


Bandit Level 11 → Level 12

SSH Parameters	
Host:	bandit.labs.overthewire.org
Port:	2220


Website URLs	
Level 11 → Level 12	OverTheWire: Level Goal: Bandit Level 11 → Level 12
Level 12 → Level 13	OverTheWire: Level Goal: Bandit Level 12 → Level 13

Passwords		
Level	User Name	Password
Level 11 → Level 12	bandit11	6zPezilDlR2RKNdNYFNb6nVCKzphIXHBM
Level 12 → Level 13	bandit12	JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv



Wargames

Information updated



We're hackers, and we are good-looking. We are the 1%.

[SSH Information](#)
Host: bandit.labs.overthewire.org
Port: 2220

[Bandit](#)
Level 0 → Level 1
Level 1 → Level 2
Level 2 → Level 3
Level 3 → Level 4
Level 4 → Level 5
Level 5 → Level 6
Level 6 → Level 7
Level 7 → Level 8

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

Commands you may need to solve this level

`grep, sort, uniq, strings, base64, tr, tar, gzip, bzip2, xxd`

Helpful Reading Material

[Rot13 on Wikipedia](#)

[Donate!](#) [Help!?](#)

```

bandit11@bandit:~$ #####
bandit11@bandit:~$ #Over the Wire - Bandit - Level 11 - Solution Set
bandit11@bandit:~$ #####
bandit11@bandit:~$ #Execution of id and whoami command to confirm presence in Bandit Level 11
bandit11@bandit:~$
bandit11@bandit:~$ id && whoami
uid=11011(bandit11) gid=11011(bandit11) groups=11011(bandit11)
bandit11
bandit11@bandit:~$
bandit11@bandit:~$
bandit11@bandit:~$ #####
bandit11@bandit:~$ #Execution of pwd command to determine current working directory
bandit11@bandit:~$
bandit11@bandit:~$ pwd
/home/bandit11
bandit11@bandit:~$
bandit11@bandit:~$ #####
bandit11@bandit:~$ #Execution of ls -la command to view contents of current directory. The -l option switch outputs file metadata in
long format and the -a option/switch lists hidden files [preceded by a period[.]]
bandit11@bandit:~$
bandit11@bandit:~$ ls -la
total 24
drwxr-xr-x  2 root    root    4096 Apr 23 18:04 .
drwxr-xr-x 70 root    root    4096 Apr 23 18:05 ..
-rw-r--r--  1 root    root     220 Jan  6 2022 .bash_logout
-rw-r--r--  1 root    root    3771 Jan  6 2022 .bashrc
-rw-r----- 1 bandit12 bandit11 49 Apr 23 18:04 data.txt
-rw-r--r--  1 root    root     807 Jan  6 2022 .profile
bandit11@bandit:~$

```

```

bandit11@bandit:~$ #####
bandit11@bandit:~$ #Determine file type of data.txt file
bandit11@bandit:~$ #####
bandit11@bandit:~$
bandit11@bandit:~$ file data.txt
data.txt: ASCII text
bandit11@bandit:~$
bandit11@bandit:~$ #Based on the output of the file command we note data.txt is an ASCII file with human readable text
bandit11@bandit:~$
bandit11@bandit:~$ #####
bandit11@bandit:~$ #Utilize the cat command to read/view the data.txt file
bandit11@bandit:~$
bandit11@bandit:~$ cat data.txt
Gur cnffjbeq vf WIA00SFzMjXXBC0KoSKBbJ8puQm5lIEi
bandit11@bandit:~$

```

```

10 #Per the Bandit Level 11-->12 instructions the text in the data.txt file is deciphered by rotating/advancing each letter (regardless of case) by
thirteen (13) characters. In example, an "A" [in the data.txt file] would translate to "N". To assess this methodology we manually decipher the
first three words and first and last characters in the data.txt string.
11
12 #The string, from the data.txt file is, "Gur cnffjbeq vf "
13
14 #An example of the translations to decipher the password is:
15 #A -----> N
16 #a -----> n
17 #B -----> O
18 #b -----> o
19 #Z -----> M
20 #z -----> m
21
22 #Translation of the first three words, via the ROT13 cipher:
23 #"Gur cnffjbeq vf"
24 #"The passsowrd is"
25
26 #Translation of the first and last letters of the Level 12 password:
27 #password string: "WIA00SFzMjXXBC0KoSKBbJ8puQm5lIEi"
28 #W -----> J
29 #i -----> v
30
31 #The tr command reads data in a file and translates it based on the source and corresponding deciphering keys provided
32 #Each character in the deciphering key is 13 characters ahead of the corresponding character in the source key
33 #In the command below the source key is "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
34 #The characters, in this string, are translated to the relative characters in the strings below
35 #The deciphering key is "NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyzabcdefghijklm"
36 #As such, W is converted J , I is converted to V, A is converted to N, etc.
37

```

```
bandit11@bandit:~$ #####
bandit11@bandit:~$ #tr command to decipher Bandit 12 password
bandit11@bandit:~$
bandit11@bandit:~$ cat data.txt | tr "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" "NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyz
bcdefghijklm"
The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv
bandit11@bandit:~$
bandit11@bandit:~$
bandit11@bandit:~$ #Based on the output of the tr command above we noted the first three letters of the output, and the first and las
t letter of the password, reconcile to the manual reconciliation performed above
bandit11@bandit:~$
bandit11@bandit:~$ #####
bandit11@bandit:~$ #We utilize the cut command [by piping the output from the tr command to it] to extract the password from the stri
ng above. We invoke the delimiter option/switch on space [-d " "] and the field option/switch to extract the fourth field [-f 4]
bandit11@bandit:~$
bandit11@bandit:~$ cat data.txt | tr "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" "NOPQRSTUVWXYZABCDEFGHIJKLMnopqrstuvwxyz
bcdefghijklm" | cut -d " " -f 4
JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv
bandit11@bandit:~$
bandit11@bandit:~$ The string above is the password to authenticate to Bash Level 12
```

```
bandit11@bandit:~$ #####
#Appendix - Over the Wire - Bandit 11 ----> 12
#####

#Watched a YouTube video and noted the creator utilized a more efficient source and destination key
#Solution, with this key, is below
#YouTube Video: https://www.youtube.com/watch?v=opzQ91vYy6k

#*****Note: It is a good idea to look at the solution sets of othert to look at alternative/more efficient methods/code/commands of
solving the levels. This is a great way to learn. Often with Linux/coding there are multiple ways to solve a problem

#####

#tr command with more efficient way of defining source and destination keys
bandit11@bandit:~$
bandit11@bandit:~$ cat data.txt | tr a-zA-Z n-za-mN-ZA-M
The password is JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv
bandit11@bandit:~$
bandit11@bandit:~$ #This password is identical to the methodology above
#Utilize cut command to extract the password portion of the string

#With the cut command we invoked the delimiter option/switch on space [-d " "] and the field option to extract the fourth column of t
he string.
bandit11@bandit:~$
bandit11@bandit:~$ cat data.txt | tr a-zA-Z n-za-mN-ZA-M | cut -d " " -f 4
JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv
bandit11@bandit:~$
```

Level 12 —> Level 13 Password

JVNBBFSmZwKKOP0XbFXOoW8chDz5yVRv