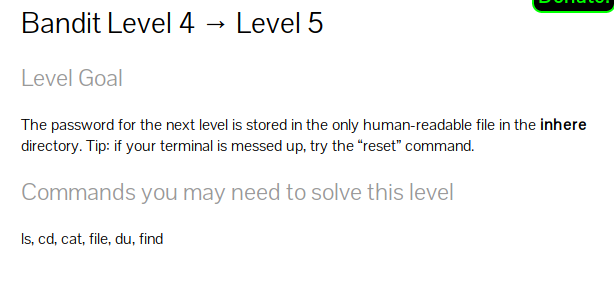
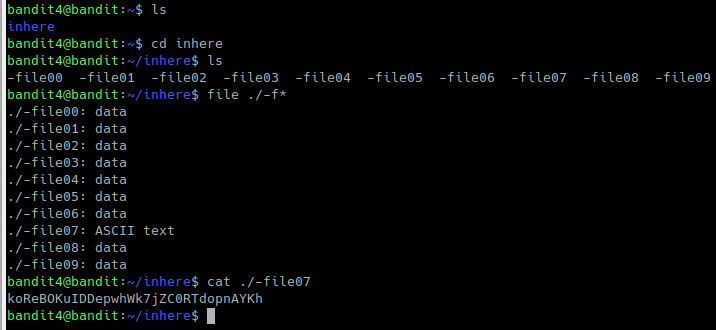


ls -a : a stands for all shows entries that starts with ‘.’ which are generally hidden

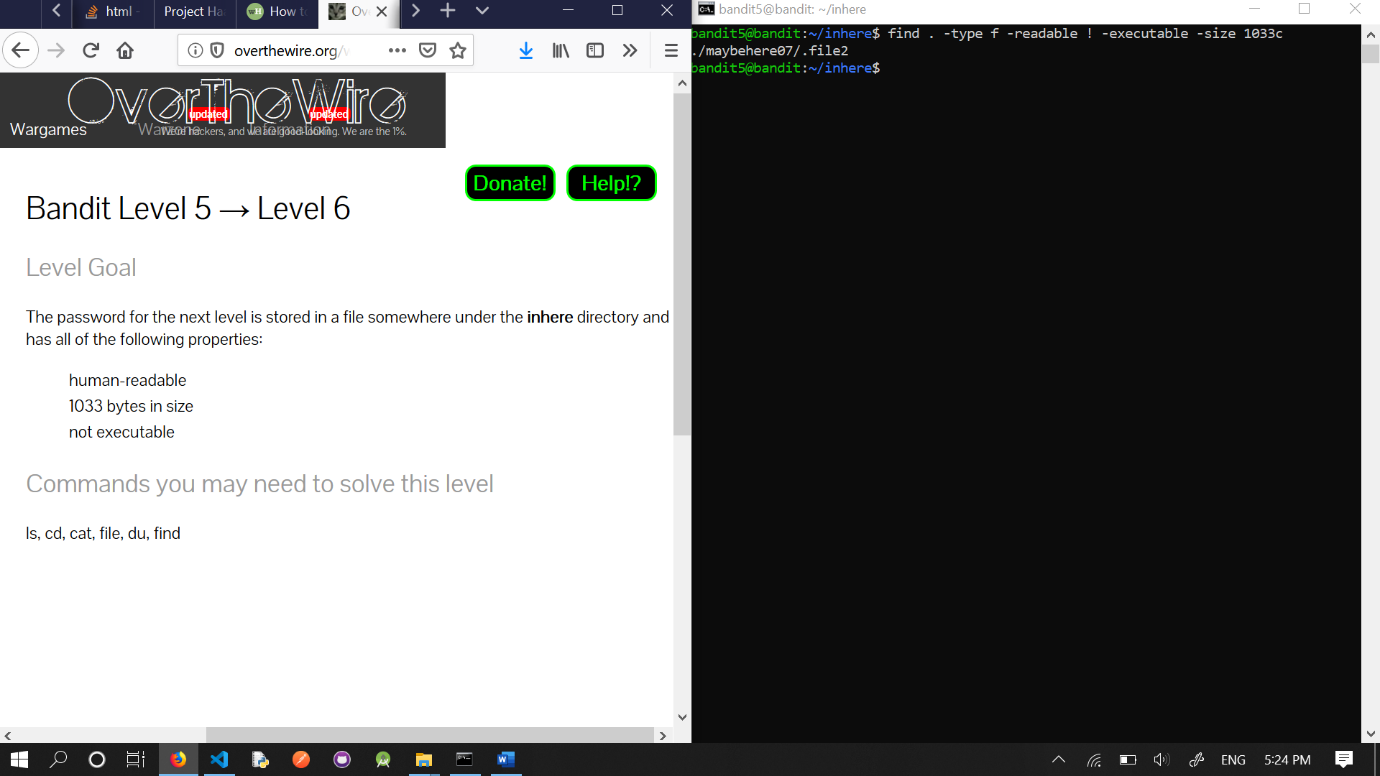




file \* lists the data types of all the files but since the file name starts with ‘-f’ \* will give ‘ ile00 ‘ instead of ‘ -file00 ‘ and so on therfore giving the wrong output.

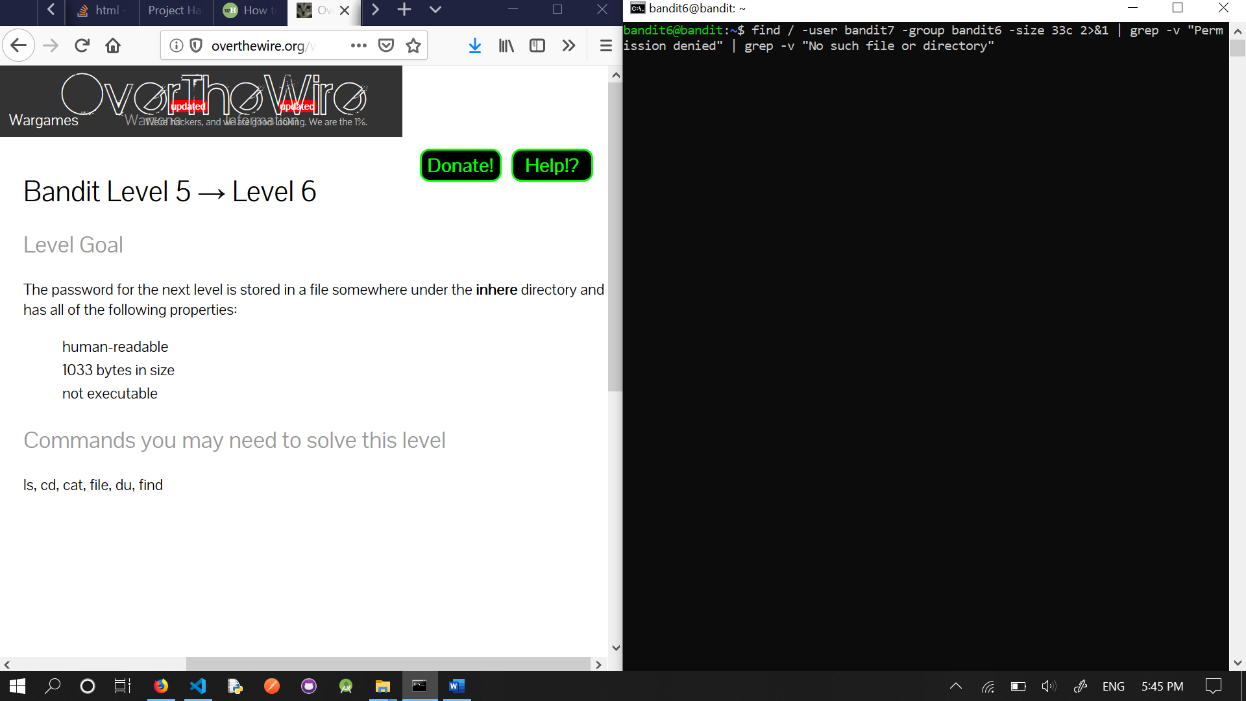
To overcome this we use ./-f\*

Level 5  Level 6



* -readable flag indicates file is “human readable” form
* ! executable because the file should not be executable :p
* -size 1033c indicating file size specified by the problem statement.

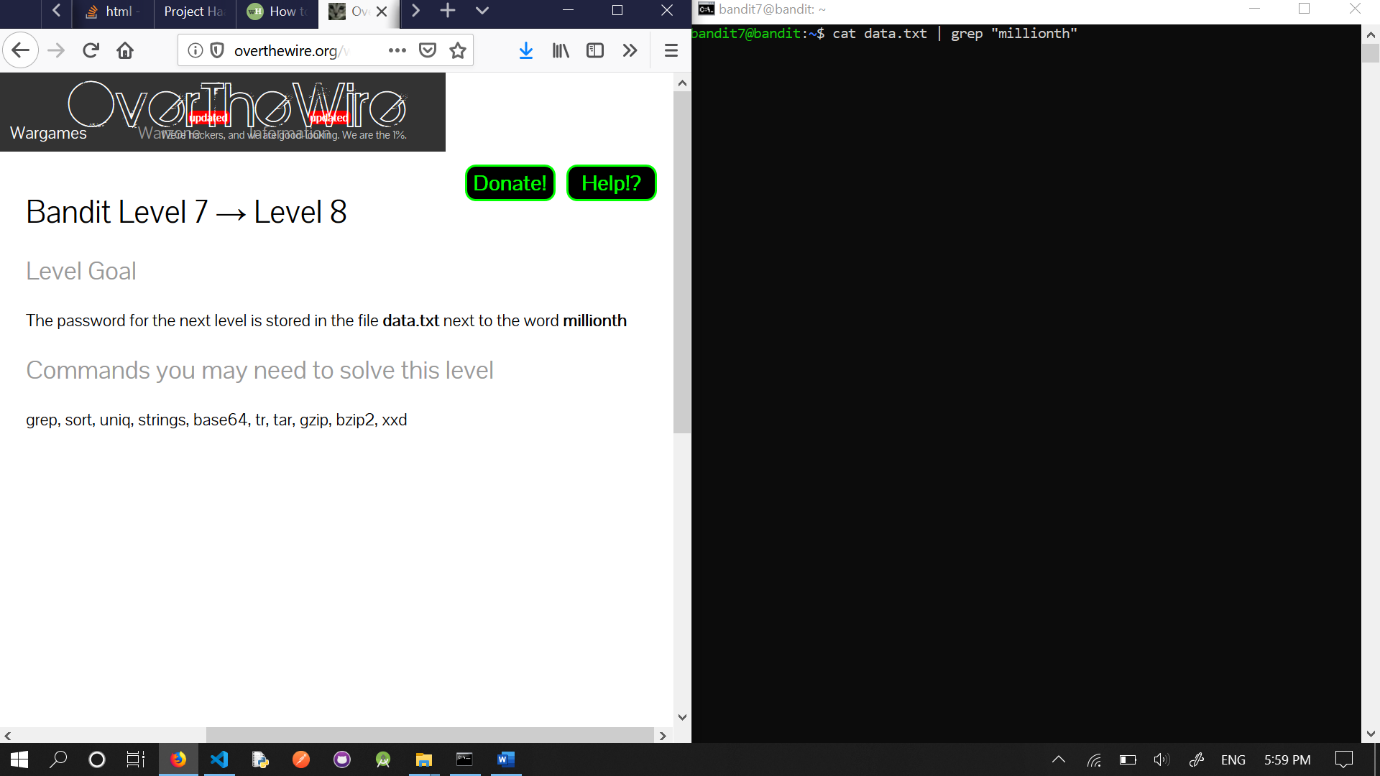
Level 6  Level 7

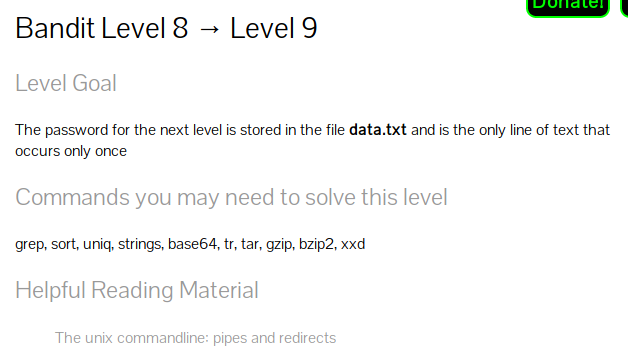


* -user, -group, -size to search for the specified user, group and file size.
* 2>&1 : basically redirect any stdout message which is an error message of type “Permission denied” or “No such file or directory” to stderr.
* The file containing password is the only file without the above mentioned errors.

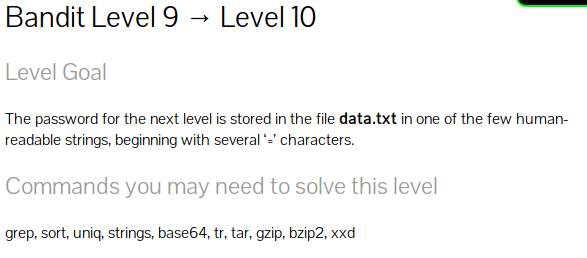
Level 7  Level 8

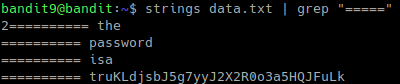
grep is a command line utility to filter text with the help of regular expressions

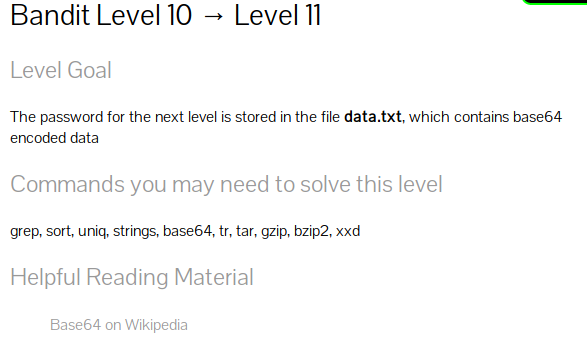


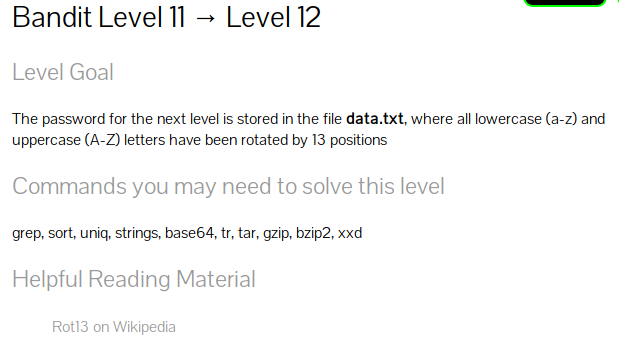








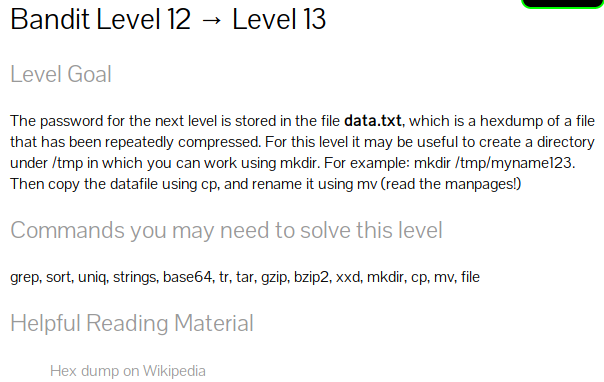


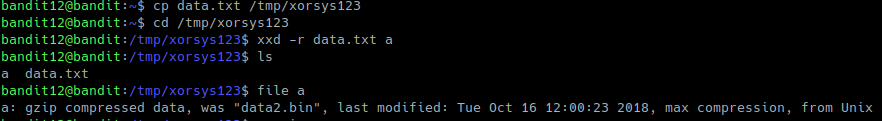


tr converts elements from one set to another

[a-zA-Z] makes a set of lowercase+uppercase in normal order

[n-za-mN-ZA-M] makes a set shifted by 13 digits

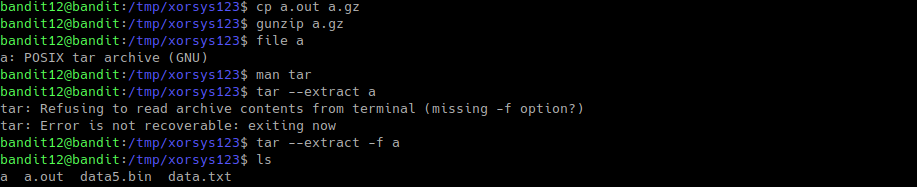
for this question we have to reverse engineer our file by finding out the type at each step and extracting them

to extract using gunzip you need to rename to the proper extension

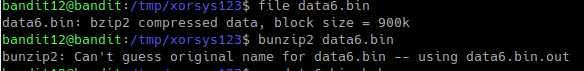






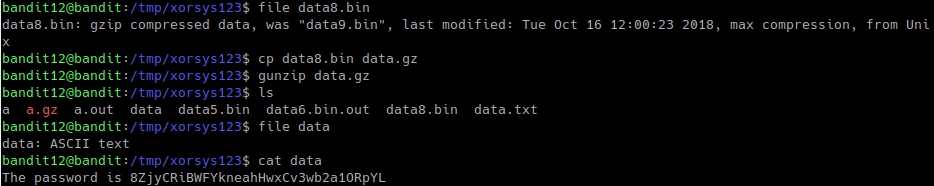


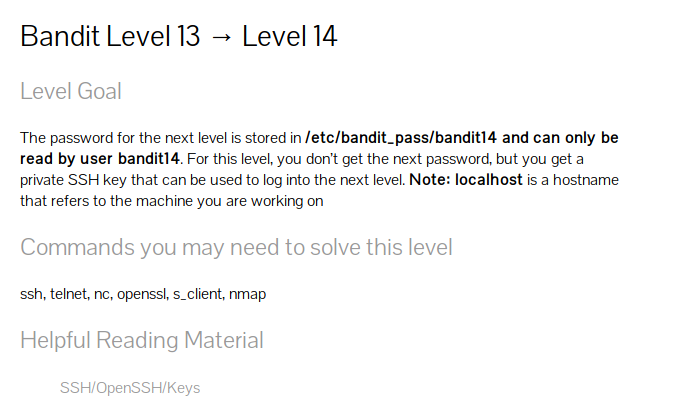








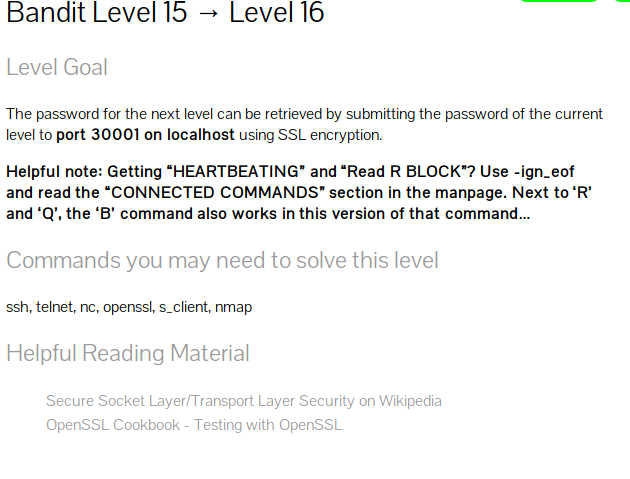
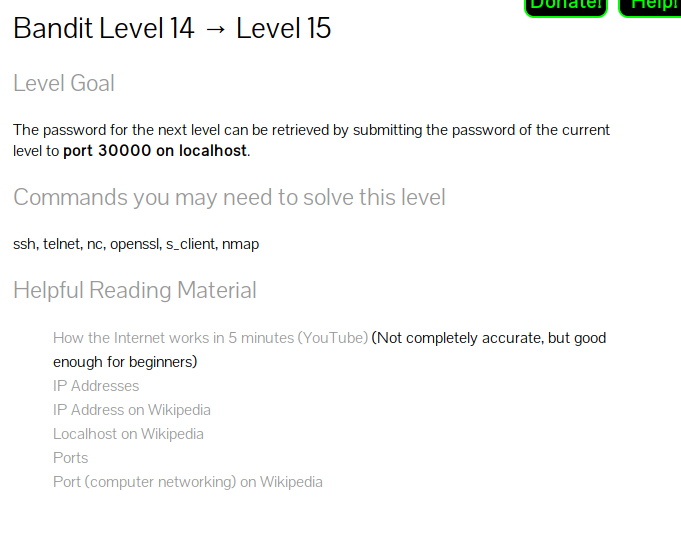
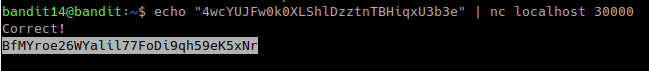








the password is only accessible by the bandit14 user therefor we need to use the bandit14 account which can be done by connecting to it using the private key.





After connecting, input password of the previous problem.

