**Bandit Level 7 → Level 8**

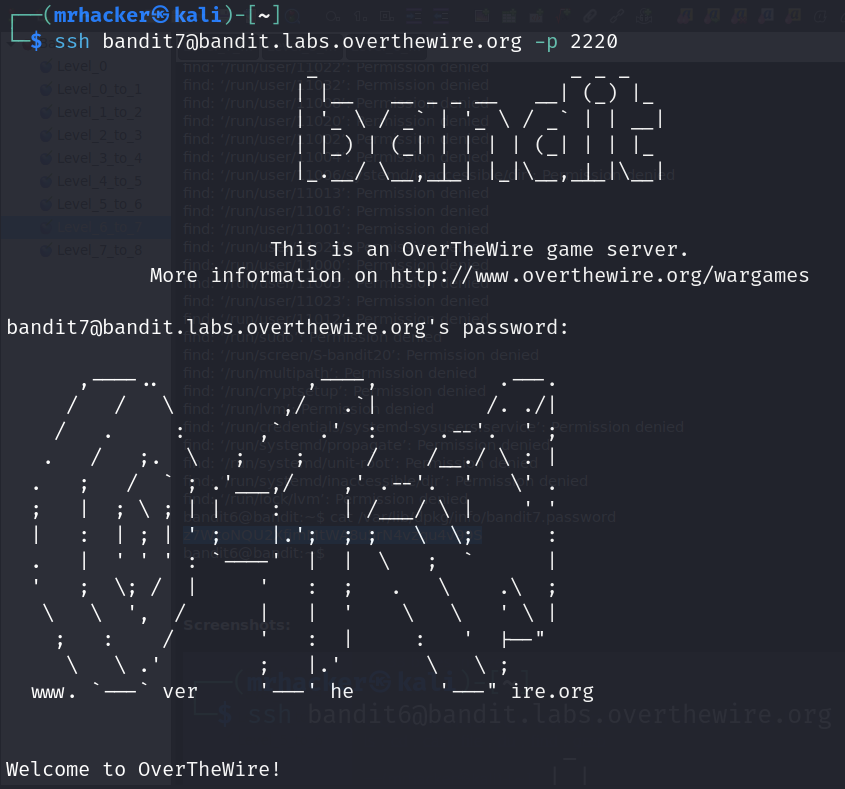
Level Goal:The password for the next level is stored in the file **data.txt** next to the word **millionth.** We will use the|: The pipe operator, which passes the output of the command on the left (cat data.txt) as input to the command on the right (grep millionth). The command grep millionth: Searches through the input for lines that contain the word "millionth" and displays those lines.

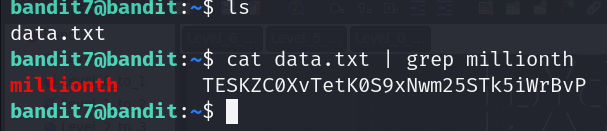
$ssh bandit7@bandit.labs.overthewire.org -p 2220

$ ls

$ cat data.txt | grep millionth

millionth **TESKZC0XvTetK0S9xNwm25STk5iWrBvP**

**SCREENSHOTS:**

****

# Bandit Level 8 → 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. **Here we used**sort: Sorts the input lines in alphabetical order. This is necessary because uniq only removes adjacent duplicate lines. uniq -u: Filters out lines that are repeated, displaying only unique lines that appear exactly once.

$ ls

$ cat data.txt | sort | uniq -u

**EN632PlfYiZbn3PhVK3XOGSlNInNE00t**

$ ssh bandit9@bandit.labs.overthewire.org -p 2220

## SCREENSHOTS:

## 

# Bandit Level 9 → 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. grep =: Searches through the input for lines that contain the equals sign = and displays those lines. The grep command filters the input based on a specified pattern (in this case, the equals sign).

$ ls

$ strings data.txt | grep =

=2""L(

x]T========== theG)"

========== passwordk^

Y=xW

t%=q

========== is

4=}D3

{1\=

FC&=z

=Y!m

$/2`)=Y

4\_Q=\

MO=(

?=|J

WX=DA

{TbJ;=l

[=lI

========== **G7w8LIi6J3kTb8A7j9LgrywtEUlyyp6s**

>8=6

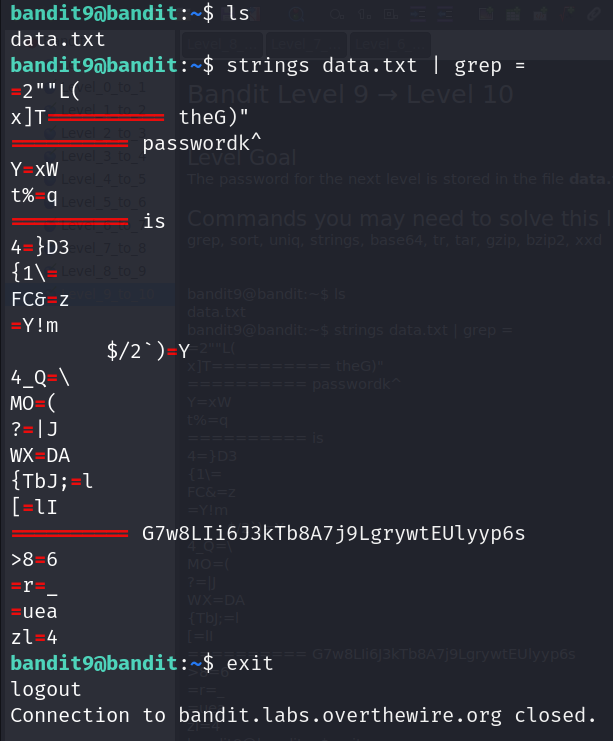
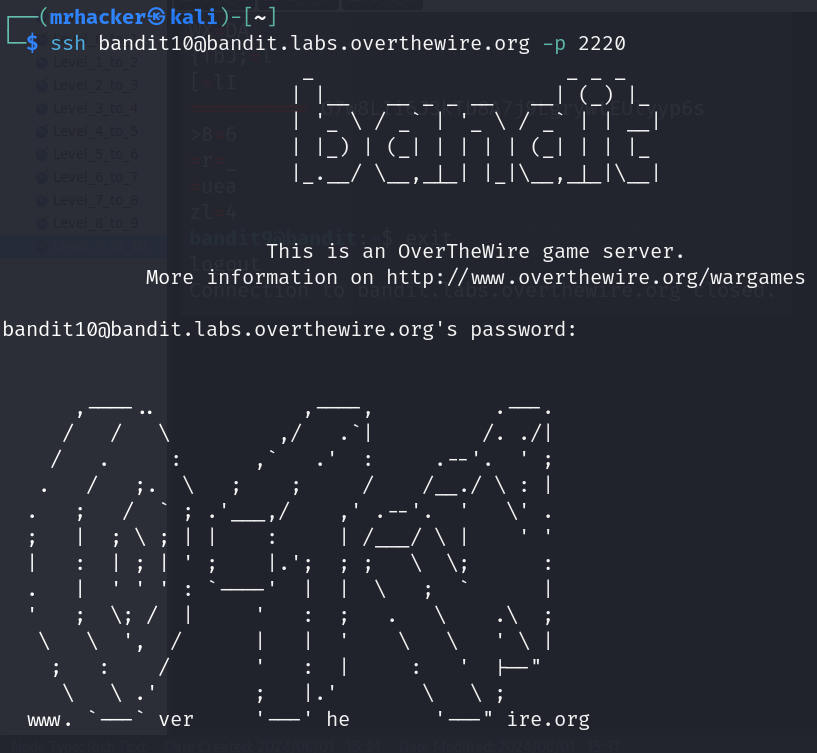
=r=\_

=uea

zl=4

$ exit

$ ssh bandit10@bandit.labs.overthewire.org -p 2220



# Bandit Level 10 → Level 11

## Level Goal : The password for the next level is stored in the file ****data.txt****, which contains base64 encoded data. Here they specified that the data is base64 encoded , so to read data we need to decode that data. base64 --decode: Decodes data encoded in Base64 back to its original binary form.

$ ls

$ cat data.txt | base64 --decode

The password is 6zPeziLdR2RKNdNYFNb6nVCKzphlXHBM

$ exit

$ ssh bandit11@bandit.labs.overthewire.org -p 2220

## SCREENSHOTS:

## 

# Bandit Level 11 → Level 12

## Level Goal: 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. The tr command translates or replaces characters from one set to another.a-zA-Z: This specifies the set of characters to be replaced, which includes all lowercase (a-z) and uppercase (A-Z) letters.n-za-mN-ZA-M: This specifies the set of characters to replace with. It represents a ROT13 transformation:

$ ls

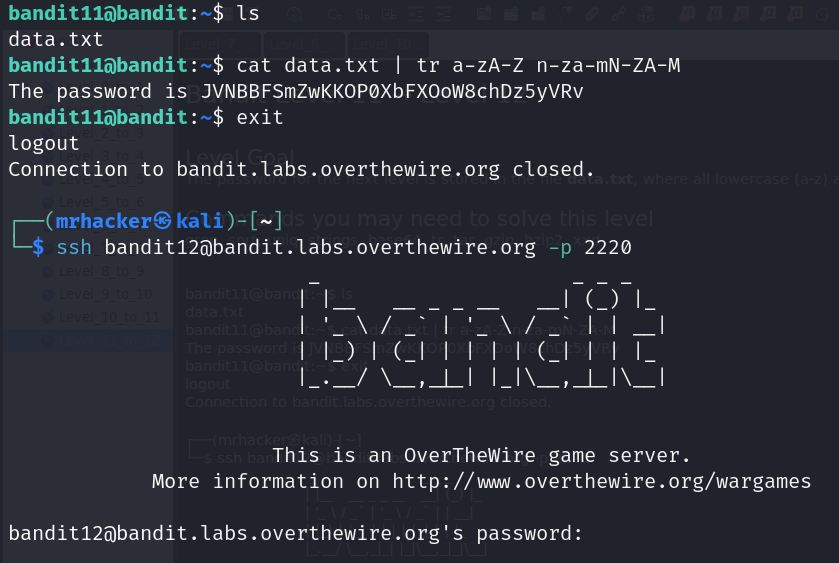
$ cat data.txt | tr a-zA-Z n-za-mN-ZA-M

The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv

$ exit

$ ssh bandit12@bandit.labs.overthewire.org -p 2220

**SCREENSHOT:**



# Bandit Level 12 → Level 13

## Level Goal: 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!).

## $mkdir /tmp/Chaithanya

## $cp data.txt /tmp/Chaithanya

## $ls

## $file data.txt #it display that it is ASCI text

## $xxd –r data.txt data1

## $mv data1 data2.gz

## $gzip –d data2.gz

## 

$ file data2

$ mv data2 data3.bz2

$ bzip2 -d data3.bz2

$ file data3

data3: gzip compressed data, was "data4.bin", last modified: Thu Oct 5 06:19:20 2023, max compression, from Unix, original size modulo 2^32 20480

$ mv data3 data4.gz

$ gzip -d data4.gz

$ file data4

data4: POSIX tar archive (GNU)

$ tar -xvf data4

data5.bin

## 

$ file data5.bin

data5.bin: POSIX tar archive (GNU)

$ tar -xvf data5.bin

data6.bin

$ file data6.bin

data6.bin: bzip2 compressed data, block size = 900k

$ mv data6.bin data7.bz2

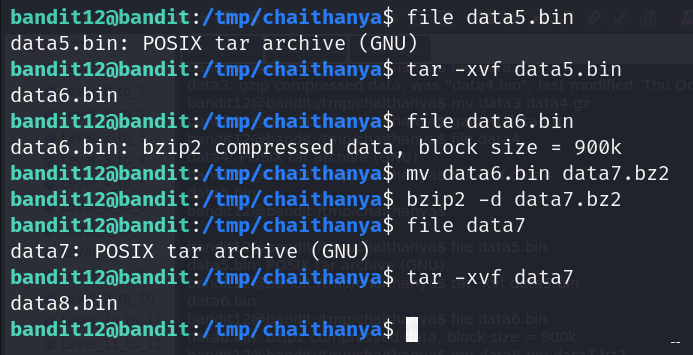
$ bzip2 -d data7.bz2

$ file data7

data7: POSIX tar archive (GNU)

$ tar -xvf data7

data8.bin



$ file data8.bin

data8.bin: gzip compressed data, was "data9.bin", last modified: Thu Oct 5 06:19:20 2023, max compression, from Unix, original size modulo 2^32 49

$ mv data8.bin data9.gz

$ gzip -d data9.gz

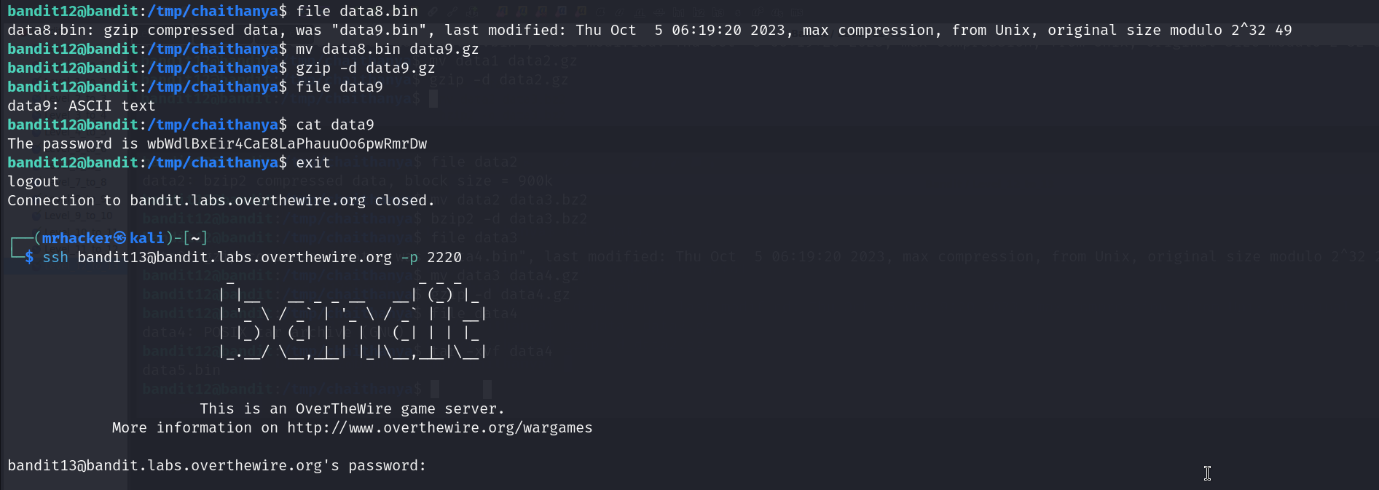
$ file data9

data9: ASCII text

$ cat data9

The password is wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw

$ exit

$ ssh bandit13@bandit.labs.overthewire.org -p 2220