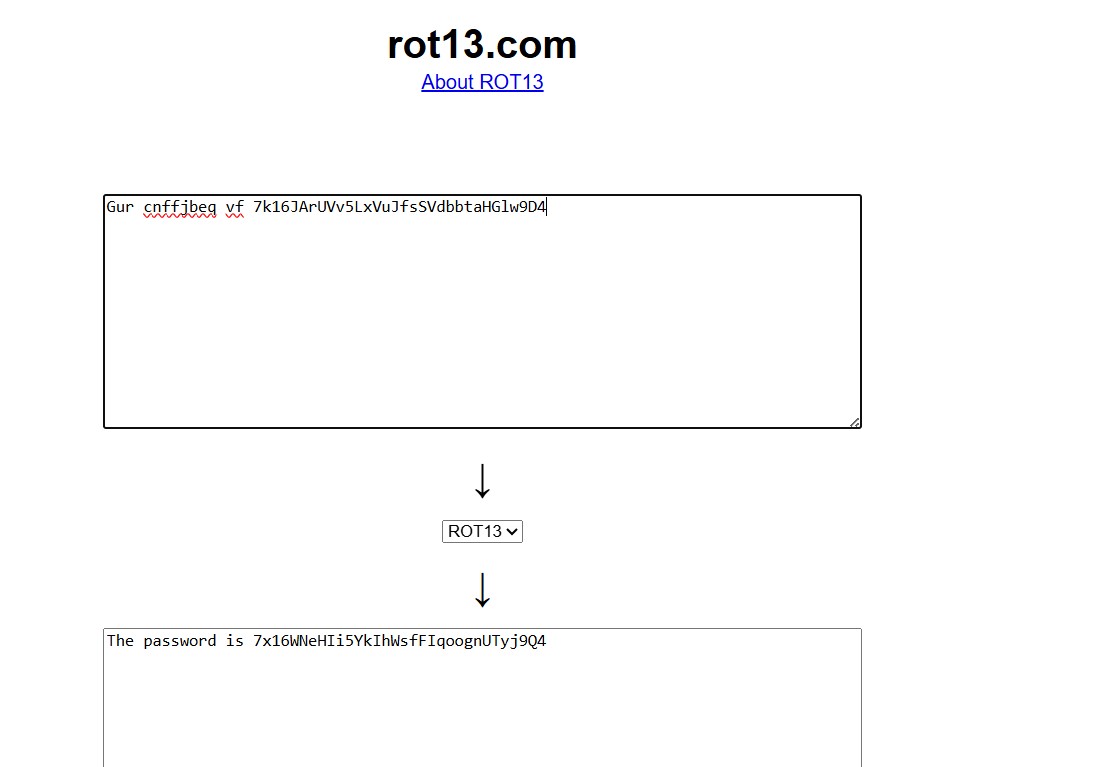
command : base64 -d data.txt

**Tool :** base64

**password** : dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr **Level 11 - 12 :**

**Task :** 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



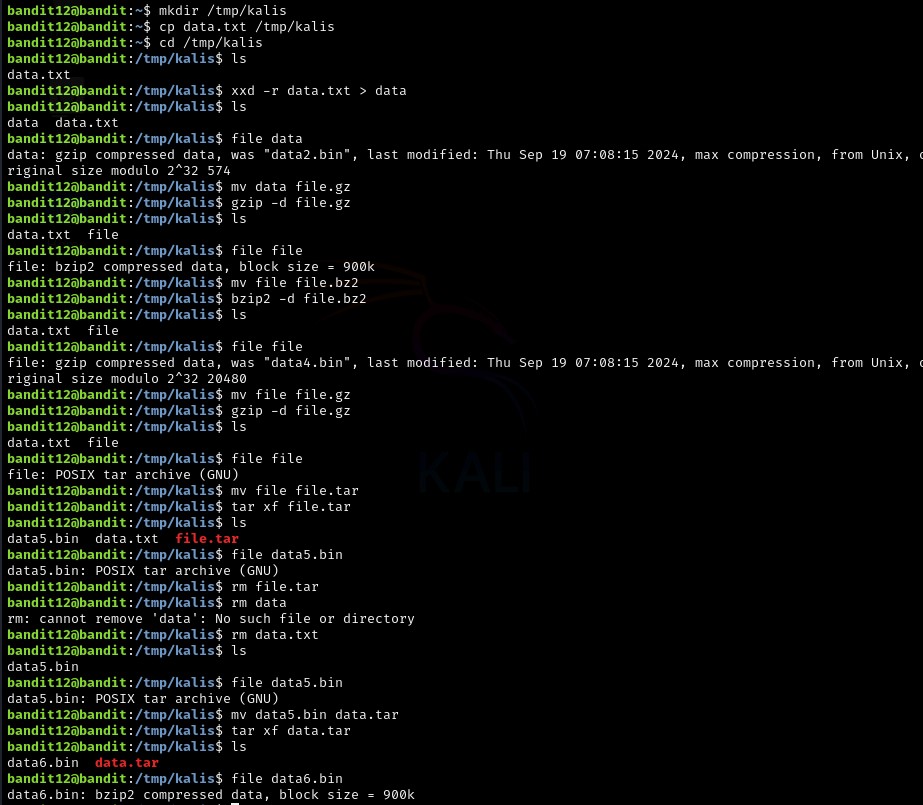


rot 13 is a tool used to decode the data. It is a online tool, the alternate for rot13 is cyberchief which is also a online platform. **Tool :** rot13

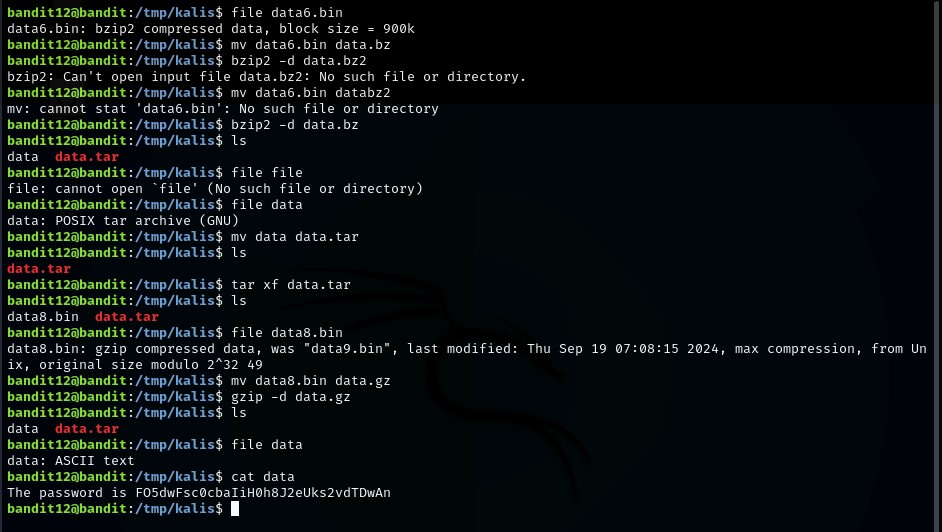
**password :** 716WNeHIi5YkIhWsfFIqoognUTyj9Q4

**Level 12 - 13 :**

**Task**  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 -dˮ. Then copy the datafile using cp, and rename it using mv (read the manpages!



Create a working directory and copy data.txt .**Convert Hexdump:** Use xxd -r to restore the binary file. Extract gzip, then bzip2, then another gzip. Extract multiple tar archives, decompressing as needed. **Retrieve Password:** Once the final file is ASCII text, use cat to display the password.

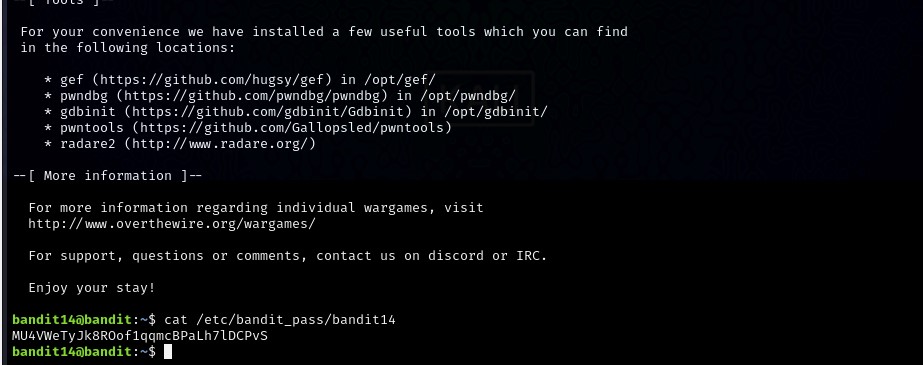


**commands :** mkdir, tar, gzip, bzip2, xxd, cp, mv, file

**password**  FO5dwFsc0cbaIiH0h8J2eUks2vdTDwAn **Level 13 - 15 :**

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



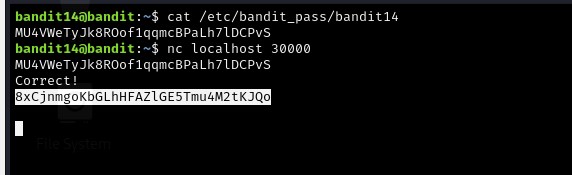


password for level 15 is derived here . The password is saved in the ssh privatekey by using bandit14@localhost the privatekey is derived from the same level.

command : ssh -i sshkey.private bandit14@localhost -p 2220 cat /etc/bandit\_pass/bandit14

**password of 14**  MU4VWeTyJk8ROof1qqmcBPaLh7lDCPvS

**password of 15**  8xCjnmgoKbGLhHFAZlGE5Tmu4M2tKJQo



**Level 15 - 16 :**

**Task**  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.

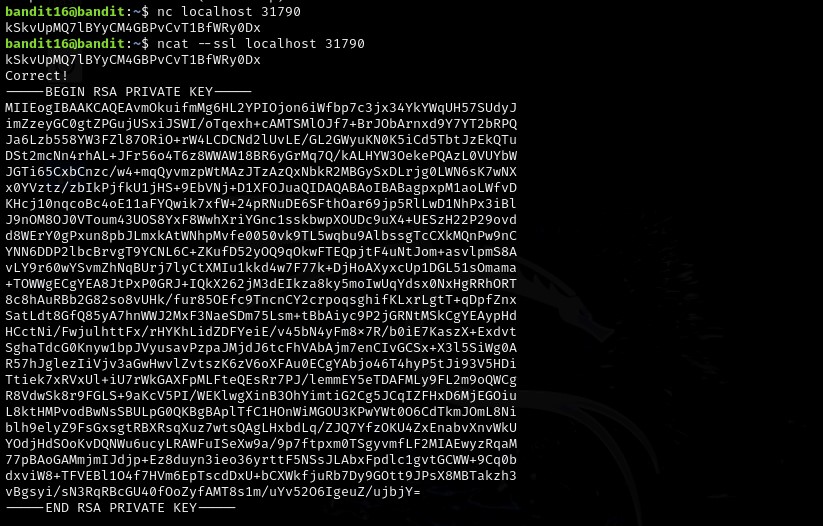


the password for next level is derived by giving the password fo current level.

**command** : ncat —ssl [localhost](http://localhost/) 30001

**password** : kSkvUpMQ7lBYyCM4GBPvCvT1BfWRy0Dx **Level 16 - 17 :**

**Task**  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/TLS and which donʼt. There is only 1 server that will give the next credentials, the others will simply send back to you whatever you send to it.





The open ports are first checked between 31000 to 32000. nmap is used to find which port is open and which is active. The result is a private sshkey a file is created to store the private keys of the levels and that file only has the permission to the user. **command** : nmap and nc.

**Level 17 - 18 :**

**Task** : here 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** Find the one line that is different between the two files.

Level 18 – 19 :

Execute a non-interactive command to read the password:

Ssh -p 2220 [bandit18@bandit.labs.overthewire.org](mailto:bandit18@bandit.labs.overthewire.org) “cat readme”

Level 19 - 20:

Use a program that allows running commands as bandit20 to access its password.

Solution:

1. Run the special program provided:

./bandit20-do cat /etc/bandit\_pass/bandit20